FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 632-638 | e-ISSN: 2347-470X

An Optimized Pipeline Based Blind Source Separation Architecture for FPGA Applications

M. R. Ezilarasan[1] and J. Britto Pari[2]

[1]*Assistant Professor. Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology, Avadi, Chennai, TN, India, arasanezil@gmail.com*
[2]*Associate Professor, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology, Avadi, Chennai, TN, India, brittopari@yahoo.co.in*

*__Correspondence:__ M. R. Ezilarasan Email: arasanezil@gmail.com

ABSTRACT- This work proposes an optimized blind source separation (BSS) architecture utilizing accumulator-based radix 4 multipliers incorporating an independent component analysis (ICA) approach. The signal observed in distinct environmental conditions degraded from its original form. ICA-based filtering is a suitable choice for recovering the desired signal components. Field Programmable Gate Array (FPGA) implementation makes the design much more attractive in high-performance. The proposed BSS-ICA architecture consists of three Random Access Memory (RAM) units and a pipeline-based accumulator radix-4 multiplier. In this work, different source signals such as sinusoidal and speech signals are considered for the analysis. The sources signals are combined with the mixing signal and the resultant signals are processed using the proposed architecture. The pipeline utilized accumulator radix-4 multiplier structure in the proposed design provides good performance in terms of speed and area. The performance of the proposed architecture is analysed using Simulink and FPGA and the results are reported. The speed of the proposed structure is improved by about 19.33% when compared with the conventional design. Likewise, the area (slices) optimization of about 27.27% is achieved for the proposed structure when examined with the existing approach. Hence, the proposed architecture separates the designed signal component with a lesser area and high speed.

Keywords: BSS, FPGA, ICA, Radix-4 Booth Algorithm.

ARTICLE INFORMATION
Author(s): M. R. Ezilarasan and J. Britto Pari;
Received: 14/07/2022; **Accepted**: 30/08/2022; **Published**: 18/09/2022;
e-ISSN: 2347-470X;
Paper Id: IJEER220727;
Citation: 10.37391/IJEER.100336
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-10/ijeer-100336.html

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

Digital Signal Processing plays indispensable role in most of the applications including communication, biomedical and noise cancellation and equalization applications. In general, the raw source signals are impure in nature as it contains different noise components. Example of such noise include a recorded signal of a human walking in a lane contain combined sounds of vehicles, birds, animals, air along with the footsteps. In such a situation, it is a challenging task to record a noiseless signal. The conversation between group of people is monitored using a microphone and the discriminative voice signals are obtained using Blind Source Separation (BSS). Independent component analysis approach makes effective blind source separation [1]. BSS is a technique of segregating n different desired source signals $S = (S_1, S_2,....S_n)^T$ from a set of m noise mixed components $X= (X_1, X_2,....X_m)^T$ wherein the characteristics of a mixed signals are unexplored. In specific, the desired speech signal is acquired from the real time noisy speech signal. Optimal algorithms and architectures are needed to address this problem. The main motivation of this work is to incorporate cocktail party problem in order to regain an original voice signal from noisy environment. The primary approach for evaluating BSS is to employ an ICA algorithm [2][3]. This method is used in processing of Biomedical signals such as electroencephalogram (EEG), electrocardiogram (ECG) signals and these signals are measured using sensors and instruments. These measured EEG and ECG signals is corrupted by the eye activity and muscle noise [2] [3] and ICA algorithms plays a major role in rectifying this problem. Most of the research works focused on processing of signals using ICA. Despite, this algorithm is used for processing EEG and ECG signals [4] [5]. Nonetheless, ICA can be implemented in Xilinx to expedite operations that require complex functions such as vectors, matrices and multiplication operation. Plenty of literature reviews addresses the implementation of ICA in real time applications. The FPGA-based design of for Multispectral analysis of image processing [6] is accomplished using ICA and part of the signal processing involves MATLAB simulation. Parallel ICA algorithm [7] is used for curtailing the dimension of images in which the FPGA implementation is performed using Tool Command Standard such as HDL code from legitimate MATLAB code. Noise cancellation is achieved using FPGA based ICA design [8]. In batching mode training, Xilinx is utilized to build an impartial components computational model for discontinuous mixture and this model is kept throughout the FPGA's storage [9]. ICA is adopted to segregate independent signal sources from mixed signals by measuring non-Gaussianity [10][11]. The source signals can split from the mixed signals by counting [12][13][14][15]. Non-Negative factorization can be applied for blind source separation which aims to factorize a matrix m into two matrixes. But it cannot be

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 632-638 | e-ISSN: 2347-470X

applied for higher order statistics. Based on the number of signal sources and number of sensors used for detection, the BSS is classified into three types of mixing namely over determined mixing, determined mixed and unpredictable mixing . Here, the signal sources are denoted as S1, S2 and S3 and sensor detected signals are specified as X1, X2 and X3. If the signal source (S1, S2) is lesser than the total number of detected sensor signal (X1, X2, X3), then the resulting signal component is specified as over determined mixing. Similarly, if the number of source signal (S1, S2) and the number of detected sensor signals (X1, X2) are identical, then it indicates determined mixing. Further, having a greater number of input signals (S1, S2, S3) when compared to the number detected sensor signals (X1, X2) results in unpredictable mixing.

The objective of BSS is to recover different source signals from the linear mixed signals with the least amount of priority information or to reconstruct the desired signal from the observed noisy signal. This work proposes the FPGA implementation of ICA algorithm which helps in recovering the appropriate speech signal from the mixed signal observed in noisy circumstances. The performance of the suggested algorithm is also verified using MATLAB Simulink model and the resultant output signal is a pure speech signal with minimal noise. FPGA realization is attractive in recent days due to its faster clock rates, processing considerable number of events per second, low cost, parallel processing and capacity to interface different applications. Model-sim generator is a kind of DSP tool which designed for performing required simulation using MATLAB and Simulink. With this observation, this work outlines a feasible way of constructing a legitimate BSS solution utilizing FPGAs that would not necessitate a full knowledge of equipment architecture and descriptive technologies. The availability of simulation model generation in DSP uses simulation graphical interface in MATLAB, thus eliminates the need for HDL language coding.

This paper is organized as follows: *Section I* presents introduction and survey related to BSS. *Section 2* discusses the details of Blind source separation; *Section 3* presents ICA algorithm and discusses the proposed pipelined architecture in which Accumulator based radix 4 booth algorithm is applied to reduce the number of multipliers. *Section 4* shadows the results and discussion of proposed work and *Section 5* presents conclusion and future work to be carried out and in *Section 6* future work and limitations is discussed.

2. BACKGROUND OF ICA ALGORITHM

The general block diagram of ICA based signal processing is shown in *Figure 1*. In this source signal is mixed with mixed matric and decomposed using ICA algorithm.

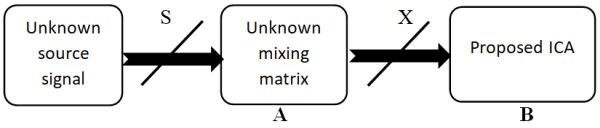

Figure 1: Mixing and unmixing model

This section presents the concepts of ICA algorithm used for segregating BSS. Although, ICA is developed earlier, still it find its potential usage in several applications. In general, this ICA is used to address the cocktail party problems. Before applying ICA algorithms following assumption to be carried out over the source signal, first one is the source signals should not dependent with one another, it should be independent. Second, this independent component components should follow non-Gaussian distribution third assumption is number of observers or sensors should be equal to number of sources. Example if the music is the source signal, then microphone should be equal to the source signal, by this assumption the mixing matrix will be square. After finding the matrix, we can estimate its inverse W by $s= A^{-1}x = Wx$ where W is the demixing matrix and x is the observed sequence. However, it is quite suitable for many applications such as signal processing, image processing, artificial neural networks, statistics, and information theory. The two source signals s1 and s2 are represented in terms of matrix S = $\begin{bmatrix} s1 \\ s2 \end{bmatrix}$ and the mixing signal M= $\begin{bmatrix} a11 & a12 \\ a21 & a22 \end{bmatrix}$ is represented in terms of mixing coefficients a11, a12, a21 and a22. The resultant mixed output signal is denoted as X= $\begin{bmatrix} a11s1 & a12s2 \\ a21s1 & a22s2 \end{bmatrix}$. Here X1=a11s1+a12s2 is a first mixed signal which is the overage of the two transmitted signals, graded (s1 and s2). The range between both the transmitted signal and the sensor device signal is adjusted to create another mixed signal combination (x2), which is computed as x2=a21s1+a22s2. Because the two sensing devices employed to sense these signals are in separate places, the two mixing coefficients a11 and a12 differ from the coefficients a21 and a22. As a result, each sensor senses different mixtures of source signals. The representation of the two mixed signal components is given in *equation (1)*.

$$X=\begin{bmatrix} x1 \\ x2 \end{bmatrix}=\begin{bmatrix} a11s1 & a12s2 \\ a21s1 & a22s2 \end{bmatrix} =A.s \qquad (1)$$

If the number of source signals increases, the mixing coefficient matrix A is also increases since each mixing signal has a different characteristic such as varying distance, environment, and microphone capacity. The general expression for mixed signal output X with n number of source signals are given in equation (2).

$$X=\begin{bmatrix} x1 \\ \vdots \\ xn \end{bmatrix}, A = \begin{bmatrix} a11 & . & . & a1n \\ \vdots & . & . & \vdots \\ an1 & . & . & ann \end{bmatrix}, s = \begin{bmatrix} s1 \\ \vdots \\ sn \end{bmatrix} \quad (2)$$

3. PROPOSED PIPELINE BASED ICA ARCHITECTURE

In the previous literatures and researches it has been with used the binary multipliers and adders to mixing and de-mixing. In this proposed work pipelined based architecture is implemented with radix 4 multipliers to get optimized FPGA based ICA architecture utilizing pipelining scheme. The proposed pipeline-based ICA architecture is shown in *figure 2*. The structure consists of three number of Random Access Memory (RAM) to store the sampled sinusoidal signal with different frequencies. The stored signals in RAM are given as an input to the

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 632-638 | e-ISSN: 2347-470X

corresponding register which provide a digital signal. Further, the obtained signal is combined with the mixing coefficients. Then the output of the register is given to the radix-4 multiplier which minimizes the computational complexity, curtails the energy consumption and enhances the system throughput rate. In this proposed ICA architecture, three different frequency component based sinusoidal input signals of 800Hz, 700Hz, and 600Hz are examined and upon mixing, the resultant mixed signals are processed using pipelining and efficient accumulator-based radix-4 booth multiplier in order to achieve an efficient implementation. The mixed signal along with the corresponding mixing coefficients is given in *equation (3)*. The simulation of this ICA algorithm is done using verilog code and its hardware realization is carried out in ModelSim tool.

$$X = \begin{bmatrix} X1 \\ X2 \\ X3 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{Sin}(2\pi 800t) \\ \mathrm{Sin}(2\pi 700t) \\ \mathrm{Sin}(2\pi 600t) \end{bmatrix}$$

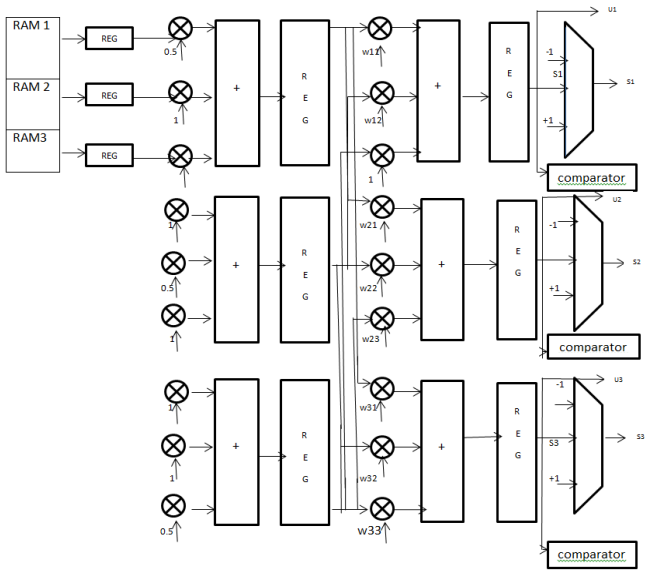

Figure 2: Proposed ICA architecture

3.1 Accumulator Based Radix 4 Multiplier

High speed multiplier design enhances parallelism and minimizes the number of successive steps. The processing speed is enhanced by using Booth's bit pair recoding approach that reduces the number of incomplete products. For n-bit data, the conventional radix-2 booth algorithm involves n-number of partial products, whereas the radix-4-bit pair recoding approach involved n/2 number of partial products. In bit pair recoding, the multiplier coefficients are recoded in terms of three bits and the partial products are computed at the same time. Hence, the switching activity is diminished to a considerable extent. The radix-4 nxn bit multiplication is accomplished with the help of accumulator unit which generate n/2 number of intermediate results throughout this research. The partial products are reorganized and accumulation is performed using carry look-ahead adder (CLA). The multiply-accumulate unit (MAC) performs multiplication and accumulates the result in every clock cycle. The accumulator-based radix-4 multiplier increases the speed of operation by reducing the number of partial products. In particular, the MAC architecture performs concurrent computation of partial products using booth radix-4 recoding scheme and then the results are combined and accumulated using a high speed CLA. Further improvement in speed of operation is obtained using pipelining scheme by inserting relevant registers in the appropriate stages. The multiple functions are combined and operations are carried out consecutively to embellish the speed of architecture.

The main goal of the radix-4 multiplier is to minimize the computational complexity and these reduced complexity multipliers are extensively used in neural networks. Hence, the radix-4 multiplier is preferred for high-speed applications. Multipliers mainly provide significant impact on the speed of high-performance systems in digital signal processors. If the architecture employs large number of multipliers, then the performance goes down. Several number of multiplier designs are reported in the literatures offers good performance in terms of speed, power and area. A binary multiplier carries out multiplication of two binary numbers. Efficient multiplier design is the challenging task which substantially improve the performance of the system. The elementary approach to carry out multiplication is to use a single two input adder. This single adder produces a partial product upon multiplication. The partial products are attained by multiplying multiplicand with multiplier (AND processing) and shifting the output according to the multiplier bit position. The number of generated partial products depends on the input number of bits. In the case of large number of partial products, speed becomes the major concern. The booth approach provides one such alternative for increasing the speed of operation. In the conventional radix-2 algorithm, zero is appended to the multiplier before shattering this into two groups of two bit each. As a result, the first pair contains additional blank and multiplication LSB bits. Then the next pair overlap with every preceding pair forming incomplete products which requires large number of addition operation which does not reduce the partial products as well. Hence, the radix-2 multiplication process becomes unattractive in many high-performance applications. The radix-4 algorithm provides a solution to address this issue. *Table 1* shows the partial products of Radix-4 Booth algorithm. In radix-4 multiplication, recoding is done based on group of three bits which forms a cluster.

Extend the sign-up to the product size after you've generated all of the incomplete items. The sign extension necessitates a huge number of complete adders, resulting in a long delay.

Table 1: Radix-4 Booth algorithm

Federation	Fractional Product
000	0* Multiplicand
001	1* Multiplicand
010	1* Multiplicand
011	2* Multiplicand
100	-2* Multiplicand
101	-1* Multiplicand
110	-1* Multiplicand
111	0* Multiplicand

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 632-638 | e-ISSN: 2347-470X

4. RESULTS AND DISCUSSION

4.1 Simulation Results Using Sinusoidal Input

The proposed ICA algorithm is tested using Simulink by considering three sinusoidal signals with a frequency of 800Hz, 700Hz, and 600Hz respectively. Then, these signals are combined with a distinct coefficient based mixing signal and is shown in *figure 3.*

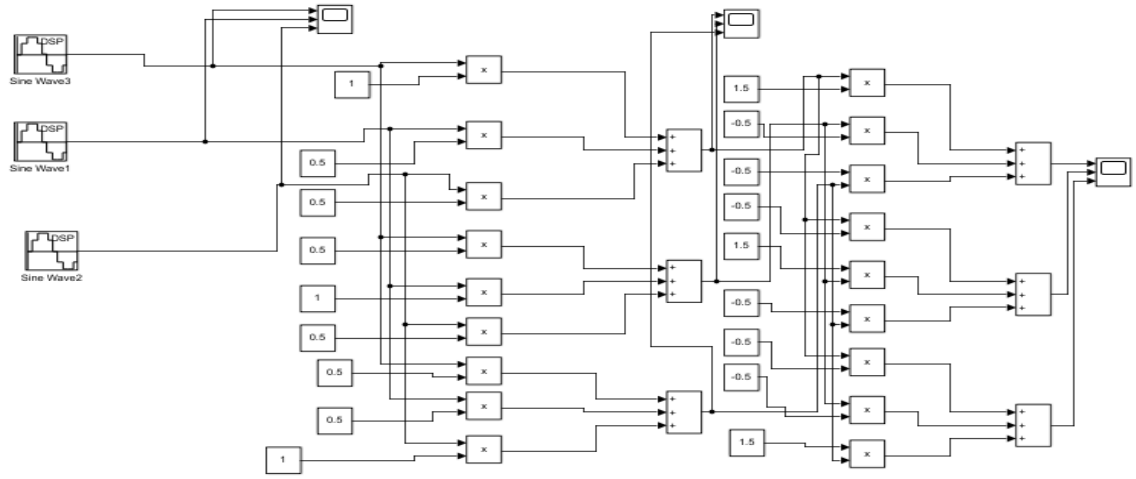

Figure 3: The design of the proposed ICA architecture in Simulink

Three sinusoidal discrete time input signals with different frequencies are taken as input source signals, and then these signals are mixed with mixer constant and the output of the ICA architecture is viewed by the scope. The generated sinusoidal mixed signals having 20Hz, 25Hz and 40Hz frequencies are illustrated in *figure 4*. Likewise, the trail output of mixed signals having 20Hz, 25Hz and 40Hz frequencies are shown in *figure 5*. The segregated resultant individual output signals are given in *figure 6*.

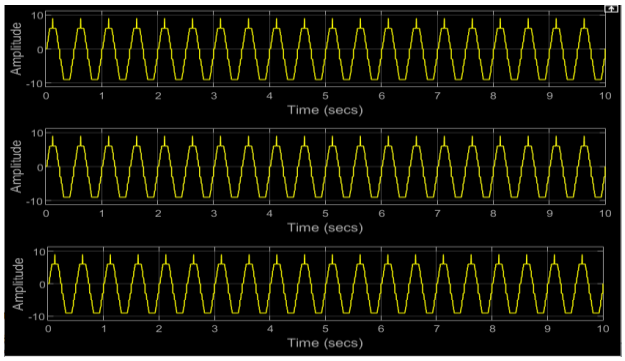

Figure 4: The sinusoidal mixture signals of different frequencies

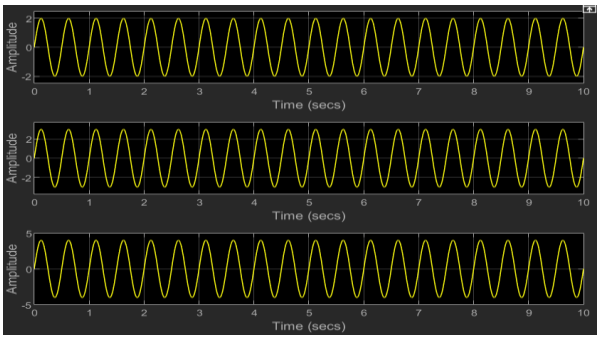

Figure 5: Trial results of Mixed-signal

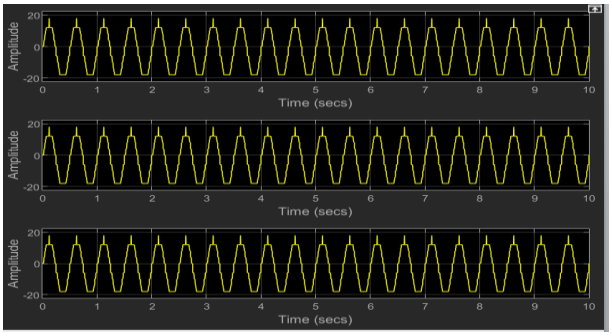

Figure 6: Trial results of independent signal

4.2 Simulation Results Using Audio Signals

Simulation is also done using audio signals. Two different audio signals are taken as input which is named as speech 1 wave and speech 2 waves and are shown in *figure 7*. Similarly, the mixture signal in pictorially given in *figure 8.* The audio signals are mixed by these mixture signals and the error signals are obtained and is shown in *figure 9*. Further, the error signals are given as an input to the proposed design. This error signal is the deviated signal from the observed signal and the input signal. Also, the resultant independent component of the proposed design is depicted in *figure 10*.

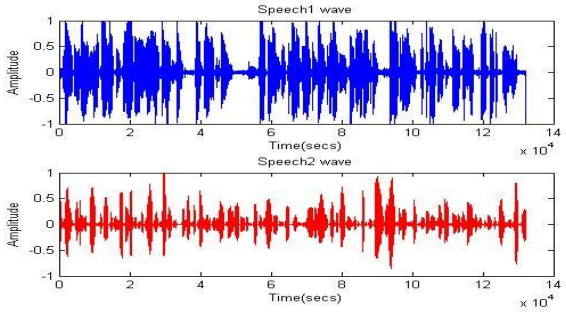

Figure 7: Two different speech signal

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 632-638 | e-ISSN: 2347-470X

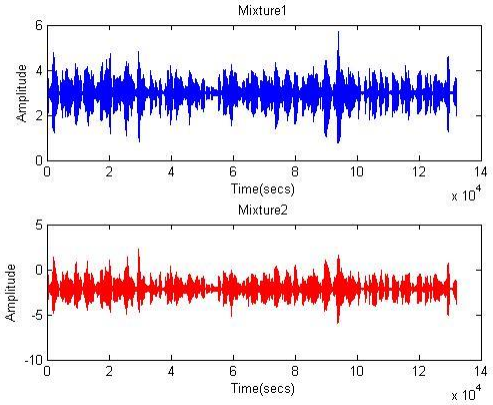

Figure.8. Two Mixture signals

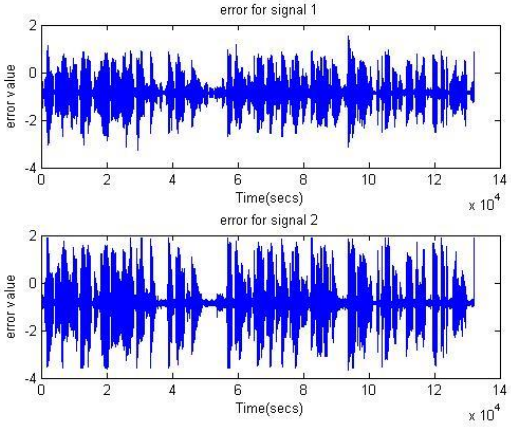

Figure 9: Error signals

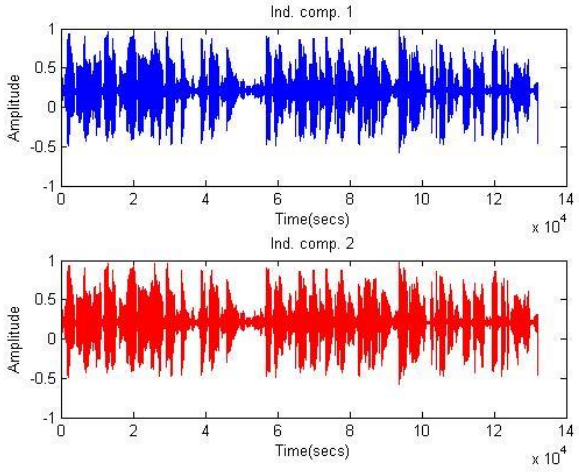

Figure 10: Two Independent components

4.3 FPGA Implementation

The proposed ICA for BSS has been simulated using Verilog and tested with Xilinx FPGA devices. The simulation waveform of the ICA is depicted in *figure 11*. In *figure 11*, s1, s2 and s3 are the sinusoidal inputs which are obtained from the sine lookup table. These sinusoidal inputs are processed through a mixer circuit and their respective outputs are x_1, x_2, and x_3. The mixer outputs again given to the de-mixing circuit and the respective outputs are obtained as u1, u2, and u3.

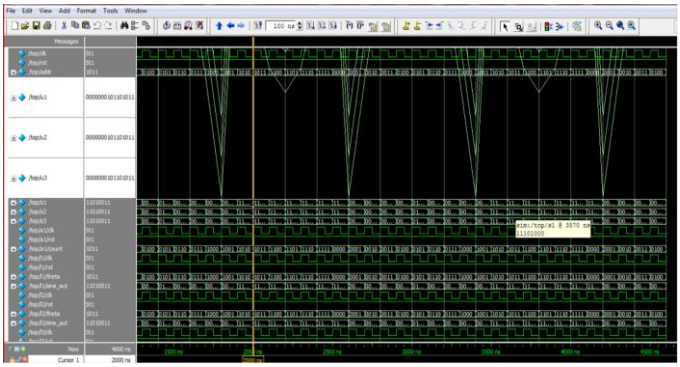

Figure 11: FPGA Implementation-Simulation Waveform of ICA algorithm

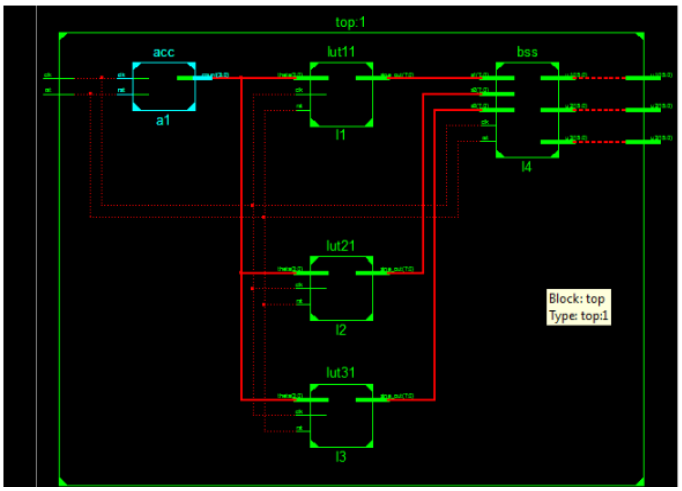

Figure 12: RTL Schematic ICA Algorithm

The RTL schematic of the proposed architecture is illustrated in *figure 12*. From this figure, it is observed that the figure contains three LUTs responsible for creating a sinusoidal wave for signal processing, and this input is given to the BSS, which utilizes ICA. There is also a feedback path in the BSS, which is given as input to the accumulator a1. Xilinx Virtex -6 XC6SL45 FPGA is used to implement the ICA algorithm.

The proposed BSS architecture is compared with existing in terms of area and speed.

The device used for analysis and synthesis is spartan- 6 XC6SL45 FPGA. The speed analysis of the proposed architecture is depicted in *figure 13*. It is observed that the proposed BSS-ICA based pipeline architecture achieves 19.33% speed enhancement when examining with the conventional ICA design. Likewise, the area (number of slice) analysis is given in *figure 14*. Proposed BSS realization employs 27.27% reduction in number of slices over the existing realization. This is achieved by reducing the number of fractional products with the help of accumulator-based radix 4 algorithm.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 632-638 | e-ISSN: 2347-470X

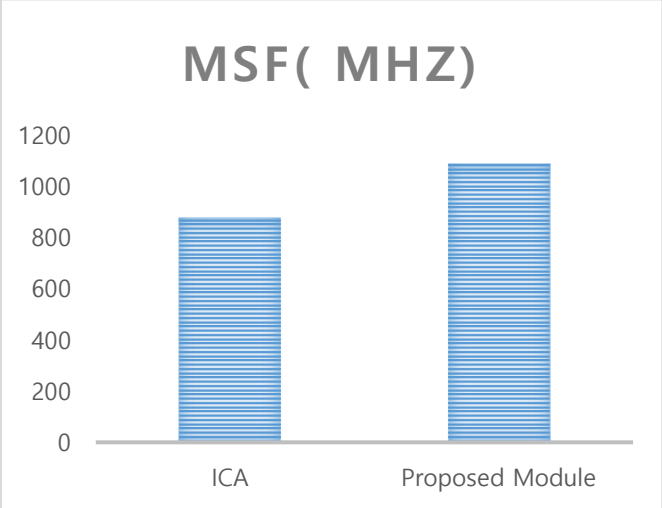

Figure 13: Speed Analysis

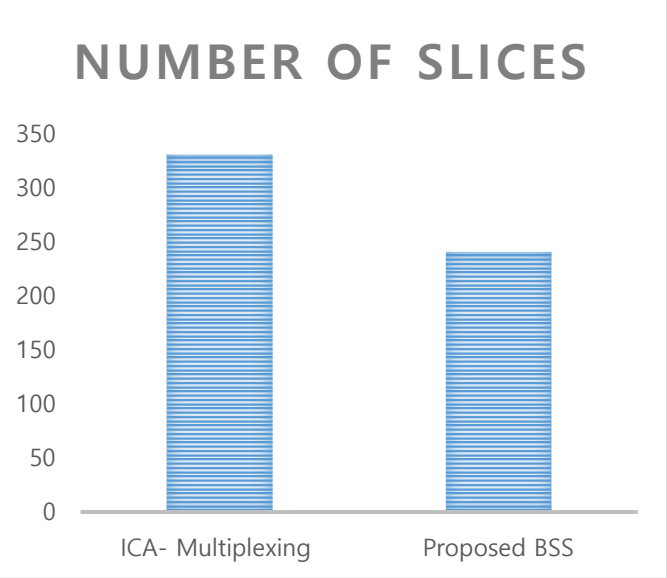

Figure 14: Area (slices) Analysis

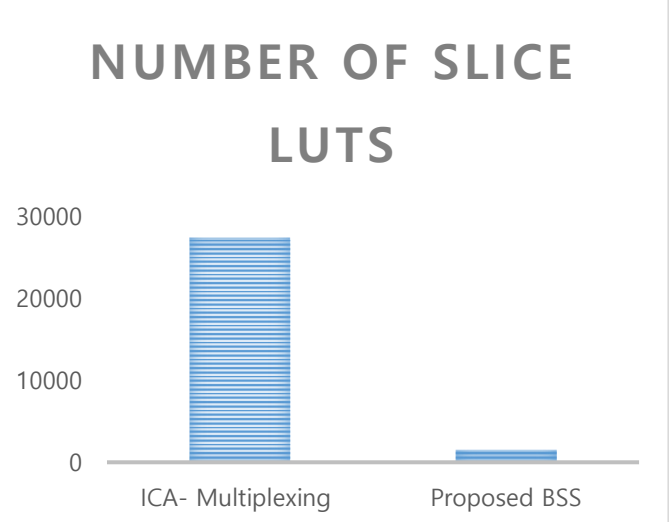

Figure 15: Area (LUT) Analysis

Further, the area (number of slices LUTs) analysis is displayed in *figure 15*. The proposed BSS architecture achieves a 94.5% of reduction in number of slices LUTs due to accumulation and partial product reduction of multiplier architecture.

5. CONCLUSION

In this research work, an optimized BSS-ICA architecture for segregating the desired signal from the noise corrupted signal is proposed. ICA approach effectively separates the needed signal. To achieve optimized performance, an accumulator-based radix 4 multiplier design and pipelining scheme is adopted. The synthesis and analysis are carried out using spartan- 6 XC6SL45 FPGA. The results show that, the proposed architecture offers good performance in terms of number of slices, number of slices lookup tables and speed. The speed of the proposed structure is enhanced about 19.33% when correlating with the conventional structure. Similarly, the area(slices) curtails of about 27.27% is achieved for the proposed structure when examining with the existing design. The future enhancement of the research work is, the multipliers used should be increased in order to increase the throughput. Booth algorithm can be used to reduce the partial products and increase the speed in view of convolutions. In future work ICA can be implemented with Adaptive filter where the mixed signals is given as input to adaptive filter and then observed signal is taken as output.

REFERENCES

[1] Vinita Singh.," FPGA Implementation of Blind Source Separation using a novel ICA Algorithm" IEEE International conference on consumer Electronics., 2017

[2] P. Berg and M. Scherg, "A multiple source approach to the correction of eye artifacts," Electroencephalogr. Clin. Neurophysiol., vol. 90, pp. 229–241, 1994.

[3] R. Vigario, "Extraction of ocular artifacts from EEG using independent component analysis," Electroencephalogr. Clin. Neurophysiol., vol. 103, pp. 395–404, 1997.

[4] T. W. Lee, Independent Component Analysis: Theory and Applications. Boston, MA: Kluwer, 1998.

[5] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," Neural Network., vol. 13, pp. 411–430, May 2000.

[6] A. Nordin, C. Hsu, and H. Szu, "Design of FPGA ICA for hyperspectral imaging processing," in Proc. SPIE Signal Image Process., Mar. 2001, vol. 4391, pp. 444–454.

[7] H. Du and H. Qi, "An FPGA implementation of parallel ICA for dimensionality reduction in hyperspectral images," in Proc. IEEE, 2004, pp:3257-3260.

[8] C. M. Kim, H. M. Park, T. Kim, Y. K. Choi, and S. Y. Lee, "FPGA implementation of ICA algorithm for blind signal separation and adaptive noise canceling," IEEE Trans. Neural Netw., vol. 14, no. 5, pp. 1038–1046, Sep. 2003.

[9] A. R. Omondi and J. C. Rajapakse, "Neural networks in FPGAs," in Proc. Int. Conf. Neural Inf. Process., Nov. 2002, vol. 2, pp. 954–959.

[10] L.Neo Oliva moreno "Implementation of Infomax ICA Algorithm for blind source separation" Electronics, Robotics and Automotive mechanics conference, 2008

[11] Hyvarinen., "independent component analysis: Recent advances" Phil. transaction 2013

[12] Hyvarinen., J Karhunen "Independent component analysis" vol 46,John Wiley & Sons,2004

[13] S. Mirzaei,Y. Norouzi, andH.Van Hamme, "Two-stage blind audio source counting and separation of stereo instantaneous mixtures using Bayesian

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 3 | Pages 632-638 | e-ISSN: 2347-470X

tensor factorisation," IET Signal Process., vol. 9, no. 8, pp. 587–595, 2015.

[14] B. Laufer-Goldshtein, R. Talmon, and S. Gannot, "Source counting and separation based on simplex analysis," IEEE Trans. Signal Process.,vol. 66, no. 24, pp. 6458–6473, Dec. 2018.

[15] S. Araki, T. Nakatani, H. Sawada, and S. Makino, "Blind sparse source separation for unknown number of sources using Gaussian mixture model fitting with dirichlet prior," in . IEEE, 2009, pp. 33–36.

[16] B. Murali Krishna, Chella Santhosh, S.K. Khasimbee (2022), FPGA Implementation of High-Performance s-box Model and Bit-level Masking for AES Cryptosystem. IJEER 10(2), 171-176. DOI: 10.37391/IJEER.100221.

[17] Kibum suh (2022), FPGA Design of Real Time Hardware for Face Detection. IJEER 10(2), 202-206. DOI: 10.37391/IJEER.100226.

[18] Guard Kanda and Kwanki Ryoo (2022), Design of an Integrated Cryptographic SoC Architecture for Resource-Constrained Devices. IJEER 10(2), 230-244. DOI: 10.37391/IJEER.100231.