International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 3.45 (SJIF-2015), e-ISSN: 2455-2585
Volume 4, Issue 5, May-2018

ROBUST CONVOLUTIVE BLIND SOURCE SEPERATION METHOD IN VLSI DESIGN

Korrapatti Mohammed Ghouse[1], R Pavan kumar[2]
[1,2]*Faculty in Department of ECE, IIIT RK VALLEY-RGUKT AP, Kadapa, AP*

Abstract- ***This brief presents an efficient very-large-scale integration architecture design for convolutive blind source separation (CBSS). The CBSS separation network derived from the information maximization (Infomax) approach is adopted. The proposed CBSS chip design consists mainly of Infomax filtering modules and scaling factor computation modules. In an Infomax filtering module, input samples are filtered by an Infomax filter with the weights updated by Infomax-driven stochastic learning rules. As for the scaling factor computation module, all operations including logistic sigmoid are integrated and implemented by the circuit design based on a piecewise-linear approximation scheme. The proposed architecture of this paper analysis the logic size and area using Xilinx 14.3.***

Key Words: CBSS, ASIC, DSP, CMOS, EDA, ECG

I. INTRODUCTION

SEPARATION of mixed sources has received extensive attention in recent years. Blind source separation (BSS) attempts to separate sources from mixed signals when most of the information for sources and mixing process is unknown. Such restrictions make BSS a challenging task for researchers. BSS has become a very important research topic in a lot of fields. Notable examples include audio signal processing, biomedical signal processing, communication systems, and image processing.Without a filtering effect, instantaneous mixing is considered a simple version of the mixing process of the source signals. However, for audio sources passing through an environmental filtering before arriving at the microphones, a convolutive mixing process occurs, and convolutive BSS (CBSS) is used to recover the original audio sources. Independent component analysis (ICA) is the conventional means of solving the BSS or CBSS problem. However, this method is often highly computationally intensive and introduces time-consuming processes for software implementation. More than a faster solution than software implementation, hardware solution achieves optimal parallelism. Providing hardware solutions for ICA-based BSS has drawn considerable attention recently. Cohen and Andreou explored the feasibility of combining above-and-subthreshold CMOS circuit techniques for implementing an analog BSS chip that integrates an analog I/O interface, weight coefficients, and adaptation blocks. This chip incorporates the use of the Herault–Jutten ICA algorithm. Cho and Lee implemented a fully analog CMOS chip based on information maximization (Infomax) ICA, as developed by Bell and Sejnowski. The chip incorporated a modular architecture to extend its use as a multichip. Apart from these analog BSS chips, various field programmable gate array (FPGA) implementations with digital architectures have been developed. Li and Lin realized the Infomax BSS algorithm based on system-level FPGA design, by using Quartus II, DSP builder, and Simulink. Du and Qi presented an FPGA implementation for the parallel ICA (pICA) algorithm, which focuses on reducing dimensionality in hyper spectral image analysis. The pICA algorithm consists of three temporally independent functional modules that are synthesized individually with some reconfigurable components developed for reuse. Based on Infomax BSS, Ounaset al. introduced a low-cost digital architecture implemented on FPGA. This design used merely one neuron to support sequential operations of the neurons in neural network. In 2008, Shyuet al. designed a pipelined architecture for FPGA implementation based on Fast ICA for separating mixtures of biomedical signals, including electroencephalogram (EEG), Magneto encephalography (MEG), and electrocardiogram (ECG). In this design, floating-point arithmetic units were used to increase the precision of the numbers and ensure the Fast ICA performance. Although FPGA has a short development time and inexpensive verification of algorithms in hardware, its hardware architecture design is not optimized in comparison with application specific integrated circuit (ASIC) fabricated in chips. A charyyaet al. designed an ASIC chip with 0.13-μm standard cell CMOS technology for 2-D Kurtotic Fast ICA. This design is characterized by reduced and optimized arithmetic units through means of removing dividers in eigenvector computation and whitening.

For portable EEG signal processing applications, Chen et al. developed a low-power verylarge- scale integration (VLSI) chip fabricated using the UMC 90-nm CMOS process. This chip can perform four-channel ICA to separate EEG and mixed EEG-like super-Gaussian signals in real time. However, the aforementioned ASIC chips focus on instantaneous mixing BSS. In this brief, we present a digital ASIC chip for CBSS, in which the source signals are convolutively mixed. The convolutive mixtures are separated using the CBSS separation network extended from Infomax theory. The CBSS problem was solved in the time domain mainly because in the frequency domain, the permutation and scalingambiguity among the frequency bins must be resolved. Tackling the permutation and scaling ambiguity requires a number of irregular operations that complicate the VLSI design of a CBSS chip. The approach used herein does not have this shortcoming. Furthermore, this approach allows us to propose a modular VLSI architecture. The proposed CBSS ASIC chip is characterized by its modular design, high speed, and low power. To the best of our knowledge, the proposed ASIC chip is the first that can execute the CBSS algorithm.

II. PRELIMINARIES

Separation of mixed sources has received extensive attention in recent years. Blind source separation (BSS)attempts to separate sources from mixed signals when most of the information for sources and mixing process is unknown .Such restrictions make BSS a challenging task for researchers.BSS has become a very important research topic in a lot of fields. Notable examples include audio signal processing, biomedical signal processing, communication systems, and image processing. Without a filtering effect, instantaneous mixing is considered a simple version of the mixing process of the source signals. However, for audio sources passing through an environmental filtering before arriving at the microphones, a convolutive mixing process occurs, and convolutive BSS(CBSS) is used to recover the original audio sources.

The CBSS problem was solved in the time domain mainly because in the frequency domain, the permutation and scaling ambiguity among the frequency bins must be resolved. Tackling the permutation and scaling ambiguity requires a number of irregular operations that complicate the VLSI design of a CBSS chip. The approach used herein does not have this shortcoming. Furthermore, this approach allows us to propose a modular VLSI architecture. The proposed CBSS ASIC chip is characterized by its modular design, high speed, and low power.To the best of our knowledge, the proposed ASIC chip is the first that can execute the CBSS algorithm.

BSS USING INFOMAX

Assume there are N source signals recorded by M sensors. This brief focuses on convolutive mixing. The related model is mathematically expressed by

$$x_m(t) = \sum_{n=1}^{N} \sum_{k=0}^{L-1} h_{mn}(k) s_n(t-k) \qquad (1)$$

Where Xm, m = 1, 2, . . .,M, is a mixed signal correspondingto sensor m; sn, n = 1, 2, . . .,N, is the nth source signal; hmnis the unknown impulse response from source n to sensor m; t is the discrete time index; and L is the number of taps in convolution. CBSS, a demixing or separation process, finds separatedsignals that approximate the original sources. The separationprocess can be expressed as

$$u_n(t) = \sum_{m=1}^{M} \sum_{l=0}^{L-1} w_{nm}^{k} x_m(t-k) \qquad (2)$$

whereunis the nth separated signal; wnmis the L-tap separationfilter for sensor m to separated signal n; and wknmdenotes the kth tap weight of wnm. The separation process (2) can be expressed inmatrix form as

$$\boldsymbol{u}(t) = \sum_{k=0}^{L-1} \mathbf{W}^{k} \boldsymbol{x}(t-k) \qquad (3)$$

Where Wk is an N ×M matrix with wk n mas its components;x(t) = (x1(t), x2(t), . . . , xM(t))T; and u(t) = (u1(t), u2(t), . . . , uN(t))T .According to the separation model described by (2) or (3), seeking the separation filters is of priority concern in CBSS. To tackle this difficult problem, this brief adopts the Infomax approach. Infomax Approach for Convolutive Mixing BSS. The BSS problem assumes that statistical independence among source signals exists.

Let sndenote the nth source signal. The joint probability density function of all the sources can be written as

$$p(\boldsymbol{s}) = p(s_1, s_2, \ldots, s_N) = \prod_{n=1}^{N} p(s_n) \qquad (4)$$

where $p(s_n)$ is the probability density function of s_n.

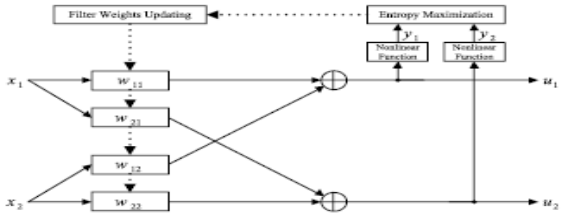


Fig. 1. Infomax-based CBSS separation network for the two-source and two-sensor case. This network contains four causal FIR filters w_{ij}^k, and u_i is the separated signal.

Accordingly, the statistically independent sources do not carry any mutual information I, which is defined in

Fig 3.1

$$I(s_1, s_2, \ldots, s_N) = \sum_{} H(s_i) - H(\boldsymbol{s}) \qquad (5)$$

Where H denotes the differential entropy. While BSS attemptsto generate separated signals close to source signals, the separation process focuses on generating output signals with zero mutual information.

To minimize the mutual information, Bell and Sejnowskideveloped the Infomax approach to learn the separating process. This approach maximizes the joint entropy of the outputs by a stochastic gradient ascent algorithm. As plain maximization of the joint entropy of the outputs may diverge to infinity, the Infomax approach maximizes the joint entropy of y = g(u), where g(.) refers to a nonlinear and monotonically transfer function. In convolutive mixing, the separation process is driven by Wk, which is a matrix comprising filter components. Consider two sources and two sensors, in which Fig. 3.1 depicts the Infomax-based CBSS separation network , the separated signals ui(t) are obtained by the following network computation:

$$\begin{aligned} u_1(t) &= u_{11}(t) + u_{12}(t) \\ &= \sum_{k=0}^{L_{11}} w_{11}^k(t) x_1(t-k) + \sum_{k=0}^{L_{12}} w_{12}^k(t) x_2(t-k) \qquad (6) \\ u_2(t) &= u_{21}(t) + u_{22}(t) \\ &= \sum_{k=0}^{L_{22}} w_{22}^k(t) x_2(t-k) + \sum_{k=0}^{L_{21}} w_{21}^k(t) x_1(t-k) \qquad (7) \end{aligned}$$

Where wkijare causal finite-impulse response (FIR) filters, anduij(t) is the partial result generated from wkijandxj(t).With logistic sigmoid as the nonlinear transfer function, thestochastic learning rules derived from the Infomax approach fornonzero delay weights are

$$\Delta w_{ij}^k(t) \propto (1 - 2y_i(t))\, x_j(t-k), \qquad k \neq 0. \qquad (8)$$

As for the zero delay weights, their stochastic learning rules are given as

$$\Delta w_{ij}^0(t) \propto (1 - 2y_i(t))\, x_j(t) + d_{ij}(t) \qquad (9)$$

where $d_{ij}(t) = \text{cofactor}(w_{ij})\ (\det \mathbf{W}^0)^{-1}$; $\text{cofactor}(w_{ij})$ is the cofactor of w_{ij}; and det is the determinant.

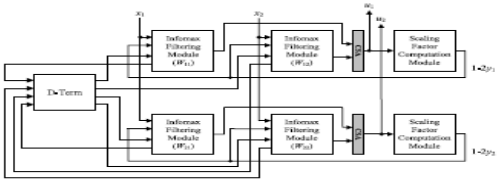


Fig. 2. Block diagram of the proposed CBSS chip that contains four Infomax filtering modules, two scaling factor computation modules, and a D-term unit. Two CSAs are used to sum up the Infomax filtering outputs.

III. IMPORTANCE OF HDLS

HDLs have many advantages compared to traditional schematic-based design.

•Design can be described at a very abstract level by us of HDLs.Designers can write their RTL description without choosing a specific fabrication technology. Logic synthesis tools can automatically convert the design to any fabrication technology. If a new technology emerges, designers do not need to redesign their circuit. They simply input the RTL description to the logic synthesis tool and create a new gate level netlist,using the new fabrication technology. The logic synthesis tool will optimize the circuit in area and timing for the new technology.

•By describing designs in HDLs,functional verification of the design can be done early in the design cycle. Since designers work at the RTL level, they can optimize and modify the RTL description until it meets the desired functionality. Most design bugs are eliminated at this point. This cuts down design cycle time significantly because the probability of hitting a functional bug at a later time in the gate-level netlist or physical layout is minimized.

•Designing with HDLs is analogous to computer programming. A textual description with comments is an easier way to develop and debug circuits. This also provides a concise representation of the design, compared to gate-level schematics. Gate-level schematics are almost incomprehensible for very complex designs.

•HDL-based designs are here to stay. With rapidly increasing complexities of digital circuits and increasingly sophisticated EDA tools,HDLs are now the dominant method for large digital designs. No digital circuit designer can afford to ignore HDL based design. Verilog HDL has evolved as a standard hardware description language. Verilog HDL offers many useful features.

•Verilog HDL is a general-purpose hardware description language that is easy to learn and easy to use. It is similar in syntax to the C programming language. Designers with C programming experience will find it easy to learn Verilog HDL.

•Verilog HDL allows different levels of abstraction to be mixed in the same model. Thus, a designer can define a hardware model in terms of switches, gates, RTL, or behavioral code. Also, a designer needs to learn only one language for stimulus and hierarchical design.

•Most popular logic synthesis tools support Verilog HDL. This makes it the language of choice for designers.

•All fabrication vendors provide Verilog HDL libraries for post logic synthesis simulation. Thus, designing a chip in Verilog HDL allows the widest choice of vendors.

•The Programming Language Interface (PLI) is a powerful feature that allows the user to write custom C code to interact with the internal data structures of Verilog. Designers can customize a Verilog HDL simulator to their needs with the PLI.

•The speed and complexity of digital circuits have increased rapidly. Designers have responded by designing at higher levels of abstraction. Designers have to think only in terms of functionality. EDA tools take care of the implementation details. With designer assistance, EDA tools have become sophisticated enough to achieve a close-to-optimum implementation.

•The most popular trend currently is to design in HDL at an RTL level, because logic synthesis tools can create gate-level net lists from RTL level design. Behavioral synthesis allowed engineers to design directly in terms of algorithms and the behavior of the circuit, and then use EDA tools to do the translation and optimization in each phase of the design.

•However, behavioral synthesis did not gain widespread acceptance. Today, RTL design continues to be very popular. Verilog HDL is also being constantly enhanced to meet the needs of new verification methodologies.

•Formal verification and assertion checking techniques have emerged. Formal verification applies formal mathematical techniques to verify the correctness of Verilog HDL descriptions and to establish equivalency between RTL and gate-level net lists. However, the need to describe a design in Verilog HDL will not go away. Assertion checkers allow checking to be embedded in the RTL code. This is a convenient way to do checking in the most important parts of a design.

•New verification languages have also gained rapid acceptance. These languages combine the parallelism and hardware constructs from HDLs with the object oriented nature of C++. These languages also provide support for automatic stimulus creation, checking, and coverage. However, these languages do not replace Verilog HDL. They simply boost the productivity of the verification process. Verilog HDL is still needed to describe the design.

•For very high-speed and timing-critical circuits like microprocessors, the gate-level netlist provided by logic synthesis tools is not optimal. In such cases, designers often mix gate-level description directly into the RTL description to achieve optimum results. This practice is opposite to the high-level design paradigm, yet it is frequently used for high-speed designs because designers need to squeeze the last bit of timing out of circuits, and EDA tools sometimes prove to be insufficient to achieve the desired results.

•Another technique that is used for system-level design is a mixed bottom-up methodology where the designers use either existing Verilog HDL modules, basic building blocks, or vendor-supplied core blocks to quickly bring up their system simulation. This is done to reduce development costs and compress design schedules. For example, consider a system that has a CPU, graphics chip, I/O chip, and a system bus.

•The CPU designers would build the next-generation CPU themselves at an RTL level, but they would use behavioral models for the graphics chip and the I/O chip and would buy a vendor-supplied model for the system bus. Thus, the system-level simulation for the CPU could be up and running very quickly and long before the RTL descriptions for the graphics chip and the I/O chip are completed.

TYPICAL DESIGN FLOW:

A typical design flow for designing VLSI-IC circuits show the level of design representation shaded blocks show processes in the design flow. The design flow used by designers who use HDLs. In any design, specifications are written first. Specifications describe abstractly the functionality, interface, and overall architecture of the digital circuit to be designed. At this point, the architects do not need to think about how they will implement this circuit. A behavioral description is then created to analyze the design in terms of functionality, performance, and compliance to standards, and other high-level issues. Behavioral descriptions are often written with HDLs. The behavioral description is manually converted to an RTL description in an HDL.

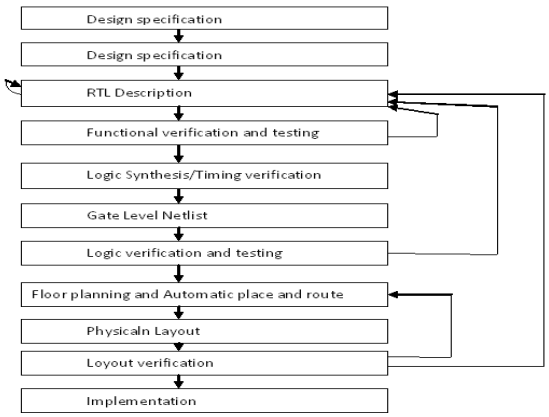


Fig Typical Design Flow

Logic synthesis tools convert the RTL description to a gate-level net list. A gate-level net list is a description of the circuit in terms of gates and connections between them. Logic synthesis tools ensure that the gate-level net list meets timing, area, and power specifications. The gate-level net list is input to an Automatic Place and Route tool, which creates a layout. The layout is verified and then fabricated on a chip.

Thus, most digital design activity is concentrated on manually optimizing the RTL description of the circuit. After the RTL description is frozen, EDA tools are available to assist the designer in further processes. Designing at the RTL level has shrunk the design cycle times from years to a few months. It is also possible to do many design iterations in a short period of time. Behavioral synthesis tools have begun to emerge recently. These tools can create RTL descriptions from a behavioral or algorithmic description of the circuit. As these tools mature, digital circuit design will become similar to high-level computer programming. Designers will simply implement the algorithm in an HDL at a very abstract level. EDA tools will help the designer convert the behavioral description to a final IC chip.

It is important to note that, although EDA tools are available to automate the processes and cut design cycle times, the designer is still the person who controls how the tool will perform. EDA tools are also susceptible to the "GIGO : Garbage In Garbage Out" phenomenon. If used improperly, EDA tools will lead to inefficient designs. Thus, the designer still needs to understand the nuances of design methodologies, using EDA tools to obtain an optimized design.

IV. SIMULATION TOOLS

After generating a programming file, configure the device. During configuration, generate configuration files and download the programming files from a host computer to a Xilinx device. Project Navigator organizes design files and runs processes to move the design from design entry through implementation to programming the targeted Xilinx device. Project Navigator is the high-level manager for Xilinx FPGA and CPLD designs, which allows to do the following:

•Add and create design source files, which appear in the Sources window..

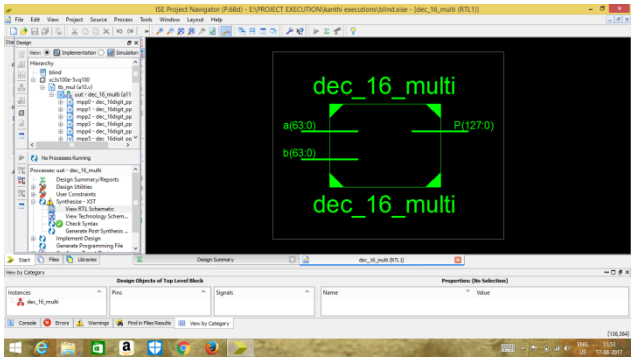
•Modify source files in the Workspace.

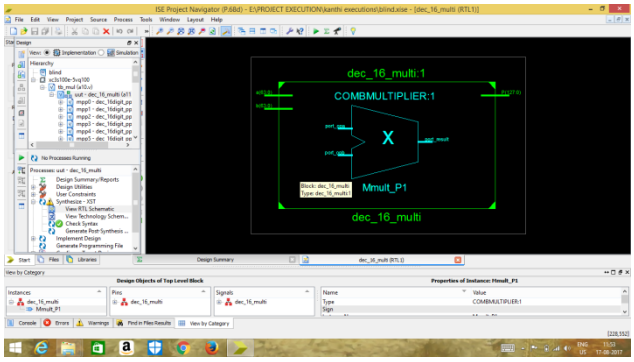
• Run processes on source files in the Processes window.

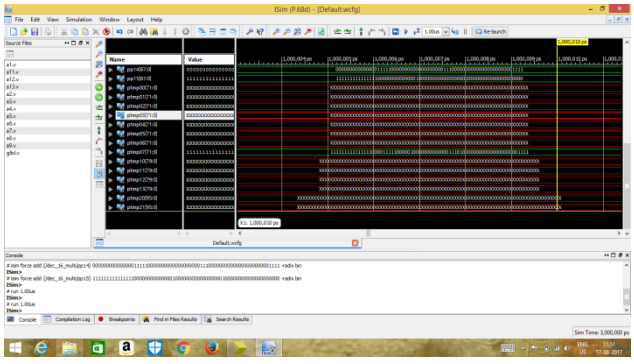
•View output from the processes in the Transcript window.

SCHEMATIC VIEW:

The Technological Optimized view of the Synthesized Design







VI. CONCLUSION

In this brief, an efficient VLSI architecture design for CBSS has been presented. The architecture mainly comprising Infomax filtering modules and scaling factor computation modules performs CBSS separation network derived from the Infomax approach. With TSMC 90-nm CMOS technology, the die size of the proposed ASIC chip is roughly 0.54 × 0.54 mm2. For the 1.8-V power supply, the maximum clock rate is 100 MHz The power dissipation is roughly 54.86mWunder the 100-MHz clock rate. The proposed CBSS ASIC chip can be used in preprocessing and integrated with other audio processing chips and peripheral components to form a whole audio processing system.

REFERENCE

[1] G. Zhou, Z. Yang, S. Xie, and J. M. Yang, "Online blind source separation using incremental nonnegative matrix factorization with volume constraint," IEEE Trans. Neural Netw., vol. 22, no. 4, pp. 550–560, Apr. 2011.

[2] M. Li, Y. Liu, G. Feng, Z. Zhou, and D. Hu, "OI and fMRI signal separation using both temporal and spatial autocorrelations," IEEE Trans.Biomed. Eng., vol. 57, no. 8, pp. 1917–1926, Aug. 2010.

[3] A. Tonazzini, I. Gerace, and F. Martinelli, "Multichannel blind separation and deconvolution of images for document analysis," IEEE Trans. ImageProcess., vol. 19, no. 4, pp. 912–925, Apr. 2010.

[4] H. L. N. Thi and C. Jutte, "Blind source separation for convolutivemixtures," Signal Process., vol. 45, no. 2, pp. 209–229, Aug. 1995.

[5] A. J. Bell and T. J. Sejnowski, "Blind separation and blind deconvolution: An information-theoretic approach," in Proc. Int. Conf. Acoust., Speech,Signal Process., May 1995, vol. 5, pp. 3415–3418.

[6] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," Neural Netw., vol. 13, no. 4/5, pp. 411–430, May/Jun. 2000.

[7] M. H. Cohen and A. G. Andreou, "Analog CMOS integration and experimentation with an autoadaptive independent component analyzer," IEEETrans.Circuits Syst. II, Analog Digit. Signal Process., vol. 42, no. 2, pp. 65–77, Feb. 1995.

[8] K. S. Cho and S. Y. Lee, "Implementation of InfoMax ICA algorithm with analog CMOS circuits," in Proc. Int. Workshop Independent Compon.Anal. Blind Signal Separation, Dec. 2001, pp. 70–73.

[9] Z. Li and Q. Lin, "FPGA implementation of Infomax BSS algorithm with fixed-point number representation," in Proc. Int. Conf. Neural Netw.Brain, 2005, vol. 2, pp. 889–892.

[10] 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., Sep. 2004, pp. 3257–3260.

[11] M. Ounas, R. Touhami, and M. C. E. Yagoub, "Low cost architecture of digital circuit for FPGA implementation based ICA training algorithm of blind signal separation," in Proc. Int. Symp. Signals, Syst. Electron., 2007, pp. 135–138.

[12] K. K. Shyu, M. H. Lee, Y. T. Wu, and P. L. Lee, "Implementation of pipelined FastICA on FPGA for real-time blind source separation," IEEETrans. Neural Netw., vol. 19, no. 6, pp. 958–970, Jun. 2008.

[13] A. Acharyya, K. Maharatna, J. Sun, B. M. Al-Hashimi, and S. R. Gunn, "Hardware efficient fixed-point VLSI architecture for 2D KurtoticFastICA," in Proc. Eur. Conf. Circuit Theory Des., Aug. 23–27, 2009, pp. 165–168.

[14] C. K. Chen et al., "A low power independent component analysis processor in 90 nm CMOS technology for portable EEG signal processing systems," in Proc. IEEE Int. Symp. Circuits Syst., May 15–18, 2011, pp. 801–804.

[15] H. Qi and X. Wang, "Comparative study of VLSI solutions to independent component analysis," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 548–558, Feb. 2007.