

FPGA Implementation of 4-Channel ICA for On-line EEG Signal Separation

Wei-Chung Huang¹, Shao-Hang Hung¹, Jen-Feng Chung^{1,2},
Meng-Hsiu Chang¹, Lan-Da Van², and Chin-Teng Lin^{1,2}

¹Department of Electrical and Control Engineering, National Chiao Tung University,

²Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C
Email: jeffjaykimo@yahoo.com.tw

Abstract—Blind source separation of independent sources from their mixtures is a common problem for multi-sensor applications in real world, for example, speech or biomedical signal processing. This paper presents an independent component analysis (ICA) method with information maximization (Infomax) update applied into 4-channel one-line EEG signal separation. This can be implemented on FPGA with a fixed-point number representation, and then the separated signals are transmitted via Bluetooth. As experimental results, the proposed design is faster 56 times than soft performance, and the correlation coefficients at least 80% with the absolute value are compared with off-line processing results. Finally, live demonstration is shown in the DE2 FPGA board, and the design is consisted of 16,605 logic elements.

Index Terms—Bluetooth, fixed-point, ICA, biomedical signal, blind source separation, multi-sensor, information maximization.

I. INTRODUCTION

IN recent years, Independent Component Analysis (ICA) has been proved as a powerful algorithm to solve blind source separation (BSS) [1] problems in a variety of signal processing applications such as speech [2], image, or biomedical signal processing. Especially biomedical signals, which are different signal sources from organs such as brain, heart, or muscles, push the ICA algorithm to process more channels than speech or image applications. However, the characteristic of general ICAs is limited to only process off-line and enormous data. On clinic, this cannot assist doctors in real-time diagnosis. Thus, more researches focus on on-line and faster ICA from points of view on software or hardware implementation.

Several FPGA implementations of the ICA algorithm have been proposed in succession. In 2002, Scatter and Charayaphan [3] implemented the ICA-based BSS algorithm on Xilinx Virtex E that contains 0.6 million logic gates. Du and Qi [4] proposed an FPGA implementation of parallel ICA on a pilchard board in 2004. Charoensak and Sattar [5] in 2005 proposed FPGA design of real-time ICA-based BSS with software solution, e.g., MATLAB Simulink, to translate the high-level language into hardware description language (HDL). Pipelined FastICA [6] using the hardware floating-point arithmetic units to increase the numbers precision was proposed in 2008. To accelerate ICA computation, they are designed by hardware solution.

The computing time relationship between a conventional off-line ICA and on-line ICA is shown in Fig. 1. As can be seen, on-line ICA can improve data throughput. Because off-line ICA is not suitable for real-time computation, to achieve this target and BSS solution, information maximization (Infomax) update [7, 8] integrated into on-line ICA (called Infomax ICA) has been proposed. But, complicated mathematics of Infomax ICA is hard to implement with VLSI technology. Therefore, the modified algorithm is presented and realized by a new effective computing unit and memory scheduling in this paper.

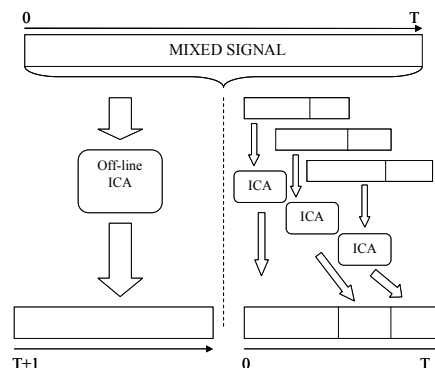


Fig. 1. The time relationship between off-line and on-line ICA processing.

This paper is organized as follows. The Infomax theory and system level design are introduced in Section II. Section III describes FPGA implementation of ICA. The experimental results and discussions are presented in Section IV, and conclusions are made in the last section.

II. THE INFOMAX ICA AND SYSTEM LEVEL DESIGN

A. The Infomax ICA Theory

Most of BSS researches so far focus on the case of mixtures. A linear mixture model is assumed:

$$x(t) = A \times s(t), \quad (1)$$

where $s(t)$ is the vector of sources at instant t , A is the mixing matrix, and the observed vector of mixtures. Fig. 2 shows a single layer feed-forward neural network to represent a mixture model. Bell and Sejnowski [7] proposed to learn the separating matrix W by minimizing the mutual information between components of $y(t) = g(u(t))$, where g is a nonlinear function approximating the cumulative density function (CDF) of the

sources. They had formulated the BBS algorithm in terms of information maximization.

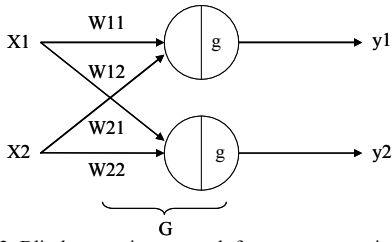


Fig. 2. Blind separation network for two source mixtures.

When a network with an input vector x , a weight matrix W , and a nonlinearly transformed output vector, $y = g(u)$, where $u = Wx$, is considered, the information transmitted by the mapping is the mutual information between the input and output as

$$I[x, y] = H[y] - H[y | x]. \quad (2)$$

Equation (2) can be differentiated as follows, with respect to a parameter, w , involved in the mapping from x to y :

$$\frac{\partial}{\partial w} I(x, y) = \frac{\partial}{\partial w} H(y). \quad (3)$$

The joint entropy of the outputs is

$$H(y) = -E[\ln P(y)] = E[\ln |J|] - E[\ln P(x)]. \quad (4)$$

Weights can be adjusted to maximize $H(y)$. As before, they only affect the $E[\ln |J|]$ term in Eq. (4).

$$\Delta W \alpha \frac{\partial H(y)}{\partial W} = \frac{\partial}{\partial W} \ln |J| = \frac{\partial}{\partial W} \ln |\det W| + \frac{\partial}{\partial W} \ln \prod_{i=1}^n |y'_i|. \quad (5)$$

The resulting learning rules are familiar in Eq. (6).

$$\Delta W \propto [W^T]^{-1} + (1 - 2y)x^T. \quad (6)$$

But this learning rule is too complex to calculate because of the inverter matrix. Multiplied by $W^T W$ change the rescale of the rule, the new learning rules as

$$\Delta W = (I + (1 - 2y)u^T)W = (I + \varphi(u)u^T)W. \quad (7)$$

Thus, the simplification much uncomplicated than before, and this learning rule is suitable to separate blind sources. The update rule for W in discrete time $t < t+1$ is defined as

$$W(t+1) = W(t) + \Delta W. \quad (8)$$

B. System Level Design

The computation diagram of ICA training model is shown in Fig. 3. Three main computing units include the ICA optimal method, accumulation of the weight-updated convergence, and the result output.

Before specification definition, it is necessary to analyze the process of data stream. First, the sampling rate is set to 64 Hz. According to the data streaming, put 512-point data into the ICA model with growing data. Due to updating 128-point data in 2 seconds, the 128-point result is regarded as a set. The processing concept is illustrated in Fig. 4.

ICA MODEL:

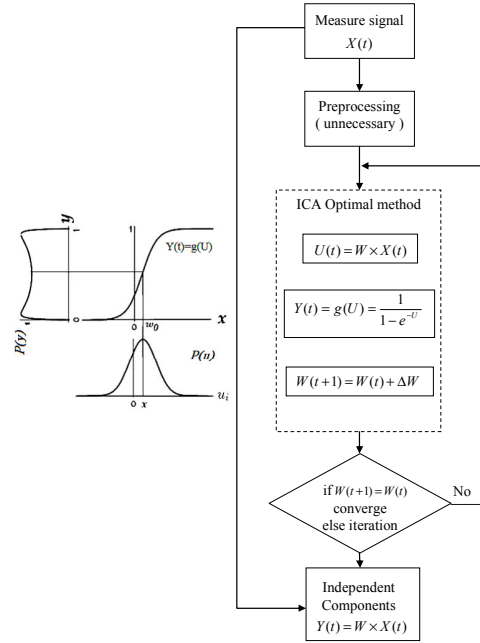


Fig. 3. The flowchart implementation for the on-line ICA learning algorithm.

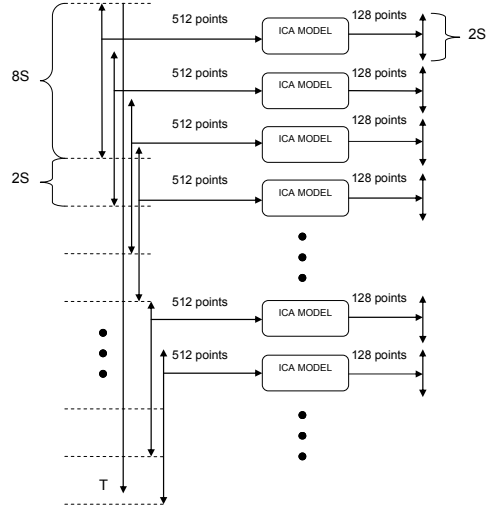


Fig. 4. Illustration of time process in on-line ICA.

III. FPGA IMPLEMENTATION

After the system simulation via software, the weight update and memory access time is measured by profile command that records information about once recursive time shown in Fig. 5.

As can be seen, the execution time with software simulation is not fast enough to achieve on-line signal processing. So, we derive from speed requirement of the overall system as

$$\text{core_speed} = \text{sample_rate} \times \text{train_step} \times (\text{w_update} + \text{converge_decision}) \quad (9)$$

In Eq. (9), the core speed should be at least 68 MHz for on-line execution with 128 times training. However, for the overall system, the main architecture shown in Fig. 6 is divided into three parts as Infomax operation circuit, system control circuit, and interface control circuit.

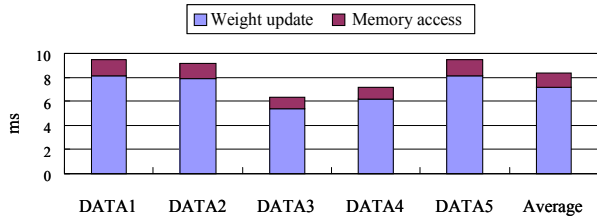


Fig. 5. Execution time with MATLAB results.

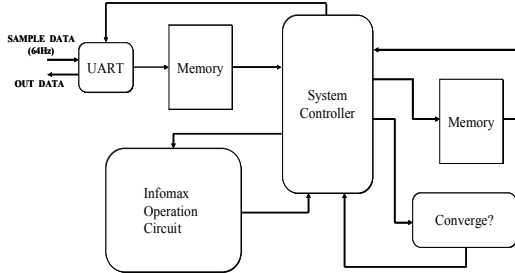


Fig. 6. Hardware architecture of ICA.

A. Infomax Operation Circuit

Both stability and high precision are the properties of the recursive operation circuit. In order to reduce errors in the iteration loop, a precision symmetrical non-linear piecewise look-up table is designed such that the root mean square error (RMSE) is enough small. Besides, the part of complex weight updating is simplified by deep pipeline design.

As the result, the computing unit shown in Fig. 7 consumes 8,192 cycles to find a new weight with gradient information update. The cycle expression is represented as Eq. (10).

$$operation_cycle = \frac{core_speed}{sample_rate * iteration} \quad (10)$$

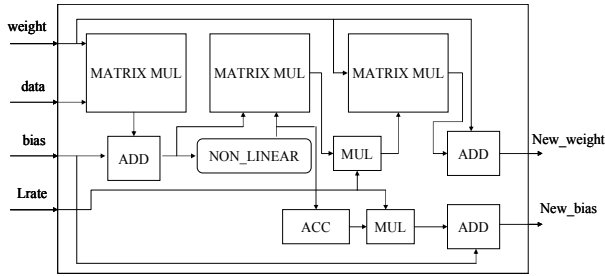


Fig. 7. Integrated computing unit of ICA.

If the maximum number of training is 128, it may cost 13 ms totally and less than sample time 16 ms. Fig. 8 shows the FPGA execution time of recursive circuit compared with software.

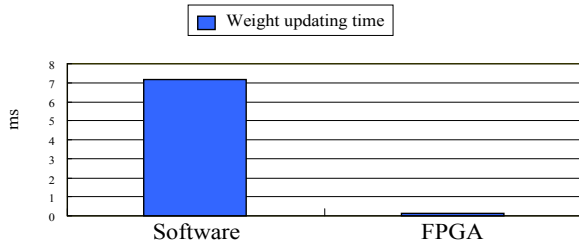


Fig. 8. Time consumption of Weight calculation.

B. Control Circuit

The main controller shown in Fig. 9 consists of an asynchronous memory controller and an ICA controller. It mainly receives data stream from UART and then decodes control signals for various modules when ICA performs processing. However, external data would be sent into internal memory by an interrupt way. Here a data counter judged by the amount of data is placed, and then the main controller can send complete signals to the correct path. The controller is designed by two recursive circuits and write-back technology to reduce half the amount of memory accessing time. This is an effective way of memory scheduling.

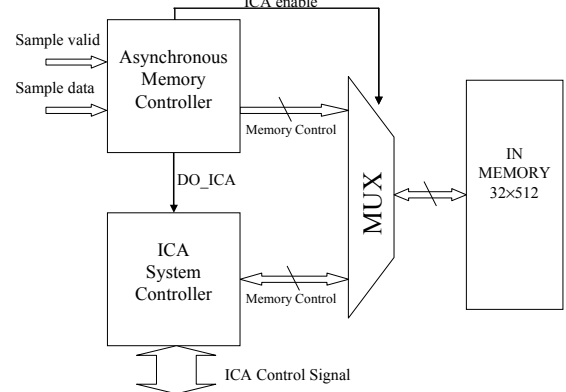


Fig. 9. Architecture of the main controller.

C. Interface Control Circuit

The serial interface shown in Fig. 10 is implemented by RS-232 standard with baud rate at 115,200 bps. The controller architecture consists of three parts such as a transceiver, header controller and encoder, and asynchronous FIFO.

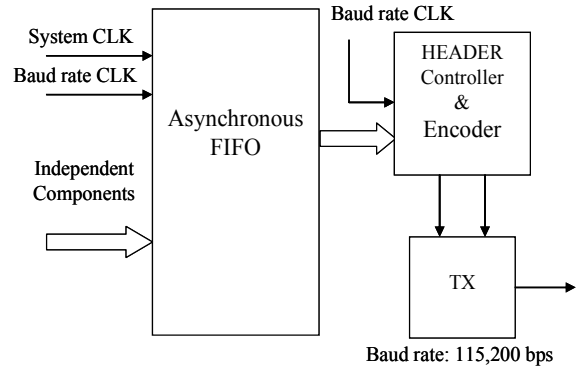


Fig. 10. Header controller architecture.

For the header controller, the encoder is involved to encode the fixed-point hexadecimal value to 8-bit integer. The serial protocol composes of a header "FF" and 4-channel data. For example, the protocol could be "FF VV XX YY ZZ".

Finally, the specification of the proposed Infomax ICA is shown in Table I. The total of logic gets is about 315,495.

TABLE I
HARDWARE SPECIFICATION

Operating Frequency	68 MHz
Sample Rate	64 Hz
Gate Counts	315,495
Operate Voltage	3.3 volts
Transmission Interface	RS-232 115,200 bps
Embedded Memory (M4K)	24,576 bits
ADC Resolution	8 bits

IV. EXPERIMENTAL RESULTS AND DEMONSTRATION

In order to verify that the function is capable of separating super-Gaussian signals, first the experiment creates four mixed signals with a linear mixed matrix. Fig. 12 shows 4-channel mixed signals and ICA separation results, respectively. The correlation results compared between on-line and off-line ICA are shown in Fig. 13. Because the on-line process collects small amount of information than the off-line process, the correlation of on-line system may be different from off-line. But, at least more than 80% correlation results can be accepted.

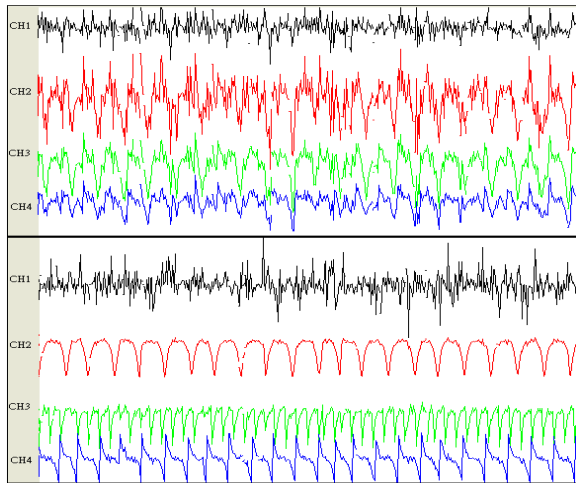


Fig. 12. 4-channel mixed signals and ICA separation results.

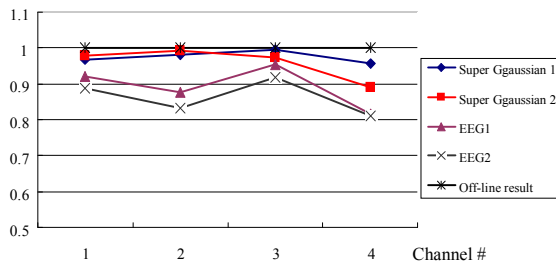


Fig. 13. Online ICA correlation compared with off-line ICA.

There are few studies about the real-time implementation of ICA which has been implemented on FPGA. In general, in terms of the numbers of channel, gate counts, and speed are compared with our proposed design in Table II. Finally, the prototype demonstration has been completed a 4-channel EEG head band, ICA DSP processing, Bluetooth wireless transmission, and GUI display.

TABLE II
COMPARISON WITH OTHER ICA DESIGNS

Name	Ref [3]	Ref [4]	Ref [5]	Ref [6]	This work
Application	speech	image	speech	speech	EEG
Channel	2	N/A	2	2	4
Gate counts (million)	0.6	0.226	0.1	N/A	0.315
Speed (MHz)	20	20.1	71.2	50	68

V. CONCLUSIONS

The 4-channel on-line ICA accompanied with flexible communication interface for real EEG signal separation has been presented in this paper. The proposed integrated mathematics architecture can allow high-speed and real-time biomedical signal separation with Infomax ICA at sampling rate 64 Hz.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council (NSC) Grant NSC-96-2220-E-009-038.

REFERENCES

- [1] T.W. Lee, *Independent Component Analysis - Theory and Applications*, Kluwer Academic Publishers, 1998.
- [2] C. M. Kim and S. Y. Lee, "A digital chip for robust speech recognition in noisy environment," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1089-1092, 2001.
- [3] F. Sattar and C. Charayaphan, *Low-cost design and implementation of an ICA-based blind source separation algorithm*, 15th Annual IEEE International ASIC/SOC Conference, pp.15-19, 2002.
- [4] H. Du and H. Qi, "An FPGA implementation of parallel ICA for dimensionality reduction in hyperspectral images," in *Proc. IEEE Int. Symp. Geosci. Remote Sens.*, vol. 5, pp. 3257-3260, Sep. 2004.
- [5] C. Charoensak and F. Sattar, "A single-chip FPGA design for real-time ICA-based blind source separation algorithm," in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol. 6, pp. 5822-5825, May 2005.
- [6] Kuo-Kai Shyu and Ming-Huan Lee, "Implementation of pipelined FastICA on FPGA for real-time blind source separation," *IEEE Transactions on Neural Networks*, vol. 19, June 2008.
- [7] A. J. Bell and T. J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," *Neurocomputing*, vol. 7, pp. 1129-1159, 1995.
- [8] T. W. Lee, M. Girolami, and T. J. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources," *Neurocomputing*, vol. 11, pp. 606-633, 1999.