Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 06, June-2022

Independent Component Analysis for Biomedical Signal Separation using Lambert's Continued Fraction

Shehnas S
PG student,
VLSI & Embedded Systems, ECE Department
TKM Institute of Technology, Karuvelil P.O, Kollam,
Kerala-691505, India

Anas A. S
Assistant Professor,
ECE Department
TKM Institute of Technology, Karuvelil P.O, Kollam,
Kerala-691505, India

Abstract:- **The separation of weak signals from multiple sources contaminated with artifacts and noise is one of the main challenge to the researchers, signal sources such as heart, endocrine system, brain etc. so their analysis becomes more important for the researchers. Biomedical signal analysis is also important for medical diagnosis and their treatment. For revealing new information about the brain and body, many groups are actively engaged in exploring both the potential of blind signal separation and signal deconvolution and therefore the application of Independent Component Analysis (ICA) to biomedical signals is rapidly expanding in area of research. The project aims to design the extend InfoMax independent component analysis (ICA) algorithm which can separate the super-gaussian signals. In order to substantially reduce the circuit area, the circuit utilizes the time sharing matrix multiplication array (MMA) to realize a series of matrix multiplication operations and employs Lambert's continued fraction to calculate the hyperbolic function tanh(u) and the overall circuit can be successfully applied to separating mixed medical signals into independent sources.**

Keywords: Independent Component Analysis (ICA), Blind Signal Separation, Lambert's continued fraction.

I. INTRODUCTION

For the separation of multivariate signals into subcomponents, independent component analysis (ICA) is used in signal processing such that one subcomponent is assumed to be a non- Gaussian signal and is statistically independent from each other. The most popular algorithm developed in recent years to solve for blind source separation (BSS) problems. A direct separation of a mixed signals with unknown coefficients of mixture is possible and is the main characteristics of the algorithm.

Artificial neural networks (ANNs), usually simply called neural networks (NNs), are computing systems that consists of several processing elements that receive inputs and delivers outputs based on their activation function, inspired by biological neural networks that constitute animal brains. An ANN is based on connected units or nodes called artificial neurons, which is related to the neurons in a biological brain. The synapses in a biological brain are the connections which can transmit a signal to other neurons. An artificial neuron receives a signal, processes it and signals the neurons connected to it. The output of each neuron is computed by some non-linear function of the sum of its inputs and the signal at a connection is a real number. The connections are called edges. The Neurons and the edges typically have a weight that adjusts as learning proceeds. The strength of the signal at a connection is determined by the weight. If the aggregate signal crosses the threshold, a signal is sent such that the neurons will have a threshold. Typically, neurons are aggregated into layers. Different transformations are performed on their inputs by different layers. After traversing the layers multiple times signals travel from the first layer (the input layer), to the last layer (the output layer). The conceptual derivation of biological neurons are ANNs which are composed of artificial neurons and the neurons receive the input and a single output from each artificial neurons.

A VLSI implementation of independent component analysis using cordic engine developed in recent years for biomedical signal separation [1]. The ICA algorithm becomes important in the field of the digital signal processing (DSP) [2], image processing [3], [4], and biomedical signal analysis [5], [6], such as electroencephalogram (EEG), electro-corticography (ECoG), magneto-encephalography (MEG), and electrocardiogram (ECG) analyses because the algorithm allows direct separation for a number of mixed signals with unknown coefficients of mixture. Biomedical signal and its related applications requires fast computation for real-time detection or processing. In this aspect, there are a number of previous researches in literature having been conducted on enhancing the computational speed of the ICA algorithm [7]–[9]. Regarding the real-time computational applications, the information maximization ICA, also alternatively referred to as the InfoMax ICA, is proposed to be suitable for real-time applications [7], and further, the extended InfoMax ICA algorithm is able to separate the source signals which are sub and/or super-Gaussian [8]. In addition, the fast ICA (FastICA) that measures non-Gaussian signal using kurtosis to find the independent sources, is proposed to improve the efficiency of ICA [9].

Recently, FPGA-based ICA implementations have been presented due to the high-speed characteristic and flexibility of FPGA platforms [10]–[15]. The FPGA-based ICA algorithm was implemented into Insight Virtex-E prototype board achieving the 20 MHz clock frequency [11] designed using MATLAB Simulink software. The Parallel ICA (pICA) presented several ICA-related reconfigurable components (RCs) that are developed for reuse to improve the computation speed [13]. In [14], an efficient hardware ICA architecture was proposed and implemented into the Virtex-E FPGA. Although this architecture provides a good balance between the hardware requirements and separation performance, the hardware description language (HDL) is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 06, June-2022

generated by MATLAB Simulink software, thus lacking in a degree of flexibility and effectiveness. Instead, another work as proposed in [15] implemented the FastICA algorithm on an FPGA using the pipeline technology with hand coding in HDL, and thus, the effectiveness of the FastICA may be expected. The application-specific integrated circuits (ASICs) are used to implement the ICA algorithm due to the small-area, highspeed, and low-power characteristics, especially for wearable applications [16]–[21]. A FastICA-based pICA algorithm on ASIC using standard library cells was presented in [17].Another example of an energy-efficient FastICA algorithm implementation with an early determination scheme for eight-channel EEG signal separation was presented in [18].
Figure 1 shows the tanh function,a non-linear function:

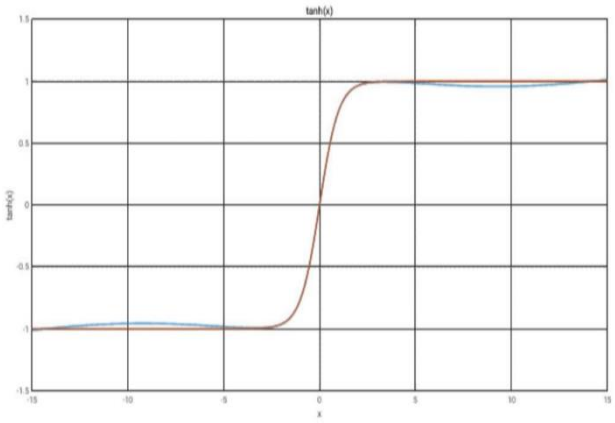

Figure 1: tanh function

Literature ranging from the simplest step and linear approximations to more complex interpolation schemes, multiple implementations of hyperbolic tangent have been published. Different tanh implementation methods and the motivation behind this paper is discussed in this section.
Storing the values of the function in a lookup table (LUT) and approximating the output with the lookup table value for the nearest input is the simplest implementation. To balance the tradeoff between accuracy and area is challenging if the range is divided in equal sub-ranges because the function is non-uniform. To address this issue, range addressable lookup table has been proposed by Leboeuf et al. [19]. Depending on the variability of the function, the step size is varied to reduce the size of LUT without disturbing the accuracy. Using a two-step LUT is a very different variation of this method, the coarse estimation and the finer estimation. Namin et al. use this method but instead of an LUT, they use a combination of linear and saturation values for coarse approximation [20].
Zamanlooy et al. take advantage of the tanh function being an odd function and divide it in three ranges based on its basic properties; pass region, processing region and saturation region [21]. The data is simply shifted and made constant in the pass region and in the saturation region respectively, therefore the hardware is optimized according to each region. In processing region, data is mapped from the input by simple bit-level mapping (i.e. the combinatorial logic).
To reduce the error, the function value can be interpolated by piecewise linear (PWL) interpolation. The function value is stored in an LUT for known values and from these values, the function is interpolated for intermediate input values [22].
Adnan et al. have approximated the tanh function by Taylor series expansion [23]. The function is more accurately computed for smaller values of inputsand the accuracy varies across the range of the input. Moreover, if the number of terms in Taylor series are increased from three to four, improvement is just 2x where the error was large while it is 10x where the error was already small.
Abdelsalam et al. have used the DCT (discrete cosine transform) interpolation filter (DCTIF) for tanh approximation [24]. Like [21], they also divide the tanh function in three regions and use DCTIF for approximation in processing region. Higher accuracy is achieved when compared to all other methods. However, it requires huge memory for storing the coefficients.

Researchers have also explored rational interpolation methods. Z. Hajduk [25] has discussed the hardware implementation of tanh using Padé Approximant. Similarly, Lambert's continuous fraction is also used for the rational function approximation of hyperbolic tangent. [26]
Various implementations have been published in the literature including complex and require huge resources with fixed point data such as deep learning. Even though, rational approximations are computationally complex as they require a divider, they are worth exploring for proper comparison. Newton-Raphson method can be applied for the reciprocal computation to implement the divider [27]. Moreover, there are methods for fast oral calculation of various trigonometric and exponential function such as the one published by Ron Doerfler [28]. This method is quite interesting and can be applied to the hardware implementations. This paper explores a hardware implementation by adopting this method and making necessary changes to make it hardware friendly.
This work aims to design a VLSI implementation for the extended InfoMax ICA algorithm which can separate the super-Gaussian signal sources. The proposed ICA employs the matrix multiplication array (MMAs) and Lambert's continued fraction for achieving a small-area and high-throughput rate in the VLSI design. To demonstrate the performance of separation, the mixed medical signals can be successfully separated into independent source signals through the proposed ICA.
This paper is organized as follows. In Section II, the mathematical derivation of the extended InfoMax ICA algorithm is given. The proposed ICA architecture that includes the Matrix Multiplication Array and the tanh computation using Lambert's continued fraction is discussed inSection III. Performance and discussion are presented in Section IV. Section V includes a brief conclusion.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 06, June-2022

II. ALGORITHM

The InfoMax ICA algorithm is proposed to separate the super-Gaussian signal sources. The extension of InfoMax algorithm as presented in [8] is able to blindly separate mixed signals into a number of independent signal sources with sub and super-Gaussian distributions. The problem formulation is described as follows. Given the received mixed signals x that are generated from the original independent source signals s, which are unmixed, through the generating matrix A, the algorithm aims to find a recovered matrix W that can separate the received mixed signals x into a number of estimated independent source signals u, so the separated signals u may identify the original signals s as closely as possible. Figure 2 shows the network of the mixed and unmixed signals.

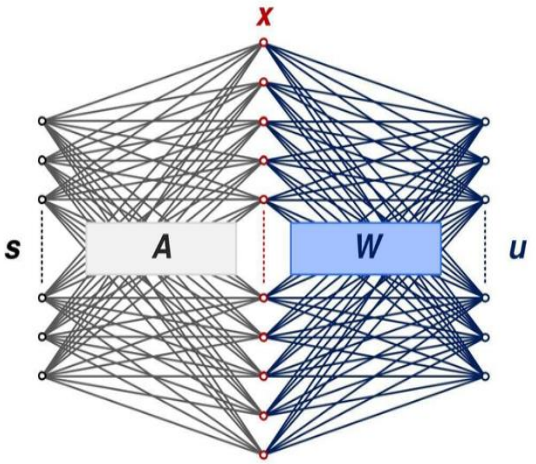

Figure2: Example network of mixed and unmixed signal

$$\mathbf{x} = \mathbf{As} \text{ ---------- (1)}$$
$$\mathbf{u} = \mathbf{Wx} \text{ ---------- (2)}$$

It should be noted that according to the extended InfoMaxICA algorithm [8], the algorithm requires the number of mixed signals to be the same as or greater than the number of original source signals. In our study, we thus considered the case where the number of mixed signals is equal to the number of original signals. The main aim of ICA is to find Wsuch that the unmixed signals u

$$\mathbf{u} = \mathbf{Wx} = \mathbf{WAs} \text{ ---------- (3)}$$

are statistically independent, upto scaling and permutation the sources are recovered. Thus, W can be updated by using a learning rule to make u = s. The following learning rule for strictly sub-Gaussian sources is

$$\Delta\mathbf{W} \propto [\mathbf{I} + \tanh(\mathbf{u})u^t - \mathbf{u}u^t]\mathbf{W} \text{ -------- (4)}$$

and the learning rule for strictly super-Gaussian sources is

$$\Delta\mathbf{W} \propto [\mathbf{I} - \tanh(\mathbf{u})u^t - \mathbf{u}u^t]\mathbf{W} \text{ ---------(5)}$$

In this work, the learning rule for super-Gaussian sources is implemented. (5) can be further formulated by the updated equation expressed as

$$\Delta\mathbf{W} = \mathbf{L}\,[\mathbf{I} - \tanh(\mathbf{u})u^t - \mathbf{u}u^t]\mathbf{W} \text{ ----------(6)}$$

III. METHOD OVERVIEW

Figure 3 shows the architecture of the proposed ICA core based on the (6). It should be noted that in order to meet the requirement of small area for this ICA chip design, we here set the number of input mixed signals to 3 and the block sizefor weight updates to 64, and bit-length of the input/output signals are all 16-bit. According to the training computation, W is updated using (6), and the estimated separated signal is then calculated using the updated W. There are a lot of matrix multiplications required by the ICA algorithm and thus, we here utilized the systolic array to implement the matrix multiplication computation. That can reduce a huge number of computations so the area cost may be greatly saved. On the other hand, since it is difficult to implement the computation of tanh(θ) onto a fixed point digital circuit, we employed the Lambert's continued fraction. This can save circuit area a lot and enhance the computation speed and thus, the proposed architecture can achieve low-cost and high-speed VLSI design. Also note that since the proposed circuit is scalable and extendable, we can always obtain a new design when the values of parameters, such as the block size, become larger (say, the block size of 128 data points).

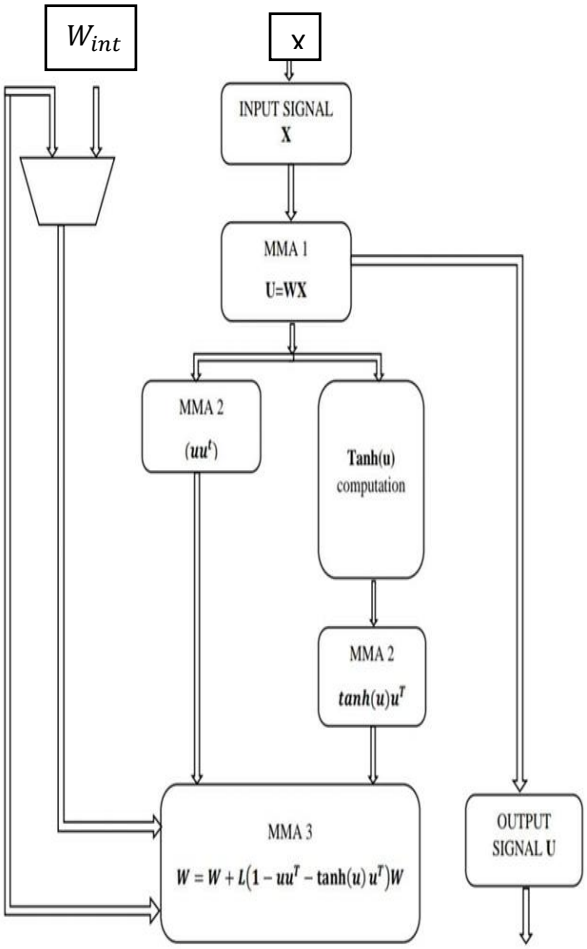

Figure 3: Architecture of the ICA core

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 06, June-2022

A. Matrix Multiplication Array (MMA)

There are three input signals in the proposed ICA architecture, and every 64 data points are used for upd are utilized at the weight matrix W. A 3×64 received data matrix x is processed in each training computation involving lots of multiplications and additions. To save the circuit area, we proposed the matrix multiplication array (MMA) with systolic array architecture to execute the required computations by reusing the hardware resources. Since the systolic array has the good feature of resource reused, we only need to design a single unique processing element (PE) for performing the matrix multiplication computations. Therefore, the total numbers of multipliers and adders can be saved efficiently. Here in our study, there are four basic modules designed in form of the systolic array structure in the proposed ICA architecture: the computation of:

(1) Wx,

(2) uu^t,

(3) tanh(u)u^t, and

(4) L(I − tanh(u)u^t − uu^t)W.

To reduce the area cost, a time sharing structure is employed to reuse the hardware resources, and one dimensional (1-D) systolic array is utilized in the proposed MMA-1 module as shown in Figure 4. The MMA-1 takes three cycles to calculate a single column vector of u within one period, and 64 periods to get the whole u matrix, thus, taking Wx operation takes 3×64 clock cycles.

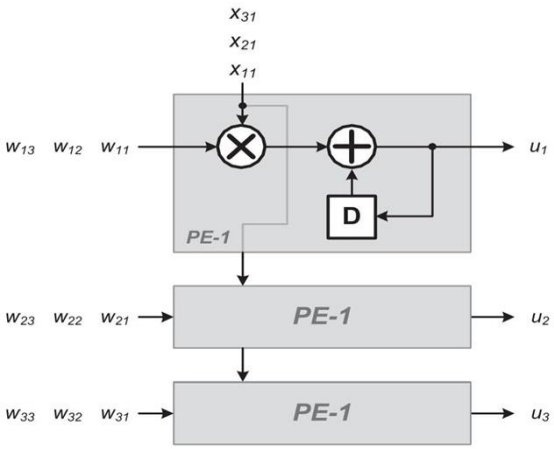

Figure4: Architecture of the proposed MMA-1 for **Wx** operation.

Figure 5 shows the architecture of proposed MMA-2. The MMA-2 consists of u$u^t$and tanh(u)u^tmodules, which execute the multiplication operation of a 3×64 and 64×3 matrices to get a 3×3 square matrix. To save the hardware resources, three multipliers take three clock cycles as an operation period to complete an operation of M3×1.N1×3. The 3×3 processing element (PE-2), as shown in Figure 5, array accumulate each result of M3×1.N1×3 for 64 periods to obtain the final result of the 3×3 square matrix. Thus, the 3×64 clock cycles are needed in the operations of u$u^t$and tanh(u)u^t.

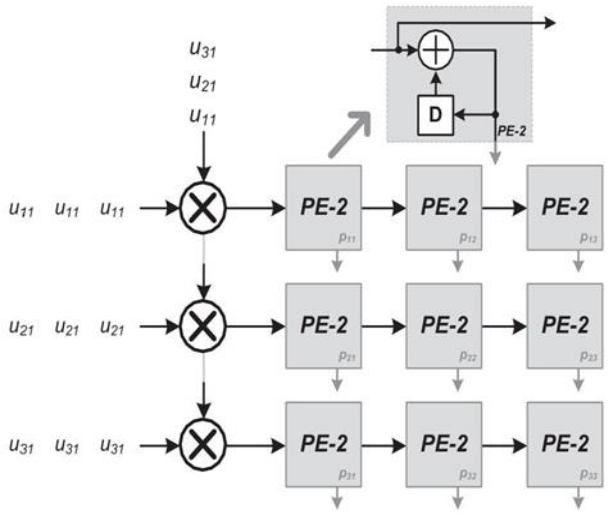

Figure5: Architecture of the proposed MMA-2 for u$u^t$and tanh(u)u^toperation.

The architecture of the proposed MMA-3 is as shown in Figure 6. According to the data flow, the results obtained from MMA-2 modules are 3×3 square matrices. These 3×3 square matrices are then fed into the proposed MMA3 module to obtain the weight updates. The proposed PE-3 consists of three multipliers and two adders to complete the multiplication of a 3×1 and a 1×3 vectors. The MMA-3 module takes three clock cycles to calculate the 3×3 square matrix so the W= L(I − tanh(u)u^t − uu^t)W can be obtained throughMMA-3 module.

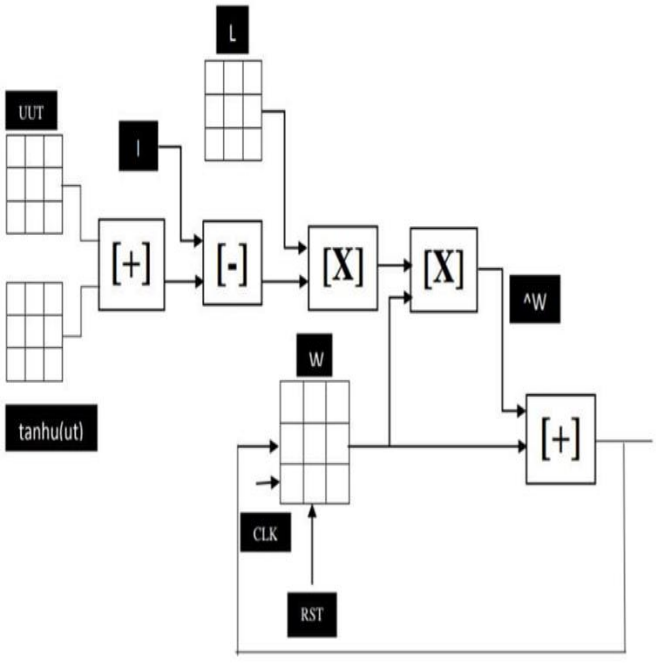

Figure6: Architecture of MMA3

B. Tanh Computation using Lambert's Continued Fraction

According to the ICA algorithm as indicated in (6), the operation of hyperbolic tangent, tanh(θ), is needed. In order

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 06, June-2022

to implement the operation of hyperbolic tangent, Lambert's continued fraction equation is utilized.
Using Lambert's continued fraction, the hyperbolic tangent can be written:

$$\tanh x = \cfrac{x}{1+\cfrac{x^2}{3+\cfrac{x^2}{5+\ldots}}}$$

The continued fraction equation is approximated as:

$$\tan h(x) = \cfrac{x}{1+\cfrac{x^2}{3+\cfrac{x^2}{5+\cfrac{x^2}{7+\cfrac{x^2}{19}}}}}$$

The variable x shown is the input which is a 10 bit floating point value and is approximated to 8 bit. Using bit shift calculator realization of bit shift operations with decimal, hexadecimal, binary and octal numbers are possible. For the x^2 calculation, a multiplier is used. When the counter is reset i.e, at zero, 19 is taken, bit adjuster, i.e, the quotient, then the value 7 gets added and is stored in the accumulator. When the counter is 1, the second line is selected in both and the same computation is performed according to the equation given below. At the last stage, the ouput is divided by x^2 and is added with 5. After performing the four clock cycles, finally it is divided by x, i.e, the input.

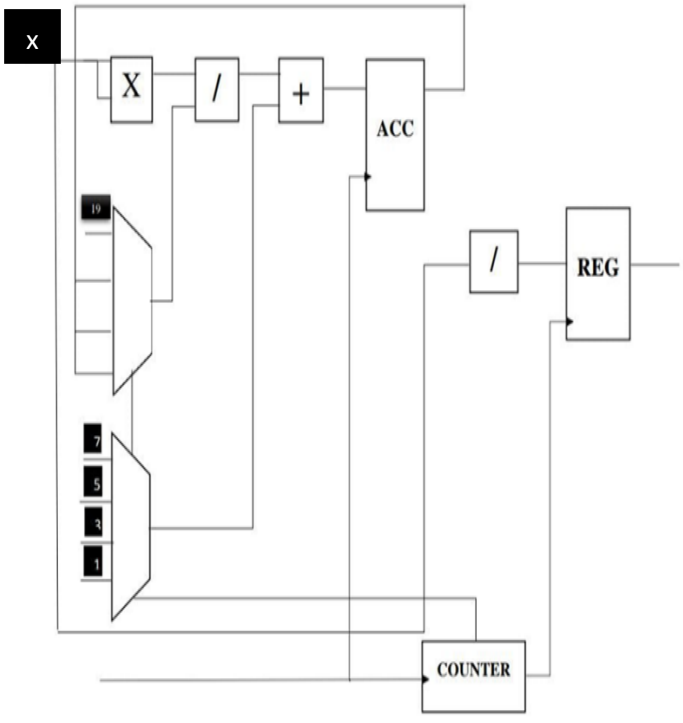

Figure7: Block diagram for tanh computation

IV. RESULT AND DISCUSSION

To evaluate the performance of the proposed ICA, three sets of PQRS waveform with the positive side of the signals were employed in the task of performance evaluation in this study. 64 data samples are taken by down sampling 256 data samples and is subtracted with the minimum most value to obtain the positive side of the signals. Random signals are added to each sets for generating the composite waveform. A 3x3 square weight matrix is initialized and a likelihood function with a value of 0.01 as diagonal matrix is provided. Weight updation takes place according to the ICA algorithm.

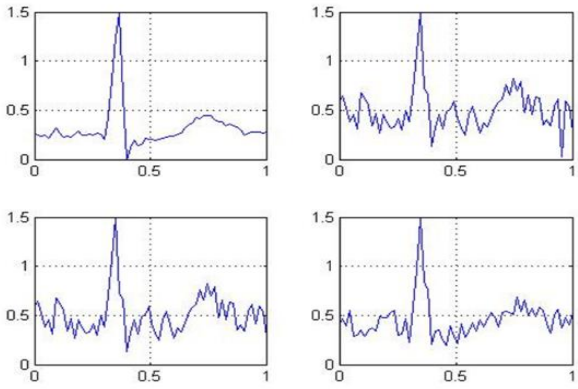

Figure8: Mixed Signal

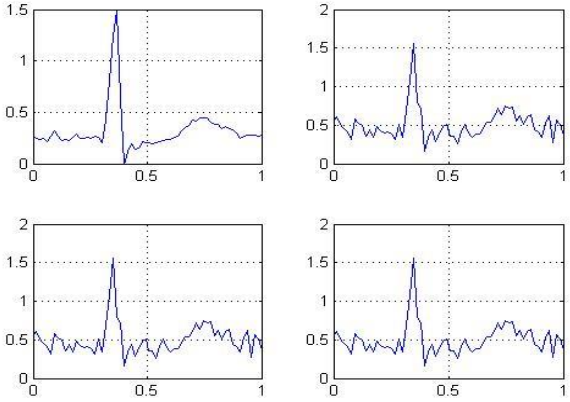

Figure9: Filtered Signal

The design entry is modeled using Verilog and the simulations of the designs were performed using ModelSim PE 5.5e.According to the ICA algorithm, in figures 10 and 11, the amplitude of a reconstructed signal may be amplified or reduced by a certain factor.

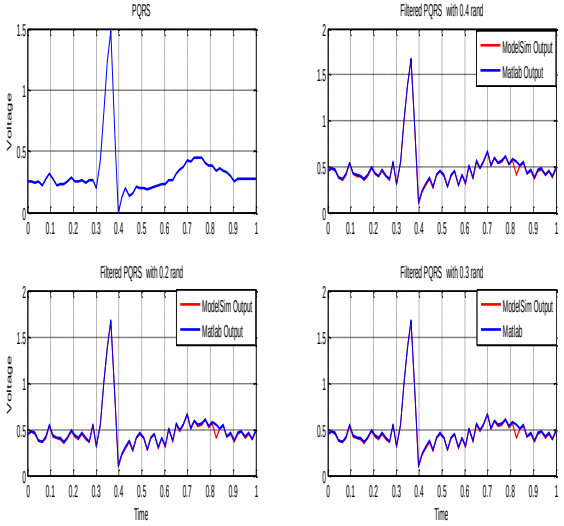

Figure10: Comparison between the signals

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 06, June-2022

Figure 10 shows the comparison between the signals obtained from ModelSim and Matlab output for the separation of mixed signals. The overall percentage of error was found to be less than 20%, this performance evaluation may pave the way for the applications in many areas. According to both the numerical experimental and the measured results obtained from this study, we may see that the proposed ICA chip can well achieve the requirements of high-speed, low-power, and small-area with supporting the separation of mixed signals contained super-Gaussian sources.

Using Xilinx ISE 14.7 for synthesizing, the proposed ICA can achieve 252MHz when implemented into Xilinx Virtex7. The Xilinx Xc7vx330t-3ffg1157 FPGA has the high performance and integration at 28nm CMOS technology. The device utilization summary gives the number of slices, number of slice LUTs, number of occupied slices, number of LUT flipflop pairs used.

TABLE 1
CIRCUIT CHARACTERISTICS AND RESOURCES USAGE OF THE ICA

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	0	408,000	0%
Number of Slice LUTs	334	204,000	1%
Number of occupied Slices	164	51,000	1%
Number of LUT Flip Flop pairs used	334		
Real Time to Xst completion		31sec	
Total CPU time to Xst		31.15sec	
Total Memory usage		330104Kb	
Total Delay		17.673ns	
Clock Frequency		252MHz	

V. CONCLUSION

This paper presents a VLSI implementation of the extended InfoMaxICA algorithm which can separate the mixed signals withsuper-Gaussian source signals. In the proposed ICA, atime sharing MMA module operates large dimension matrixmultiplications and the Lambert's Continued Fraction is employed to execute hyperbolic functions so the ICAalgorithm can be effectively and efficiently realized.The implementation presented here offers a highly accurate and scalable circuit for tanh computation and the circuit achieved a low area, high speed and low power design and the separated signals exhibit excellent quality.

REFERENCES

[1]. Yuan-Ho Chen, Szi-Wen Chen , andMin-Xian Wei, "AVLSI implementation of independentcomponent analysis for biomedicalsignal separation using CORDIC engine," IEEE transactions on biomedical circuits and systems, vol. 14, no. 2, Apr. 2020.

[2]. C. M. Kim and S. Y. Lee, "A digital chip for robust speech recognition in noisy environment," in *Proc.IEEE Int. Conf. Acoust., Speech, SignalProcess.*, vol. 2, 2001, pp. 1089–1092.

[3]. Kwak and W. Pedrycz, "Face recognition using an enhanced independent component analysis approach," *IEEE Trans. Neural Network.*, vol. 18, no. 2, pp. 530–541, Mar. 2007.

[4]. T. A. Soomro, T. Mahmood Khan, M. A. U. Khan, J. Gao, M. Paul, and L. Zheng, "Impact of ICA-based image enhancement technique on retinal blood vessels segmentation," *IEEE Access*, vol. 6, pp. 3524–3538, 2018.

[5]. C. F. Beckmann and S. M. Smith, "Probabilistic independent component analysis for functional magnetic resonance imaging," *IEEE Trans. Medical.Imaging.*, vol. 23, no. 2, pp. 137–152, Feb. 2004.

[6]. G. Wang, C. Teng, K. Li, Z. Zhang, and X. Yan, "The removal of EOG artifacts from EEG signals using independent component analysis and multivariate empirical mode decomposition," *IEEE J. Biomed. HealthInformation.*, vol. 20, no. 5, pp. 1301–1308, Sep. 2016.

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

[8]. T.-W. Lee, M. Girolami, and T. J. Sejnowski, "Independent componentanalysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources," *Neural Comput.*, vol. 11, no. 2, pp. 417–441, 1999.

[9]. T.-W. Lee, *Independent Component Analysis: Theory and Applications*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[10]. W. C. Huang, S. H. Hung, J. F. Chung, M. H. Chang, L. D. Van, and C. T. Lin, "FPGA implementation of 4-channel ICA for on-line EEG signal separation," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Nov. 2008, pp. 65–68.

[11]. F. Sattar and C. Charayaphan, "Low-cost design and implementation of an ICA-based blind source separation algorithm," in *Proc. Annu. IEEE Int.ASIC/SOC Conf.*, 2002, pp. 15–19.

[12]. 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.

[13]. H. Du and H. Qi, "An FPGA implementation of parallel ICA for dimensionality reduction in hyperspectral images," in *Proc. IEEE Int.Geosci. Remote Sens. Symp.*, vol. 5, Sep. 2004, pp. 3257–3260.

[14]. C. Charoensak and F. Sattar, "A single-chip FPGA design for real-time ICA-based blind source separation algorithm," in *Proc. IEEE Int. Symp.Circuits Syst.*, May 2005, pp. 5822–5825.

[15]. K. Shyu, M. Lee,Y.Wu, and P. Lee, "Implementation of pipelined FastICA on FPGAfor real-time blind source separation," *IEEE Trans.NeuralNetw.*, vol. 19, no. 6, pp. 958–970, Jun. 2008.

[16]. M. Stana´cevi´c, S. Li, and G. Cauwenberghs, "Micropower mixed-signal VLSI independent component analysis for gradient flow acoustic source separation," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 63, no. 7,pp. 972–981, Jul. 2016.

[17]. H. Du, H. Qi, and G. D. Peterson, "Parallel ICA and its hardware implementation in hyperspectral image analysis," in *Proc. SPIE DefenseSecur. Symp.*, vol. 5439, Apr. 2004, pp. 74–83.

[18]. L. D. Van, D. Y. Wu, and C. S. Chen, "Energy-efficient FastICA implementation for biomedical signal separation," *IEEE Trans. NeuralNetw.*, vol. 22, no. 11, pp. 1809–1822, Nov. 2011.

[19]. K. Leboeuf, A. H. Namin, R. Muscedere, H. Wu and M. Ahmadi, "High Speed VLSI Implementation of the Hyperbolic Tangent Sigmoid Function," 2008 Third International Conference on Convergence and Hybrid Information Technology, Busan, 2008

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 06, June-2022

[20]. A. H. Namin, K. Leboeuf, R. Muscedere, H. Wu and M. Ahmadi, “Efficient hardware implementation of the hyperbolic tangent sigmoid function,” IEEE Int. Symp. on Circuits and Systems, 2009
[21]. B. Zamanlooy, M. Mirhassani, “Efficient VLSI Implementation of Neural Networks With Hyperbolic Tangent Activation Function”, IEEE Trans. on VLSI system, 22(1), 2014
[22]. C-W. Lin and J-S. Wang, “A digital circuit design of hyperbolic tangent sigmoid function for neural networks”, IEEE International Symposium on Circuits and Systems (ISCAS), 2008
[23]. F. H. Adnan, M. F. O. Mahmud and W. F. H. Abdullah, "Hyperbolic tangent activation function integrated circuit implementation for perceptrons," 2012 IEEE Student Conference on Research and Development (SCOReD), Pulau Pinang, 2012
[24]. A. M. Abdelsalam, J. M. P. Langlois, F. Cheriet, “A Configurable FPGA Implementation of the Tanh Function using DCT Interpolation”, 2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines
[25]. Z. Hajduk, “Hardware implementation of hyperbolic tangent and sigmoid activation functions” Bulletin of the Polish Academy of Sciences Technical Sciences 66(5), 2018
[26]. Weisstein, Eric W. "Hyperbolic Tangent." From MathWorld--A Wolfram Web Resource, online (accessed 13th Februray 2020)
[27]. E. Weisstein, “Newton’s method”, online (accessed 13th Februray 2020)
[28]. Ron Doerfler, “Fast Approximation of the Tangent, Hyperbolic Tangent, Exponential and Logarithmic Functions”, 2007, online (accessed 13th Februray 2020)