2010 7th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2010)
Tuxtla Gutiérrez, Chiapas, México. September 8-10, 2010.

Optimized Infomax-ICA algorithm on FPGA Architecture for Blind Source Separation

L. Noé Oliva-Moreno[1], Jesús de la Cruz-Alejo[2], José A. Moreno-Cadenas[3]
[1] The Superior School of Computer Sciences, National Polytechnic Institute, Mexico D.F., Mexico
[2] Tecnológico de Estudios Superiores de Ecatepec, México, México
[3]Department of Electrical Engineering, CINVESTAV-IPN, Mexico D.F., Mexico
Phone (52) 57296000 E-mail: noetronix@hotmail.com

Abstract — **This work presents an optimized Implementation on Field Programmable Gate Array (FPGA) Architecture for an Infomax algorithm based on Independent Component Analysis (ICA). We use this algorithm to solving Blind Source Separation (BSS) problems in real-time mixed signal processing in order to clean speech signals under noisy environments and to probe the potential of this kind of algorithms embedded in hardware architectures. The work shows a new digital architecture of neural network composed by two dimensional arrays, the output signals present successful results according to theorical analysis and achieving the signals separation.**

Keywords — **Independent Component Analysis, Blind Source Separation, FPGA.**

I. INTRODUCTION

In recent works The ICA algorithm has been studied and developed because it is a potential method to do signals processing that it can be used in many applications, such as: features extraction, Adaptive Noise Canceling (ANC) and blind Signal Separation (BSS), which are used in biomedical signals, telecommunication systems, image processing, and voice recognition, among others [1]. Also, ICA has been studied by many researchers in the neural networks and statistical signal processing areas and they have developed an unsupervised learning algorithm based on entropy maximization (Infomax) in a single-layer feedforward neural network [1], [2].

This paper presents the digital architecture implementation on FPGA for Infomax algorithm, which is used to clean speech signals under noisy environments. The algorithm architecture performance was verified by ISE Web Pack software. Finally, real-time results show that the noise is reduced and the speech signal is cleaned.

This paper is organized as follows. Section 2, describes the background of the ICA algorithm. Section 3, presents the architecture of the neural network, and VHDL ISE simulations. Section 4, presents the implementation algorithm architecture on VHDL. Section 5, shows the experimental results. Finally, conclusions are presented in Section 6.

II. MATHEMATIC DESCRIPTION

Fig. 1 shows a measure signal ***x*** which it is a mixture of the independent signal ***s,*** which it passes through an Unknown matrix **A**. The goal of the ICA algorithm is to recover the unknown independent signal ***s,*** given only the sensor observations ***x***, which are the output linear unknown mixtures of **A,** which it represents the unknown independent source signals. The model is given by:

$$\boldsymbol{x}=\boldsymbol{As} \tag{1}$$

The ICA algorithm finds the un-mixing matrix **W**, which is the inverse matrix of **A**. The matrix **W** makes the outputs ***s*** as independent as possible. On this manner, we have:

$$\mathbf{s} = \mathbf{Wx} \tag{2}$$

Furthermore, the Infomax algorithm is a way to find the un-mixing matrix **W** by means of maximizing the entropy *H(y)* of the outputs, as Bell and Sejnowski proposed [1,2].

Later, the joint entropy at the outputs of a neural network is given by:

$$H(y_1,y_2,..,y_n)=H(y_1)+(y_2)+...+H(y_n)-I(y_1,y_2,...,y_n) \tag{3}$$

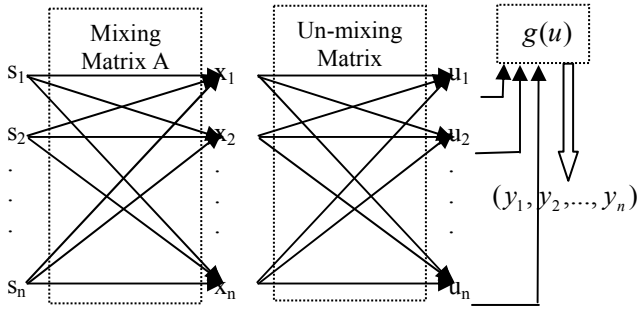

Fig. 1 Mixing and unmixing instantaneous model.

IEEE Catalog Number: CFP10827-ART
ISBN: 978-1-4244-7314-4
978-1-4244-7314-4/10/$26.00 ©2010 IEEE

Where, $H(y_i)$ are the marginal entropies of the outputs and $I(y_i,...y_n)$, are their mutual information. Also each one marginal entropy can be written as:

$$H(y_i) = -E\{\log p(y_i)\} \tag{4}$$

The nonlinear mapping between the output density $p(y_i)$ and the estimated signal density $p(u_i)$, can be described by the absolute value of the derivative with respect to u_i [3], given by (5).

$$p(y_i) = \frac{p(u_i)}{\left|\partial y_i / \partial u_i\right|} \tag{5}$$

Taking the derivative of the joint entropy in (3) and rewriting we obtain:

$$\frac{\partial H(y)}{\partial W} = \frac{\partial}{\partial W}(-I(y)) - \frac{\partial}{\partial W}\sum_{i=1}^{n} E\{\log \frac{p(u_i)}{\left|\partial y_i / \partial u_i\right|}\} \tag{6}$$

A direct minimization of the mutual information is achieved when $p(u_i) = \left|\partial y_i / \partial u_i\right|$ is satisfied. So, the mutual information will be minimized when the nonlinearity $y_i = g_i(u)$ will be the cumulative density function of the estimated signal u_i.

The maximum of the joint entropy *H(y),* can be obtained by deriving *H(y)* with respect to *W*, i.e. computing the gradient of *H(y)*. The natural gradient, re-scales the entropy by post-multiplying the entropy gradient by $W^T W$, given:

$$\Delta W \alpha \frac{\partial H(y)}{\partial W} W^T W = \left[I + \left(\frac{\partial p(u)/\partial u}{p(u)} \right) u^T \right] W \tag{7}$$

Where, *I* denote the identity matrix. Then, the nonlinearity is defined by:

$$\varphi(u) = -\frac{\partial p(u)/\partial u}{p(u)} \tag{8}$$

Also, it is called the score function. Now, substituting (8) in (7), we have:

$$\Delta W \alpha \left[I - \varphi(u) u^T \right] W \tag{9}$$

This equation is called the original Infomax Algorithm. An alternative way to derive the general learning rule is to give the maximum likelihood estimation (MLE) [4, 5].

The Hebb rule is applied to a recurrent neural network to update the weight. Where γ is the learning rate.

$$W(\Delta t + t) = W(t) + \gamma \cdot \Delta W$$

III. DIGITAL ARCHITECTURE DESIGN ON FPGA

The first step is to define the architecture algorithm given by (2) and (9). Fig. 2, shows the block diagram of a 2x2 Neural Network architecture which it was designed in Matlab´s Simulink. Some sine-wave signals were used to test the algorithm performance.

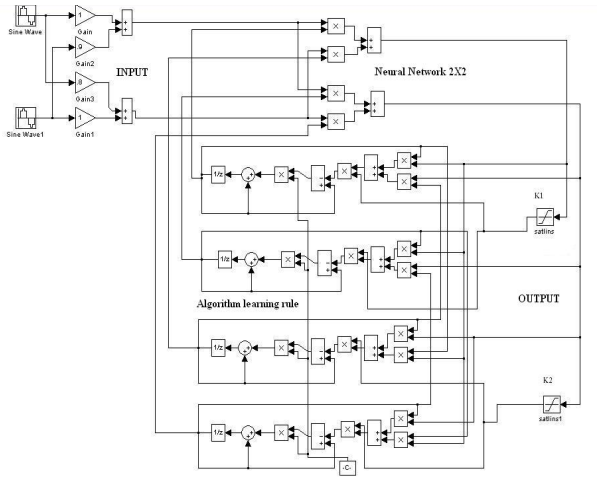

Fig. 2 Neural network architecture.

This block diagram was designed in a digital Architecture in order to be implemented on a FPGA. The FPGA is the core of the system, but there are other blocks that have to be taking in to account such as converters ADCs and DACs 12-bits serial. Fig. 3, shows a block diagram where the full system is described. The signals generators and the Operational Amplifier (*OPAMP*) based on voltage adders were used to obtain the mixing signals.

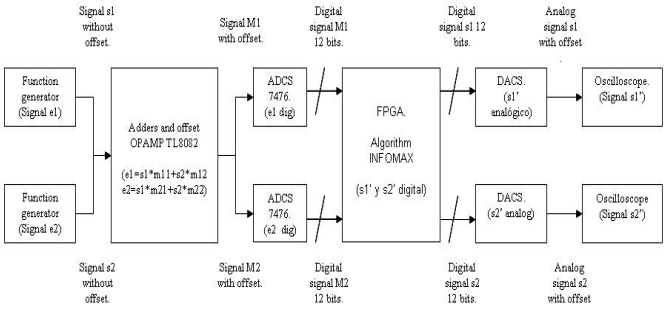

Fig. 3 Block diagram for the ICA system.

IEEE Catalog Number: CFP10827-ART
ISBN: 978-1-4244-7314-4
978-1-4244-7314-4/10/$26.00 ©2010 IEEE

Fig. 4, shows the schematic diagram to implement the Algorithm INFOMAX. To implement the Neural Network given by (2), we used 12 and 16-bits multipliers and 28-bits signed adders. The learning rule given by (9) is proposed by simple operations like signed adders and multipliers based on the modules concept. On the bottom of figure 4, we show the no-liner function (8) using a sat-linear to approximates a tanhyperbolic function, this let us a lower dimension in hardware.

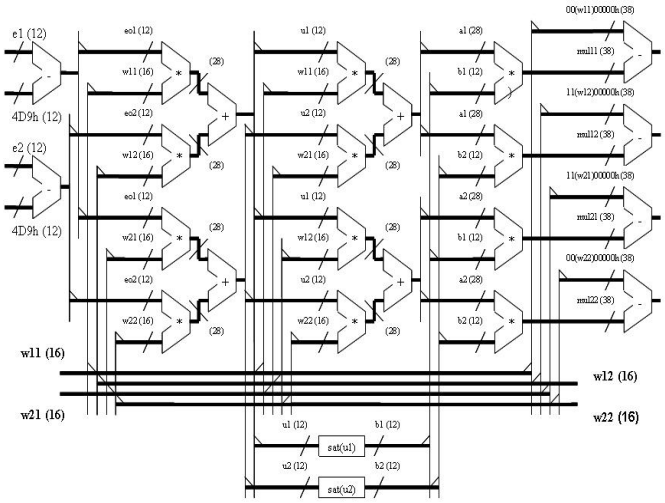

Fig. 4 Schematic diagram for two channels for implement the Neural Network and the learning rule.

Fig. 5 shows the weight update rule $w \leftarrow w + \gamma \cdot \Delta w$. The weight length has 16 bits. Furthermore, the accuracy can be increasing using a major length of bits. The arithmetic used in this implementation is simple, only using a 2's complement representation for negative numbers.

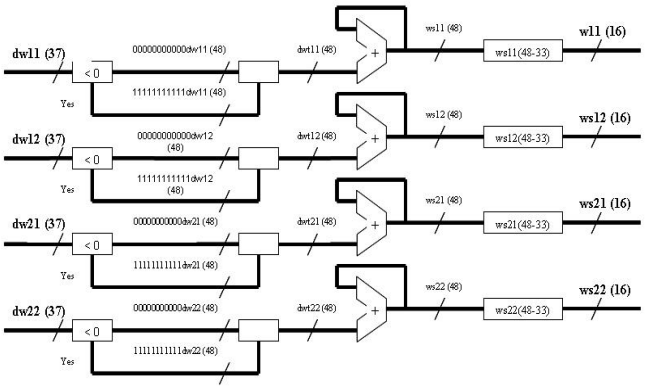

Fig. 5 Block diagram for the weight update.

The input signals from ADC have a 12-bits length, this increase its length according to the operation that was it done. Finally, the 16-bits outputs are truncated in order to fit the input of the DAC. The signals obtained will be shown in next section.

IV. SIMULATION RESULTS

The simulation was done using sine waves which they first were taken as the source signals. Then the mixed signals were generated by multiplying the mixing matrix and source signals. The expression for the mixing matrix is shown in (10):

$$\begin{bmatrix} x_i \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.9 \\ 0.8 & 1 \end{bmatrix} \bullet \begin{bmatrix} \sin(2\pi 1000t) \\ \sin(2\pi 800t) \end{bmatrix} \qquad (10)$$

Figure 6 shows the mixed signals.

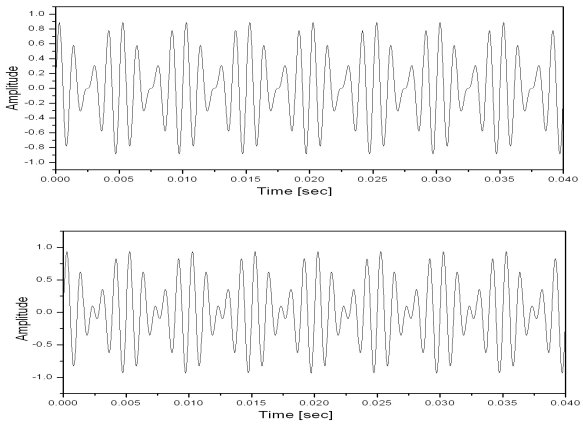

Fig 6 Input mixed signals.

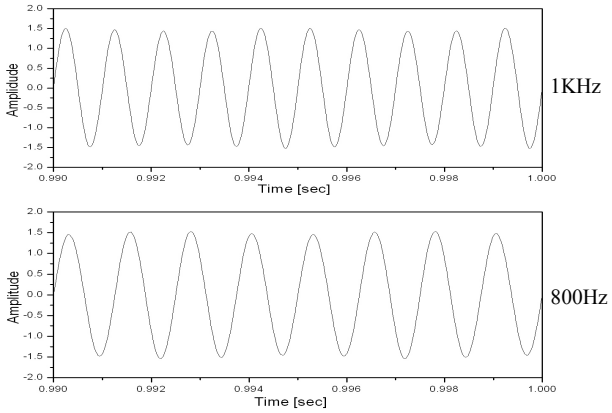

Fig. 7 Estimated independent component Sign.

Simulations results were done in Xilinx ISE Web Pack. These signals are shown in Fig 7. As we can see, the mixed signals were successfully separated (32dB approximately). The time simulation was 1 second and the convergence for the net was reached at 400ms approximately.

Now the net implementation will be done in next section and the experimental results will be presented.

IEEE Catalog Number: CFP10827-ART
ISBN: 978-1-4244-7314-4
978-1-4244-7314-4/10/$26.00 ©2010 IEEE

V EXPERIMENTAL RESULTS

As we mention in previous section, for the implementation we use 2 sine waves signal generators, 2 serial ADC, 1 DAC of 12 bits and an OPAMP in adder configuration for to get the mixed signals. One of the objectives of this works is to probe the potential of this kind of algorithms embedded in hardware architectures.

The Analog to Digital Module Converter, works at a maximum sampling rate, this is, about one million samples per second, fast enough for the most demanding audio applications. *Digital to Analog peripheral module*, converts signals from digital mode to analog voltages on two channels simultaneously with 12-bits of resolution. These two devices were selected due to theirs characteristics. Finally the code used for simulations on VHDL was implemented on FPGA SPARTAN3. Fig. 8 shows the experimental results obtained by Neural Network implementation. We can see that two sine wave was obtained of the mixture according to the simulation.

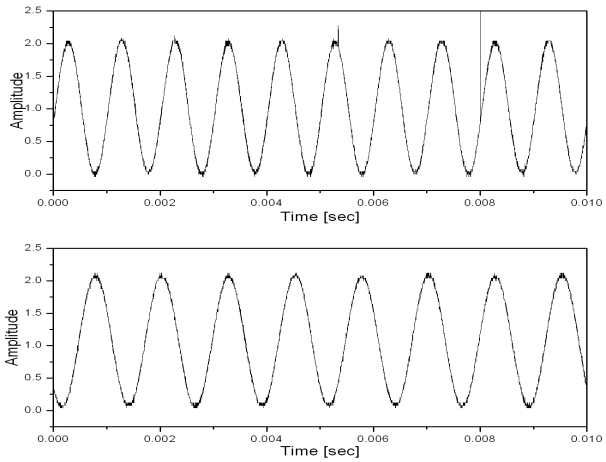

Fig 8. Experimental Results by Neural Network.

Table 1. Mixing matrix (M) and Separating Matrix (W)

Original Signals	Mixing Matrix (M)	Separating Matrix (W)	Error
$s_1 = 800Hz$ $s_2 = 1000Hz$	$\begin{bmatrix} 1 & 0.8 \\ 0.9 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -0.8076 \\ -0.8919 & 1 \end{bmatrix}$	$w_{12} = 0.0076$ $w_{21} = 0.0081$
	$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -0.5 \\ -0.5002 & 1 \end{bmatrix}$	$w_{12} = 0.0$ $w_{21} = 0.0002$
	$\begin{bmatrix} 1 & 0.3 \\ 0.1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -0.2933 \\ -0.1069 & 1 \end{bmatrix}$	$w_{12} = 0.0076$ $w_{21} = 0.0081$
	$\begin{bmatrix} 1 & 0.3 \\ 0.8 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -0.3382 \\ -0.7591 & 1 \end{bmatrix}$	$w_{12} = 0.0076$ $w_{21} = 0.0081$
	$\begin{bmatrix} 1 & 0.8 \\ 0.3 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -0.7591 \\ -0.3382 & 1 \end{bmatrix}$	$w_{12} = 0.0076$ $w_{21} = 0.0081$
$s_1 = 1500Hz$ $s_2 = 2000Hz$	$\begin{bmatrix} 1 & 0.8 \\ 0.3 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -0.7591 \\ -0.3382 & 1 \end{bmatrix}$	$w_{12} = 0.0409$ $w_{21} = 0.0382$

The prototype system works without the need of a personal computer and compared with other system [6, 7, 8] is very fast and small.

Table 1 presents the values for the matrix mixing and un-mixing. Different mixtures and different frequencies were proved and the same results were obtained to characterize the system.

VI. CONCLUSIONS

In this paper, we present a FPGA implementation for two simultaneous channels based on ICA-Infomax applied to the BSS problem and an ANC in real-time, accompanied with related ADC and DAC peripherals for real world signal processing. This architecture is optimized and improved using the FPGA resources. The tests were done in SPARTAN 3 (XC3S200), which is one of the smallest FPGA of Xilinx.

The experimental results obtained shows that the performance was good for all matrixes tested and the error was minimum.

ACKNOWLEDGMENT

This work was supported by The Superior School of Computer Sciences of National Polytechnic Institute.

REFERENCES

[1] Te-Won Lee, M. Girolami and T.J. Sejnowski," Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Subgaussian and Supergaussian Sources". Neural Computation,11, pp. 417-441, 1999.

[2] Te-Won Lee, "Independent Component Analysi", Kluwer Academic Publisher, California, 2000.

[3] Papoulis A., "Probability and Statistics", Prentice Hall, New Jersey, 1990.

[4] Gaeta M.. and Lacoune J., "Source Separation without prior Knowledge: the Maximum Likelihood Solution", Proc. EUSIPO, pp 621-624, 1990

[5] Cardoso J., "Infomax and maximum likelihood for Blind Source Separation", IEEE Signal Processing Letters, 4(4), pp 112-114, 1997.

[6] Chang-Min Kim, Hyung-Min Park, Taesu Kim, Yoon-Kyung Choi and Soo-Young, "FPGA Implementation of ICA Algorithm for Blind Source Separation and Adaptive Noise Canceling", IEEE Transaction on Neural Networks, Vol. 14, No. 5, 2003, pp. 1038-1046.

[7] Kuo-Kai Shyu, Ming-Huan Lee, Yu Te Wu and Po-Lei Lee, "Implementation of Pipelined FastICA on FPGA for Real-Time Blind Source Separation", IEEE Transactions on Neural Networks, IEEE, Col 19, No6, June 2008, pp. 958-970.

[8] W. Valenzuela, G. Carvajal and M. Figueroa, "Blind Source-Separation in Mixed-Signal VLSI Using the Infomax Algorithm", Spring-Verlag Berlin Heidelberg, ICANN, Part II, LNCS 51464, pp. 208-217, 2008.

IEEE Catalog Number: CFP10827-ART
ISBN: 978-1-4244-7314-4
978-1-4244-7314-4/10/$26.00 ©2010 IEEE