



A FPGA based Generic Architecture for Polynomial Matrix Multiplication in Image Processing

Prof. Dr. S. K. Shah¹, S. M. Phirke²

Head of PG, Dept. of ETC, SKN College of Engineering, Pune, India¹

PG Student [VLSI & Embedded System], Dept. of ETC, SKN College of Engineering, Pune, India²

ABSTRACT: Most of the image processing applications uses convolution for obtaining various filtering effects depending on kernel image. As convolution in time domain can be computed as simple multiplication in frequency domain. Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) is best solution to convert time domain matrix into frequency domain and vice versa. In this proposed system, architecture is designed for Polynomial Matrix Multiplication (PMM) using Xilinx system generator tool. A dedicated generic architecture for 1D/2D FFT/IFFT is designed with no dependency on order of input matrices. A model for PMM is simulated in MATLAB Simulink and the response time and Mean square error is calculated for methods like multiplication with FFT and Without FFT. VHDL code for the same model is generated from Xilinx System Generator and simulated in Xilinx ISE, and resources utilized are calculated for order of polynomial up to 64. Proposed model is also tested on some image filtering applications such as sharpening, blurring and smoothing. Also the architecture is implemented on Virtex-5 to get effective filtering effects as results.

KEYWORDS: Fast Fourier Transform, Inverse Fast Fourier Transform, Polynomial Matrix Multiplication, Xilinx System Generator tool, MATLAB Simulink.

I. INTRODUCTION

In MIMO communications, signal processing, image processing, biomedical engineering, optical computing, etc convolution is involved for different signals. For complex signals convolution takes more time and resources also, hence here PMM is considered equivalent to convolution. As convolution in time domain can be represented as multiplication of their FFTs in frequency domain. FFT can be used to characterize the signal into their magnitude and phase [1]. The Fourier transform is a powerful tool to analyse the signals and construct them to and from their frequency components, hence this is chosen to perform transformation in proposed architecture. One and multi-dimensional digital convolution and correlation operations are widely used for pre-processing in image processing applications such as image filtering, enhancement and recognition. As the number of arithmetic operations is very large and the demand for real time high resolution images is increasing day by day, this computation requirement becomes extensive and the need for high-performance parallel image processing algorithms is becoming more important [2].

To effectively compute convolution, PMM is best solution as it gives accurate results in less time. Also a very large class of computations with matrices, graphs, and regular and Boolean expressions can be reduced to matrix multiplication [3]. But no attention is given on hardware implementation of PMM till now. In image processing for image enhancement sharpening is the most important pre-processing step. Such pre-processing steps like sharpening, blurring, smoothing are performed with convolution method and will be best executed if performed with PMM architecture proposed.

Convolution in time domain can be computed as just simple multiplication in frequency domain. Time complexity for convolution is $O(N^2)$ while same for FFT based convolution is $O(N \log_2 N)$. As image sizes and bit depths grow larger, order of input matrix also increases thus software has become less useful [4]. Still less attention is given on hardware implementation of PMM. If PMM is implemented on a dedicated parallel processing system, fast convolution would



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

require $O(N \log_2(N)/\kappa)$ time, κ being the number of processing units [5]. Field-Programmable Gate Array (FPGA) are ideal choice for implementation of PMM on hardware. FPGAs hardware implementation is preferred because of its parallelism working properties. Because of this speed of operation increases, so it will be beneficial in reduction of execution time.

II. LITERATURE SURVEY

For many signal processing applications the problems of narrowband signals mixing is extended to broadband signals. For that Singular Value Decomposition (SVD) [6], Eigen Value Decomposition (EVD) [7], Polynomial Eigen Value Decomposition (PEVD) tools are used, in which PEVD can factorize the para-Hermitian polynomial matrix into a product of diagonal polynomial matrix and Para-Unitary (PU) matrix. PU matrix can preserve the total signal power at every frequency [4]. Second order best sequential rotation (SBR2) is one of the methods used to generate FIR PU matrix to diagonalize the polynomial matrix. For high speed real time applications polynomial matrix manipulations become difficult as diagonalization method is less efficient at high speeds. For that SBR2 algorithm used in parallel (SBR2P) is used which produces the diagonalized para-Hermitian polynomial matrix and related FIR PU filter bank. Server Kasap, and Soydan Redif shows that SBR2P algorithm can be implemented in hardware with the help of highly pipelined FPGA architecture [8-9]. Chi Hieu Ta and Stephan Weiss presented an efficient method for shortening the Order of PU matrices in SBR2 algorithm [10]. J. Foster *et al.* proposed method to limit the order of polynomial matrices in SBR2 algorithm [11].

III. PROPOSED ARCHITECTURE

After referring to different papers and keeping in mind the need & challenges in developing generic architecture for PMM, the proposed design is undertaken as a project. In many image processing applications there is need of performing convolution between two images. In such cases there is need to use PMM instead of convolution with the generic FFT/IFFT architecture. This will help in ease of operation with less data so automatically processing time and memory requirement for storage will be reduced. This design deals with the PMM of input images and kernels so as to get respective output image like sharpened, blurred, and smooth. This design includes implementation of the same architecture on MATLAB Simulink. In this paper, the proposed system is designed as the optimized hardware implementation of PMM developed in Xilinx system generator tool [12] in MATLAB Simulink. The proposed block diagram is as shown in Fig. 1. The same architecture is also implemented on Virtex-5 and device utilization is observed.

The proposed system has operations as follows:

1. Firstly, the input images or polynomial equations are converted into the polynomial matrices.
2. These matrices/ vectors are fed to generic 1D/2D FFT architecture to compute FFT of it.
3. The output of FFT is in the form of real and imaginary values. These real and imaginary values of one matrix are multiplied with real and imaginary value of other matrix.
4. The multiplied values i.e. output of multiplication process are fed to IFFT architecture.
5. As a result, PMM performed on Xilinx system generator is obtained. Here it is applied to application of image filtering i.e. sharpening, blurring.
6. The same architecture is executed on FPGA by converting the System generator file into Xilinx environment.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

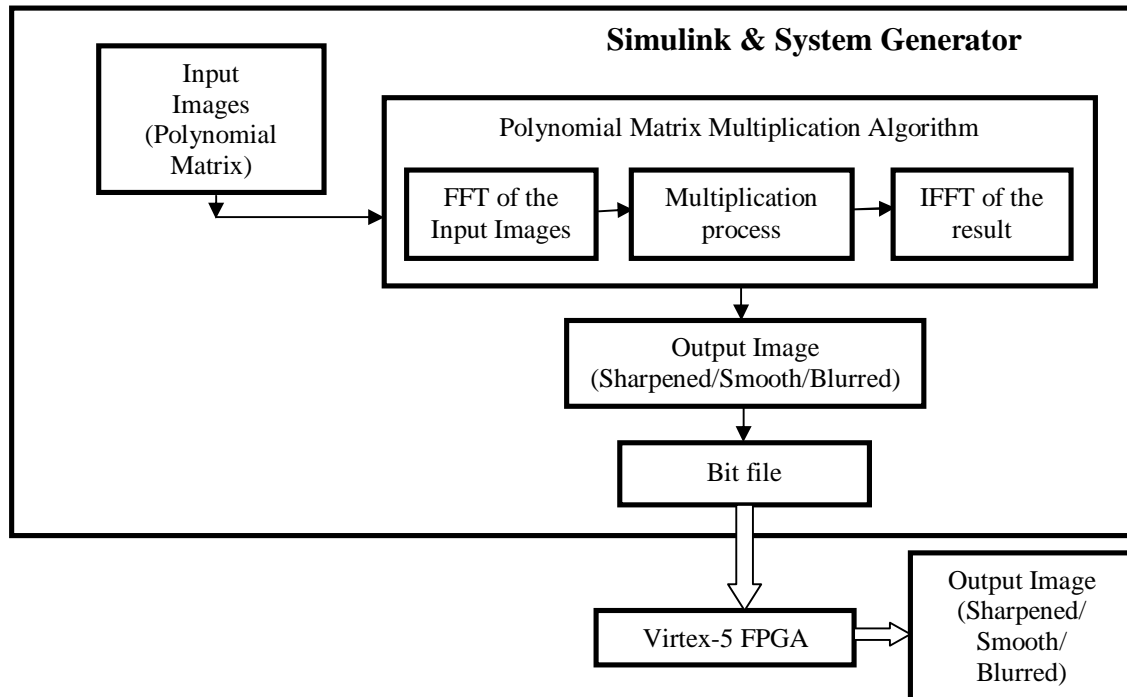


Fig. 1: Proposed Block Diagram

The important part of the proposed PMM architecture is a generic 1D/2D FFT/IFFT architecture. The block diagram for this architecture is as shown in Fig. 2. As many applications in image processing which uses PMM have 2D data or image with complex data also. To access the data efficiently and give accurate output multiplication must be performed on 2D complex data. In many papers inbuilt FFT/IFFT blocks are used to design the architecture but these blocks are very complex and requires more resources. So in this paper a new architecture is put forward for simple and easy FFT/IFFT calculation with less resources. FFT 2D is calculated as follows

$$\text{FFT-2D (I)} = (\text{FFT-1D (I)} * ((\text{FFT-1D (K)})')) \dots\dots (1)$$

Here, I- Input Image and K- Kernel Image. For architecture of 1D/2D FFT/IFFT working starts with storing the 2D data in 1D array format as FPGA can't understand this 2D data. It works like first row elements considered first then second row and so on. After that each element is accessed row wise with column x column times and twiddle factors are found out. After that the row data is multiplied with twiddle factors. That multiplied data is accumulated column times and immediately it is stored and accumulator is reset. At this point one row is done with all the FFT calculations.

Now second row is considered and all the operations are performed. Also same procedure is repeated till all the rows are considered. Now FFT is calculated for all the rows, columns are still remaining. So now take static transpose of the 1D array and follow the whole FFT procedure for all the transposed rows. At the end the transposed version of FFT for 1D array is the result. So after again taking transpose of that array actual FFT for 1D array is completed. Same procedure is followed for second 2D data or here polynomial. After that the FFT data is multiplied with each other considering real data with real and imaginary data with imaginary one. IFFT is performed on this multiplication output. In this way the total architecture for PMM is obtained.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

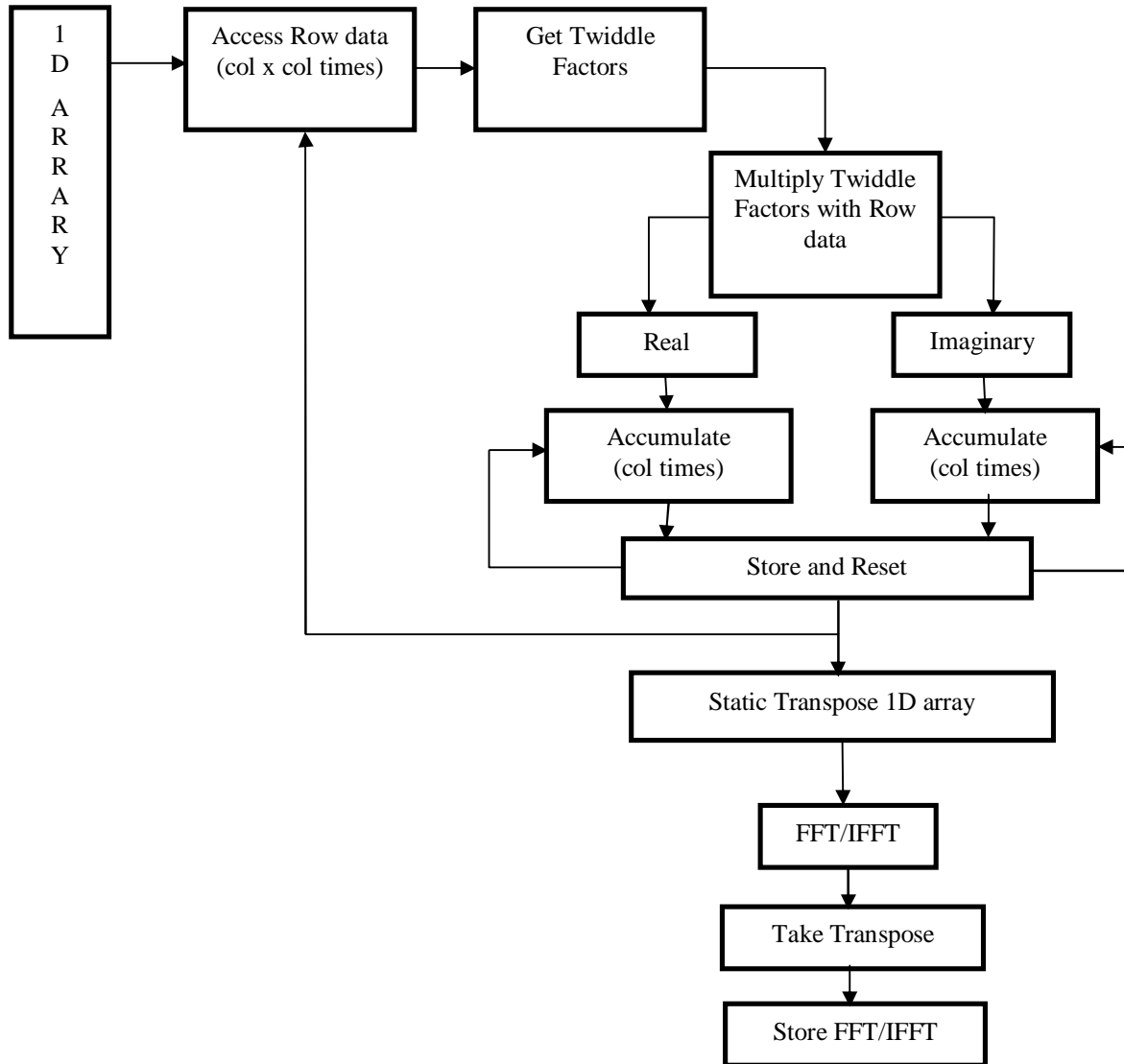


Fig. 2: Generic architecture for 1D/ 2D FFT/IFFT

The final architecture of PMM is developed in MATLAB Simulink as shown in Fig. 3. The input to this architecture in Xilinx system generator too is given through MATLAB via gateway input port. Two images i.e. input image and kernel image is fed as input and FFT are calculated respectively for both inputs. Then at the output of FFT real and imaginary part of both inputs are obtained. These are multiplied as real part of first image with real part of other and imaginary part of first image with imaginary part of other input. At the end one real component and one imaginary component are collected. These are fed to IFFT block and final filtered image i.e. sharpened; blurred or smooth image is retrieved depending on the kernel provided.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

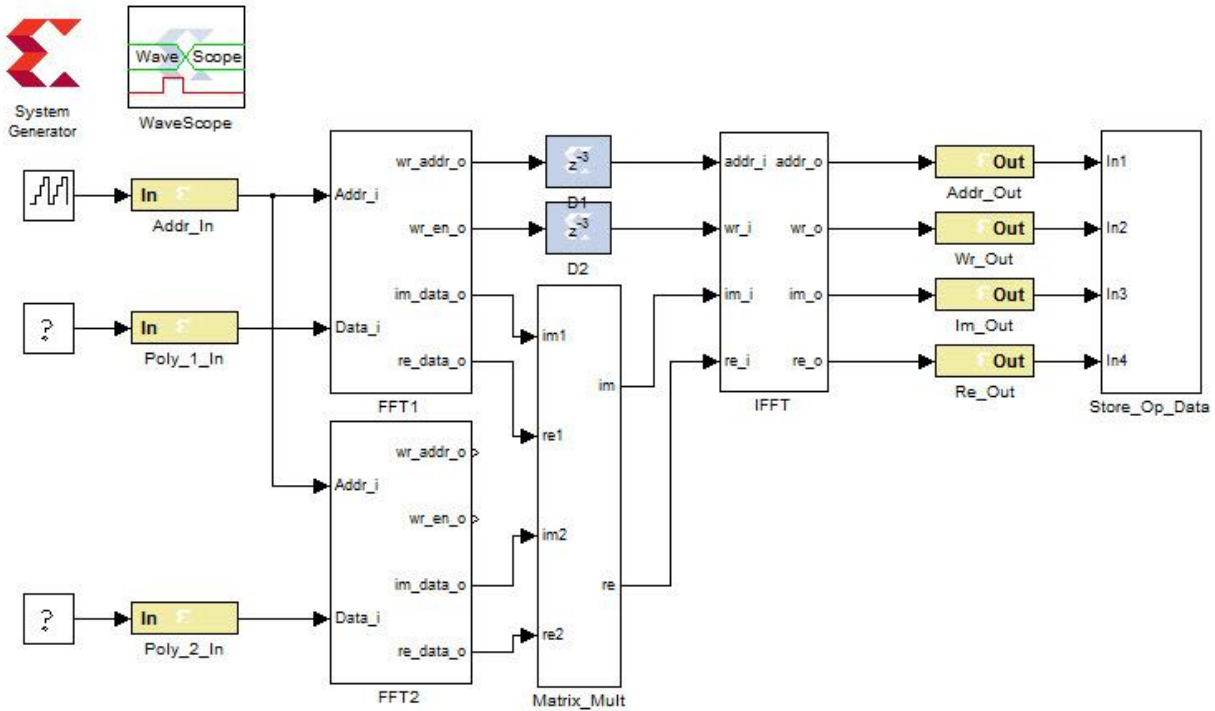


Fig. 3: Top level diagram for proposed PMM architecture using Simulink

IV. RESULTS

The results for the proposed architecture are shown in Fig. 4(a) and 4(b). Here Order of matrix (N) is varied and for every value of N Clock cycles required and mean square errors are calculated. From these results it is clear that MSE for various values of N are much small.

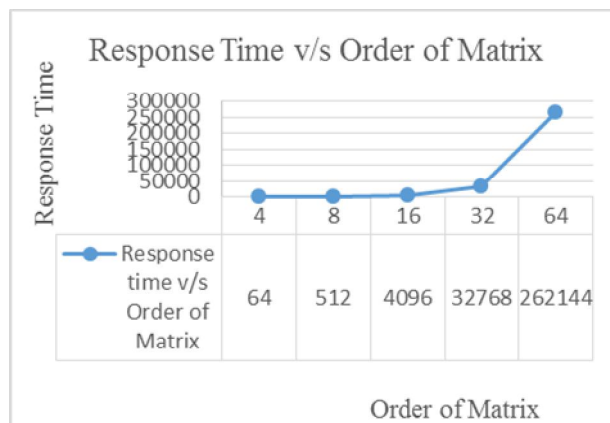


Fig. 4(a): Plot of Response time verses Order of Matrix

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

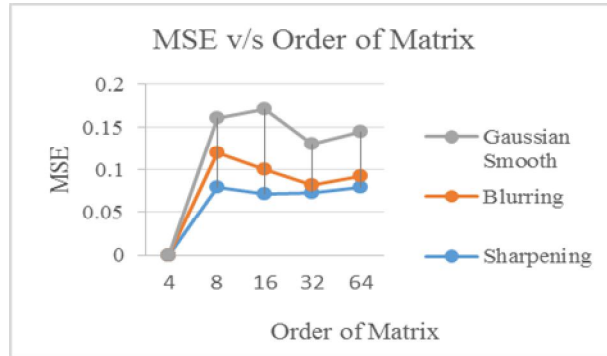


Fig. 4(b): Plot of MSE versus Order of Matrix

This PMM architecture is applied to the application of image filtering i.e. image sharpening, image blurring and image smoothing, etc. To filter any image it is to modify the pixels in an image based on some function of a local neighbourhood of each pixel. A kernel is a small matrix of numbers that is used in image convolutions. There are differently sized kernels containing different patterns of numbers produce different results under convolution. The size of a kernel is arbitrary but 3x3 is often used. This kernel matrix is to be convolved with image matrix to get filtering effects. If large size of kernel is needed zeros are padded. Here the results are shown for the proposed PMM architecture to the application of image sharpening. The value of order of matrix is kept 64. As seen in Fig. 5 the output image is sharp than the input image. Hence good quality sharpening is achieved.

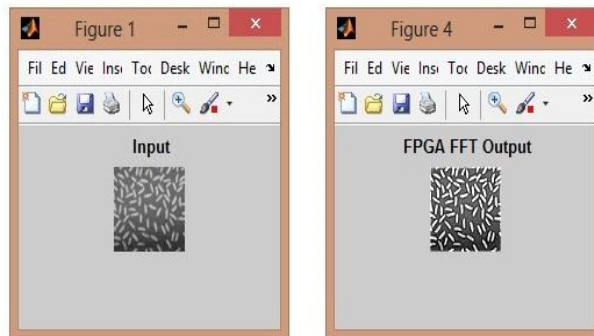


Fig. 5: Results for PMM as application to sharpening (N=64)

Table 1 states the device utilization of the HDL code generated from the system generator model for N=64. From this table it is clear that when the system generator model is converted into HDL code and simulated in Xilinx environment, the no. of slice registers used, no. of slice LUT's used, no. of Block RAM/FIFO, and no. of DSP's used are very less.

Parameters	Used	Available	Utilization
No. of slice registers	10271	32640	31%
No. of slice LUT's	17392	32640	53%
No. of Block RAM/ FIFO	61	132	46%
No. of DSP48Es	46	288	15%

Table 1: Device utilization of proposed system for N=64



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

V. CONCLUSIONS

Here, a unique architecture is proposed to compute Polynomial Matrix Multiplication of any complex matrices. Various signal processing tasks can be realized in real systems with the help of convolution that can be more effectively realized in frequency domain with polynomial matrix multiplication like sharpening; blurring, smoothing and multiple input multiple output systems, etc. FFT/IFFT is used to convert time domain matrix to frequency domain. A generic architecture is developed for computing 1D/2D FFT/IFFT of polynomial matrices. As a result device utilization, MSE and response times are computed for different order of matrices. The proposed architecture uses limited FPGA resources and less execution time. This is achieved by using Xilinx system generator tool. From the results obtained it is clear that the accuracy of output is achieved with less time. More effects are needed to reduce the utilization further and also increase the value of order of polynomial matrix.

REFERENCES

- [1] Ondrej Fialka, Martin Cadik, "FFT and Convolution Performance in Image Filtering on GPU", *IEEE Computer Society*, pp. 609-614, 2006.
- [2] Sami Kadhim Hasan, FPGA Implementations for Parallel Multidimensional Filtering Algorithm, (Newcastle University, June 2013)
- [3] Xiaohan Huang, "Fast Rectangular Matrix Multiplication and Applications", *journal of complexity*, pp. 257_299, 1998.
- [4] Matz Johansson Bergström, "Study of Convolution Algorithms using CPU and Graphics Hardware", *Master of Science Thesis in the Programme Computer Science*, University of Gothenburg, Sweden, Sept 2012.
- [5] Server Kasap, and Soydan Redif, "Novel Reconfigurable Hardware Architecture for Polynomial Matrix Multiplications", *IEEE Transactions on Very Large Scale Integration Systems*, issue 99, 2014.
- [6] John G McWhirter, "An Algorithm for Polynomial Matrix SVD Based on Generalized Kogbetliantz Transformations", *18th European Signal Processing Conference*, 2010.
- [7] J.G. McWhirter, P.D. Baxter, T. Cooper, S. Redif and J.A. Foster, "An EVD algorithm for para-Hermitian polynomial matrices", *IEEE Trans Signal Processing*, vol. 55, no. 5, pp. 2158–2169, May 2007.
- [8] Server Kasap, and Soydan Redif, "Novel Field-Programmable Gate Array Architecture for Computing the Eigenvalue Decomposition of Para-Hermitian Polynomial Matrices", *IEEE Transactions on Very Large Scale Integration Systems*, vol. 22, no. 3, 2014.
- [9] Soydan Redif and Server Kasap, "Parallel algorithm for computation of second order sequential best rotation", *International Journal of Electronics*, vol. 100, no. 12, pp. 1646-1651, 2013.
- [10] Chi Hieu Ta and Stephan Weiss, "Shortening the Order of Paraunitary Matrices in SBR2 Algorithm", *6th Conference on Information, Communications, and Signal Processing*, pp. 1-5, 2007.
- [11] J. Foster, J.G. McWhirter and J. Chambers, "Limiting the Order of Polynomial Matrices within the SBR2 Algorithm", *IMA International Conference on Mathematics in Signal Processing*, Cirencester, 2006.
- [12] Xilinx, System Generator for DSP Getting Started Guide, UG639 (v 14.2) July 25, 2012.