35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

An Efficient VLSI Implementation of On-line Recursive ICA Processor for Real-time Multi-channel EEG Signal Separation

Wei-Yeh Shih, Jui-Chieh Liao, Kuan-Ju Huang, Wai-Chi Fang, *Fellow*, *IEEE*, Gert Cauwenberghs, *Fellow*, *IEEE*, and Tzyy-Ping Jung, *Senior Member, IEEE*

***Abstract*—This paper presents an efficient VLSI implementation of on-line recursive ICA (ORICA) processor for real-time multi-channel EEG signal separation. The proposed design contains a system control unit, a whitening unit, a singular value decomposition unit, a floating matrix multiply unit and, and an ORICA weight training unit. Because the input sample rate of the ORICA processor is 128 Hz, the ORICA processor should produce independent components before the next sample is input in 1/128 s. Under the timing constraints of commutating multi-channel ORICA in real time, the design of the ORICA processor is a mixed architecture, which is designed as different hardware parallelism according to the complexity of processing units. The shared arithmetic processing unit and shared register can reduce hardware complexity and power consumption. The proposed design is implemented used TSMC 90nm CMOS technology with 8-channel EEG processing in 128 Hz sample rate of raw data and consumes 2.827 mW at 50 MHz clock rate. The performance of the proposed design is also shown to reach 0.0078125 s latency after each EEG sample time, and the average correlation coefficient between the original source signals and extracted ORICA signals for each 1s frame is 0.9763.**

I. INTRODUCTION

Electroencephalogram (EEG) is a noninvasive tool for measuring the electrical activity in the brain, and to date has found many useful applications in the medical, consumer and entertainment industries. Brain-computer interface systems allow people suffering from severe motor disabilities to control external devices without moving by using EEG signals. However EEG signals are very weak, and thus often contaminated by various noise such as eye movement, EMG and electrical noise from nearby instruments. Fortunately, this problem can be alleviated by independent component analysis (ICA), which separates artifacts and noise from the measured EEG signals[1]. To date, many ICA algorithms, such as Infomax [2], extended Infomax [3], JADE [4], and FastICA [5] have been proposed. These ICA algorithms are not suitable for online implementation in a real-time setting. Since on-line recursive independent component analysis (ORICA) [6] has a fast convergence rate and satisfactory separation performance which those ICA algorithms could not achieve, it is suitable for online implementation with only a low additional computational load. However, the complexity computation of ORICA is so intense that real-time ORICA analysis in not feasible for a PC-based implementation. Du [7] presented a comparative survey of very large scale integration solutions to ICA. Therefore the VLSI hardware implementation of ORICA is required to achieve real-time ICA analysis. This study proposes an efficient VLSI implementation of ORICA processor for real-time multi-channel EEG signal separation. Under the timing constraints of commutating multi-channel ORICA in real time, the design of the ORICA processor is a mixed architecture, which is designed as different hardware parallelism according to the complexity of processing units. The shared arithmetic processing unit and shared register reduce hardware complexity and power consumption. The design methods of the proposed ORICA processor are provided in this paper. In section II, the algorithms adopted in the system are described. In section III, the system architecture and design methods are given. Finally, the results and conclusions are provided in sections V and VI.

II. ALGORITHM

This section describes the algorithms adopted in this ORICA processor. Fig. 1 shows the ORICA processing data flow. After EEG raw data X are acquired from front-end control unit, whitening is performed for the uncorrelated vector Z to effectively accelerate the training processing from (1) to (3).

$$Cov(X) = E[X, X^T] \tag{1}$$

$$P = Cov(X)^{-1/2} \tag{2}$$

$$Z = P \times X \tag{3}$$

And then Z is processed to estimate the independent component Y and the unmixed weight W in ORICA training as (4) to (9). Finally, W and Y are delivered to UART unit to produce the result.

$$Y = W \times Z \tag{4}$$

$$k = sign\left(\frac{E\{Y^4\}}{\left(E\{Y^2\}\right)^2} - 3 \right) \tag{5}$$

$$\left\{ \begin{array}{l} k = 1, f = -2\tanh(Y) \\ k = -1, f = \tanh(Y) - Y \end{array} \right\} \tag{6}$$

$$\Delta W = \frac{\lambda}{1-\lambda}[W - \frac{Y \times f^T \times W}{1 + \lambda(f^T \times Y - 1)}] \tag{7}$$

$$W_0 = W + \Delta W \tag{8}$$

$$W = W_0^{-1/2} \times W_0 \tag{9}$$

Wei-Yeh Shih and Wai-Chi Fang are with the Department of Electrical Engineering and Institute of Electronics, National Chiao Tung University, Hsinchu 30010, Taiwan, (e-mail: karlshin211.ee99g@nctu.edu.tw; wfang@mail.nctu.edu.tw).

Gert Cauwenberghs and Tzyy-Ping Jung are with Swartz Center for Computational Neuroscience, Institute for Neural Computation, University of California San Diego, San Diego, California, United States of America.

978-1-4577-0216-7/13/$26.00 ©2013 IEEE

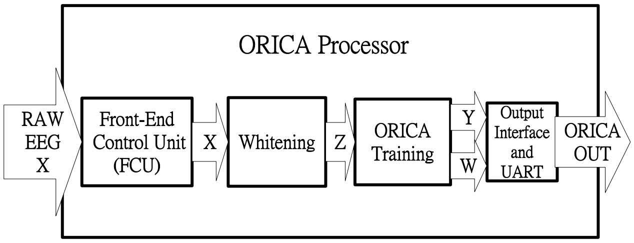


Fig 1. The ORICA processing data flow.

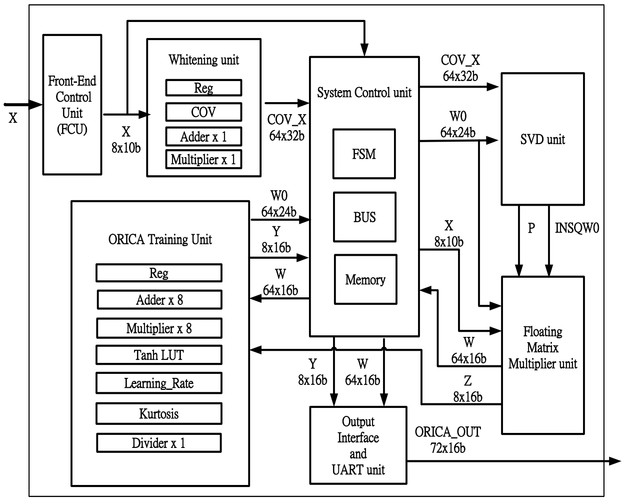


Fig 2. The hardware architecture of the ORICA processor.

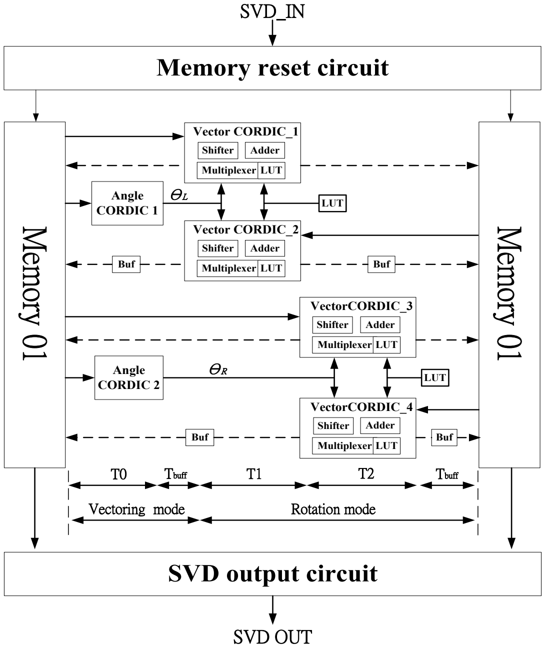


Fig 3. The hardware architecture and timing analysis of an SVD unit.

III. Proposed SVD Processor Architecture And Implementation

A. Online Recursive ICA processor

The hardware architecture of the ORICA processor (shown in Fig. 2) comprises: 1) a system control unit (SCU) for saving and controlling the processing data, 2) a whitening unit (WU) for calculating COV_X, 3) a singular value decomposition unit (SVDU) for calculating eigenvalues and eigenvectors of the covariance matrix and inverse square root matrix, 4) a floating matrix multiplier unit (FMAMU), 5) an ORICA weight training unit (ORICAWTU) for processing Z to estimate the W and Y, and 6) an output interface delivering results ORICA_OUT through the UART unit. The operation is described as following. First, the WU performs the calculation of COV_X which is the mean and covariance of X, and then the COV_X is stored inside the memory. Second, the SCU fetches COV_X from memory and delivers it to the SVDU to obtain whitening matrix P. Third, the FMAMU performs the multiplication of P and X and produces the whitened EEG vector Z. After the calculation of whitening, the ORICAWTU performs the un-whitened weight W0 training and Y by processing Z and W. When ORICA weight training is completed, W0 and Y are stored inside the memory. And then, the SCU fetches Y and W from memory and delivers results ORICA_OUT through the output interface and the UART unit, and SVDU simultaneously calculates whitened weight matrix INSQW0 by processing W0. Finally, the multiplication of W0 and INSQW0 are performed through FMAMU to obtain the W, and W is stored into the memory to update W for the next ORICA processing.

B. SVD unit

The hardware architecture and timing analysis of SVDU are shown in Fig. 3. In order to reduce the latency of SVD operation and avoid extra power consumption, this SVDU replaces a duple-port SRAM with two single-port SRAMs in storage data. First of all, memory reset circuit stores the data, SVD_IN, in different SRAMs. The Angle CORDIC will capture the corresponding elements, which are fetched from Memory_01 to calculate θ_L and θ_R. Then, the specific elements are taken by using Vector CORDIC_1 and Vector CORDIC_2 from both memories at the same time. After the vector CORDIC operation, the SVDU will obtain updated elements on the corresponding origin data. However, in order to avoid the structural hazard during the renewal of memories, this work delayed a few clock cycles to store the updated data by using buffers. Furthermore, in terms of timing analysis, the execution time per iteration of vectoring mode [8], rotation mode and whole SVDU are shown in (10), (11), and (12) respectively for multi-channel ORICA processor.

$$T_{\text{vectoring mode}} = T_0 + T_{\text{buff}} \tag{10}$$

$$T_{\text{rotation mode}} = T_1 + T_2 + T_{\text{buff}} \tag{11}$$

$$T_{\text{total}} = C_2^8 \times T_{\text{vectoring mode}} \times 16 \times T_{\text{rotation mode}} \tag{12}$$

C. Floating matrix multiply unit

The FMAMU is designed for hardware sharing resource of the system to narrow down chip area and cost. To improve the accuracy of multi-channel ORICA processing, the FMMU is employed to avoid fraction truncation by using IEEE 754 format. Each sub-module should transform the fixed point into the standard floating point format before calculation. After the multiplication is completed, the FMAMU transforms floating into fixed point and sends updated data to the corresponding sub-module.

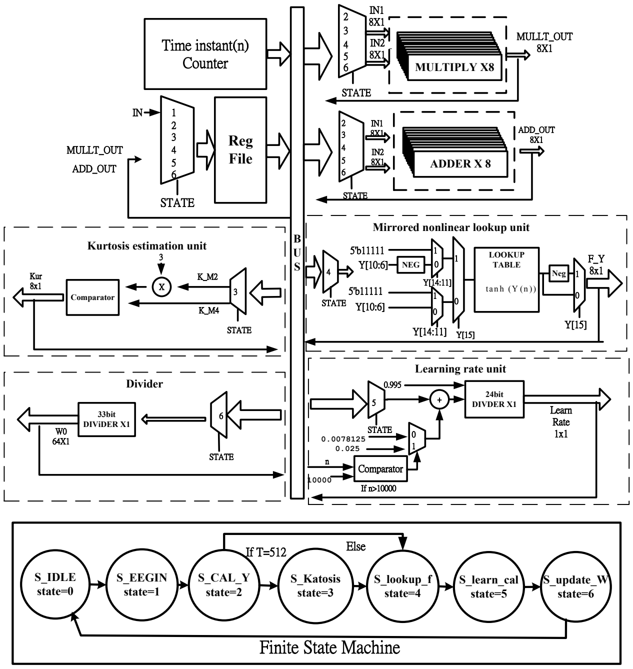


Fig. 4. The architecture of ORICAWTU.

D. *ORICA Weight training unit*

The main purpose of the ORICAWTU is to estimate the Y and W. Fig. 4 shows the hardware architecture and different running states controlled by a finite state machine of the ORICAWTU. The ORICAWTU employs one shared divider unit, one shared multiplier array, one shared adder array, a mirrored nonlinear lookup unit, a kurtosis estimation unit, and a learning rate unit. The calculation of ORICA training requires many adders and multipliers, so one shared multiplier array is composed of 8 16-bit scalar multipliers and one shared adder array is composed of 8 32-bit scalar adders. The mirrored nonlinear lookup unit is designed to minimize the ROM size for the lookups of non-linear function tanh(Y). The kurtosis estimation unit identifies the distribution of Y and the learning rate unit determines the convergence speed of ORICA training. Since the processing units and operation flow are well arranged, the ORICAWTU can reach the highest performance and real time processing.

IV. Result and Comparison

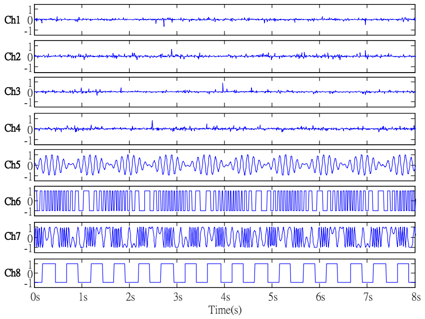


Fig. 5a. Source signal.

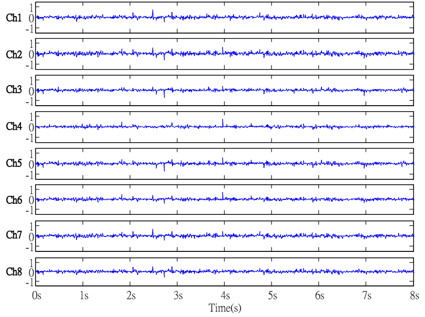


Fig. 5b. Mixed signal.

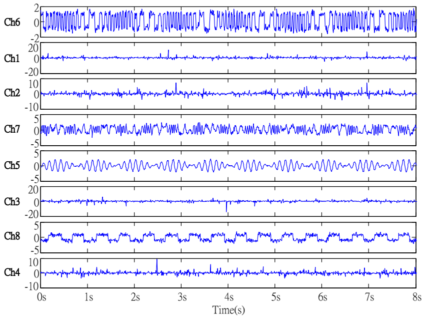


Fig. 5c. Extracted ORICA signal.

The simulated source signal (shown in Fig. 5a) contains four independent super-Gaussian signals and four independent sub-Gaussian signals, and the maximum correlation between each source signal is 0.0032. To verify the performance of the proposed design, the simulated mixed signal (shown in Fig. 5b) which is a mixture of source signal and random matrix performs the extracted ORICA signal (shown in Fig. 5c) through the proposed ORICA processor. The average correlation coefficient between the source signal and extracted ORICA signal for each 1s frame is 0.9763. To verify the performance of real EEG signal separation, the raw EEG recorded signals (shown in Fig. 6a) were collected from 8 scalp electrodes placed according to the international 10-20 system. The separation result of the recorded raw EEG is shown in Fig. 6b. It can be seen that eye blink artifacts are exactly separated by using the proposed ORICA processor. The real chip and the silicon layout of the proposed ORICA processor are shown in Fig. 7. The ORICA processor is fabricated using TSMC 90nm CMOS technology. The chip gate count, core area and operating frequency of the proposed ORICA processor are 0.269 million, 1200 x 1200 μm^2 and up to 50MHz, respectively. The performance and processing results of the proposed design are also shown to reach 0.0078125 s latency after each EEG sample time. The chip is tested by Agilent 9300 and the power consumption is 2.827mW. The comparisons of this study with others are given in Table I. Considering the gate count per channel, the proposed ORICA processor has a lower gate count than [9]. Similarly, the correlation of this work is higher than [9]. Comparing with [10] which has also implemented an eight-channel ICA processor, the gate count and power consumption of this work are both less than the work of [10]. Moreover, the proposed ORICA processor has less output latency than the works in [9]-[10], so it can efficiently achieve on-line processing in real time.

This work achieves higher correlation and lower power consumption than the previous work [11] for effective shared

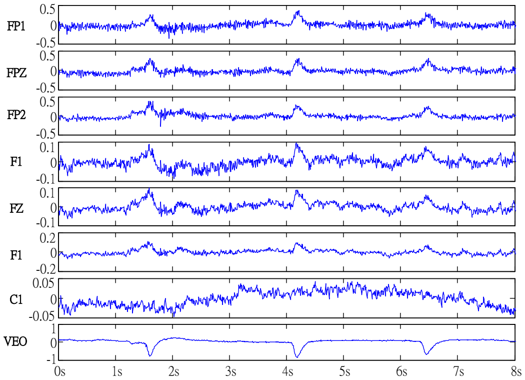


Fig. 6a. The raw EEG recorded signals.

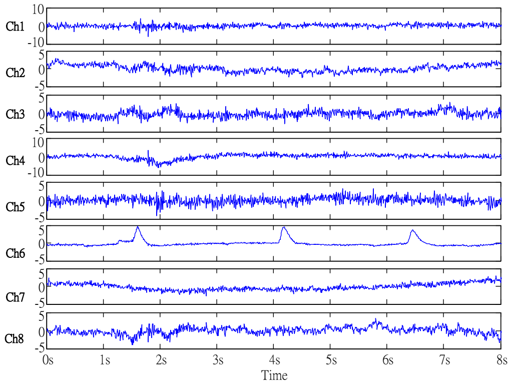


Fig. 6b. The separation result of raw EEG recorded signals.

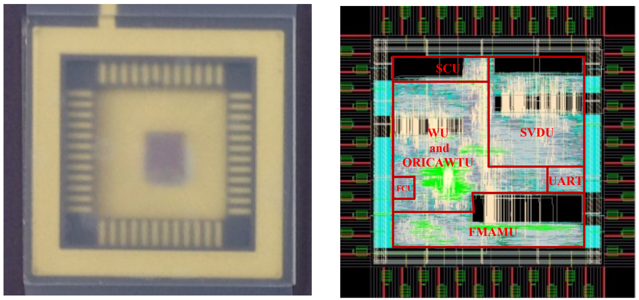
arithmetic processing unit and data arrangement.

V. Conclusion

This paper presents an efficient VLSI implementation of the ORICA processor for real-time multi-channel EEG signal separation. The proposed deign is implemented used TSMC 90nm CMOS technology in 128 Hz sample rate of raw data and consumes 2.827 mW at 50 MHz clock rate. The performance of the proposed design is also shown to reach 0.0078125 s latency after each EEG sample time, and the average correlation coefficient between the original source signals and extracted ORICA signals for each 1s frame is 0.9763.

Acknowledgment

This work was supported in part by the National Science Council of Taiwan, R.O.C., under grant NSC101-2220-E-009-049 and NSC101-2221-E-009-169-MY2. The authors would also like to express their sincere appreciation to the National Chip Implementation Center for chip fabrication and testing service.



(a) (b)

Fig. 7. (a) The real chip. (b) The silicon layout of the proposed design.

TABLE I. Comparison With Other On-line ICA Implementations

	Chen [9]	**Van [10]**	**Shih [11]**	**This work**
Technology	UMC 90nm	UMC 90nm	TSMC 90nm	TSMC 90nm
Channel	4	8	8	8
Prepocessing	A/V	A/V	A/V	A/V
Core Size (μm^2)	760x 760	1221x 1218	800x 800	1200x1200
Gate Count (million)	0.199	0.272	0.172	0.269
Output latency(s)	0.25	0.29	0.0078125	0.0078125
Power Consumption (mW)	0.53	16.35	4.18	2.827
Correlation	0.9044	>0.95	0.9583	0.9763
Operating Freq. (MHz)	5	100	50	50

References

[1] R. Vigario, “Extraction of ocular artifacts from EEG using independent component analysis,” *Electroencephalogr. Clin. Neurophysiol.*, vol. 103, pp. 395-404, 1997.

[2] 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.

[3] 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,” *Neural Computation*, Vol.11,pp. 417-441, 1999.

[4] Cardoso, J.F.; Souloumiac, A.; ,”Blind beamforming for non-Gaussian signals,” Radar and Signal Processing, *IEE Proceedings F* , vol.140, no.6, pp.362-370, Dec 1993.

[5] A. Hyvärinen; and E. Oja.; “A Fast Fixed-Point Algorithm for Independent Component Analysis,” Neural Computation, Vol. 9, No. 7, pp. 1483-1492, 1997.

[6] Akhtar, M.T.; Tzyy-Ping Jung; Makeig, S.; Cauwenberghs, G.; “Recursive independent component analysis for online blind source separation,” Circuits and Systems (ISCAS), 2012 IEEE International Symposium on , vol., no., pp.2813-2816, 20-23 May 2012.

[7] H. Du, 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.

[8] Meher, P.K.; Valls, J.; Tso-Bing Juang; Sridharan, K.; Maharatna, K.; , "50 Years of CORDIC: Algorithms, Architectures, and Applications," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.56, no.9, pp.1893-1907, Sept. 2009

[9] Chiu-Kuo Chen, Ericson Chua, Chih-Chung Fu, Shao-Yen Tseng, and Wai-Chi Fang, “ A Hardware-Efficient VLSI Implementation of a 4-Channel ICA Processor for Biomedical Signal Measurement,” *in Proc. IEEE Int. Conf. on Consumer Electronics*, Jan. 2011, pp. 607 – 608.

[10] Lan-Da Van, Di-You Wu, and Chien-Shiun Chen, “Energy-Efficient FastICA Implementation for Biomedical Signal Separation,” *IEEE Trans. Neural Networks*, vol.22, no.11, pp.1809-1822, Nov. 2011.

[11] Wei-Yeh Shih; Kuan-Ju Huang; Chiu-Kuo Chen; Wai-Chi Fang; Cauwenberghs, G.; Tzyy-Ping Jung, "An effective chip implementation of a real-time eight-channel EEG signal processor based on on-line recursive ICA algorithm," *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE* , vol., no., pp.192,195, 28-30 Nov. 2012.