

# PARALLEL BIT-STREAM NEUROHARDWARE FOR BLIND SEPARATION OF SOURCES

*Karsten Fanghänel, Kuno Köllmann and Hans Christoph Zeidler*

Universität der Bundeswehr Hamburg, Holstenhofweg 85, 22043 Hamburg, Germany  
Karsten.Fanghaenel@UniBw-Hamburg.DE

*Ralf Pleßmann and Karl-Ragnar Riemschneider*

C. Plath, Gotenstraße 18, 20097 Hamburg, Germany

## ABSTRACT

In this paper the use of digital stochastic computing is proposed to realize a recurrent network for blind separation of undelayed linearly superposed signals into their original components. The stochastic representation of signals allows to design very simple digital processing elements to implement all the operations necessary for the algorithm of Herault and Jutten. A first hardware implementation has been designed and the experimental results are presented.

## 1. INTRODUCTION

### 1.1. Blind Separation of Sources

Blind separation of sources is a basic and important problem in signal processing. Various methods have been introduced to solve problems such as the "cocktail-party" one. For example, contrast functions [1], cumulants [2], or higher-order moments generated by nonlinear functions [3] are used. Likewise hardware implementations have been proposed, but still problems arise when integrating a solution, expanding the design, or concerning distorted signals [4, 5].

For a hardware implementation it was decided to use a separation method based on the learning algorithm of Herault and Jutten [6, 7]. To simplify its explanation let us assume an array of only two sensors measuring instantaneous linear superpositions<sup>1</sup>  $y_i$  of two unknown independent sources  $x_i$ ,  $i = 1, 2$ :

$$\vec{y} = \mathbf{A}\vec{x} \iff \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

The object is to extract two output signals  $s_i$  from the sensor signals  $y_i$  in a way that each output signal is proportional only to one source signal. The problem is to find a learning algorithm without a priori knowledge of the mixing matrix  $\mathbf{A}$ . The output signals are in general described by the following equation

$$\vec{s} = \mathbf{W}\vec{y} \iff \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 1 & -w_{12} \\ -w_{21} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

<sup>1</sup>For clarity reasons the time dependence of all signals will be left out.

where  $\mathbf{W}$  is the weight matrix and  $w_{12}$  and  $w_{21}$  are two synaptic weights. Two different theoretical solutions of the synaptic weights are possible:

1.  $w_{12} = a_{12}/a_{22}$  and  $w_{21} = a_{21}/a_{11}$
2.  $w_{12} = a_{11}/a_{21}$  and  $w_{21} = a_{22}/a_{12}$

To realize an appropriate learning algorithm, Herault and Jutten proposed a learning rule of the form

$$\frac{dw_{12}}{dt} = \mu \phi(s_1) \psi(s_2) \quad \text{and} \quad \frac{dw_{21}}{dt} = \mu \phi(s_2) \psi(s_1),$$

where  $\mu > 0$  is an arbitrarily adaptable gain factor.  $\phi$  and  $\psi$  are odd nonlinear functions which contain odd power terms only. Presented examples are

$$\phi(x) = x^3 \quad \text{and} \quad \psi(x) = \tanh(x).$$

An overview of the structure of the separation net is shown in Figure 1.

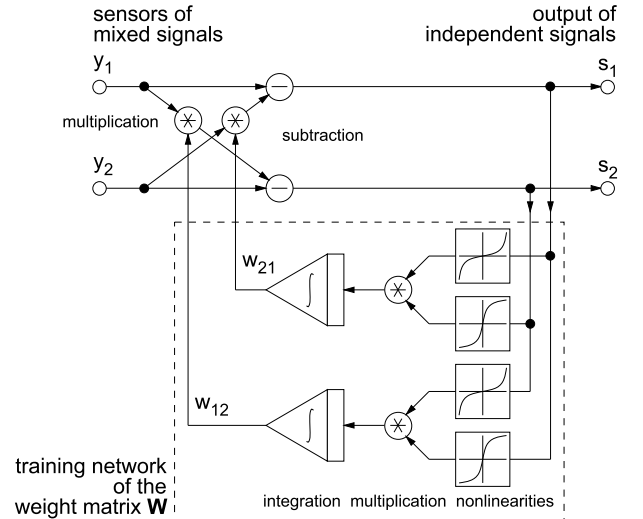


Figure 1: Structure of the separation net in the case of two signals

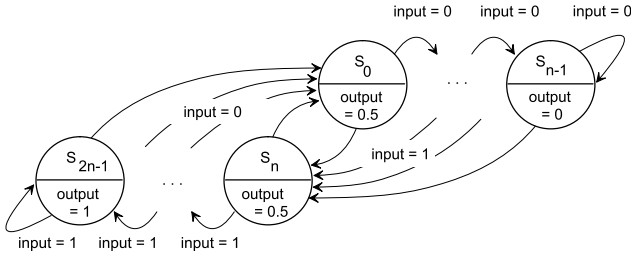


Figure 2: Scheme of the automaton generating the nonlinear function  $\phi$

## 1.2. Bit-Stream Techniques

The goal is to use digital stochastic computing to implement the algorithm described. In consequence, the information carriers between the processing elements are not digital numbers but probabilities in binary representation. Such components are known as stochastic processing elements. It has been shown that the exponential dependency of accuracy and time needed can be kept under control [8, 9]. In contrast to common bit-stream techniques the arrangement used is characterized, especially, by a signed zero-symmetric coding leading to a very simple multiplication and to simple automata for the learning functions. These stochastic processing elements allow to implement the inherent parallelism of the algorithm using digital VLSI technology.

## 2. BASIC ELEMENTS

To combine the bit-stream technique with blind source separation it must be shown how to construct each processing element presented in Figure 1. Circuits are required which are able to integrate, to multiply, to add, to invert and to form the nonlinear transmission functions.

### 2.1. Coding

It is possible to transform signed values in a limited range into a probability using a suitable coder [8, 9]. In each clock-cycle the coder compares the signed input value with a random value of the same bit length and generates the next bit of the bit-stream. Thus, the expected value of the probability to find a binary "1" in the bit-stream corresponds to the signed input value. The more clock-cycles are used, the higher is the accuracy of the bit-stream representation. The silicon implementation of the coder uses a pseudorandom generator consisting of a feedback 28 bit-shift-register with 52 four-stages-parities.

### 2.2. Arithmetic Operations

As the most frequent and conventionally limiting operation during signal processing the multiplication of two probabilities in the bit-stream representation can simply be implemented by combining both streams in a digital equivalence [9].

In contrast to the multiplication it cannot be guaranteed that during an addition the limited range of probabilities will not be left. A solution is to replace the addition with

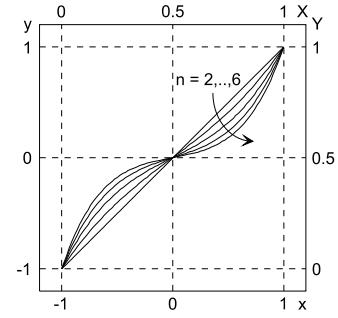


Figure 3: Plot of nonlinear function  $\phi$

the calculation of the arithmetic mean. In hardware this operation can be implemented with a multiplexer which is randomly toggled between the two inputs.

The negation may easily be implemented with an inverter.

### 2.3. Learning Functions

The algorithm of Herault and Jutten requires two learning functions which have to contain odd power terms only. The way proposed to set up these functions is to react on sequences of identical bits in the bit-stream, that means to make use of higher order statistical characteristics in the bit-stream. An automaton counts identical bits consecutively and switches the output accordingly.

The first automaton generates a function which has a curve similar to the cubic function  $\phi$  (Figure 2). To calculate the transmission function of the automaton, it is useful to have a look at the case of stationary input and unlimited observation time. Then the automaton can be described like a Markovian process as follows<sup>2</sup>

$$\vec{P} = \mathbf{M}(X, n) \cdot \vec{P} \quad \text{with} \quad \sum_{i=0}^{2n-1} p_i = 1.$$

The state equation can be evaluated as a transmission function of probabilities

$$Y = \frac{1}{2} (X^n - (1 - X)^n + 1).$$

This function can be transformed into a function of signed coded input values

$$y = \frac{1}{2^{n-1}} \sum_{\substack{i=1 \\ i \text{ odd}}}^n \binom{n}{i} x^i.$$

As mentioned the transmission function contains odd power terms only. The curve for different lengths of bit-sequences is shown in Figure 3.

|                        |                                            |
|------------------------|--------------------------------------------|
| <sup>2</sup> $\vec{P}$ | vector of automaton's status probabilities |
| $p_i$                  | component of vector $\vec{P}$              |
| $\mathbf{M}(X, n)$     | matrix of transition probabilities         |
| $X$                    | input probability                          |
| $Y$                    | output probability                         |
| $x$                    | signed coded input value                   |
| $y$                    | signed coded output value                  |
| $n$                    | sequence length of identical input bits    |

The function  $\psi$  has been realized by a second automaton which generates a sigmoid-similar function. The design of this automaton is described in detail in [8]. The implementation of both automaton schemes in digital hardware has been carried out by using sequential networks (synchronous finite state machines).

#### 2.4. Storing Elements

In order to get two signals separated two storing elements are needed to accumulate the synaptic weights  $w_{12}$  and  $w_{21}$ . In addition, these two elements have to integrate the incoming bit-streams. An integrative element of the stochastic arithmetic is the simplest arrangement to fulfil the requirements [10]. Furthermore, an adaptive element is connected in series to the integrative element to implement a pulse term (also called momentum) which has been proved to accelerate the learning process [11].

### 3. NEUROHARDWARE IMPLEMENTATION

The hardware to be implemented has to fulfil at least two requirements:

- to show the feasibility of combining blind source separation and digital bit-stream techniques
- to offer the opportunity to tune hardware parameters, e.g. to optimize the learning behavior

A PC board has been developed with programmable parameters and several test functions (Figure 4). One self-designed VLSI neuro chip (1.0  $\mu\text{m}$  digital CMOS, 10 mm<sup>2</sup> core) and six complex programmable logic devices (AMD's MACH circuits) contain all the necessary elements and the bus interface.

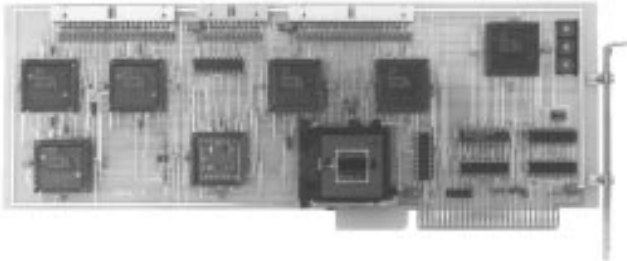


Figure 4: Photo of the PC board containing the neuro chip (center, at the bottom)

### 4. EXPERIMENTAL RESULTS

The PC board has been tested and is running up to a frequency of 14 MHz. An exemplary signal separation is illustrated in Figure 7. The two source signals

$$x_1 = \sin(2\pi \cdot 10\text{Hz} \cdot t) \cdot \sin(2\pi \cdot 300\text{Hz} \cdot t) \quad \text{and}$$

$$x_2 = \sin(2\pi \cdot 50\text{Hz} \cdot t)$$

are mixed by the following matrix

$$\mathbf{A} = \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix}.$$

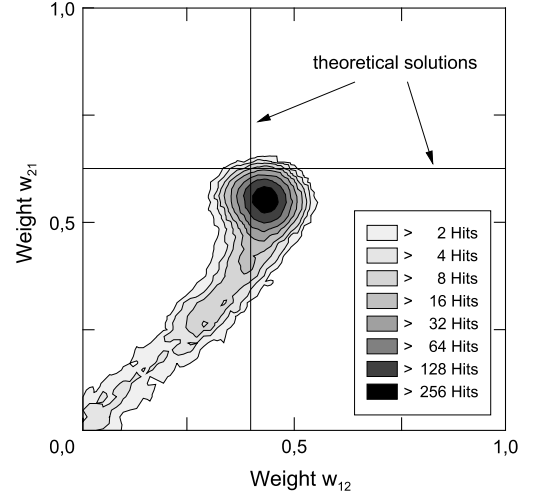


Figure 5: Plot of 50000 pairs of calculated weights during  $2 \cdot 10^8$  hardware clocks (starting with  $w_{12} = w_{21} = 0$ )

The mixed signals are separated quite well but with additional noise as expected from digital stochastic computing. In the graph of the weights (Figure 5 and 6) the noise is clearly visible. Furthermore, the computed weights converge to slightly different values compared with numerical simulations utilizing floating point arithmetic. This is comprehensible since the transformation of the mixed signals into stochastic bit-streams leads to an independent noise portion in the representation which cannot be separated by the algorithm.

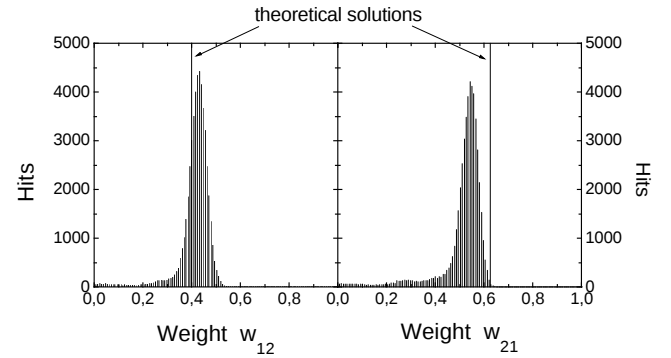


Figure 6: Plot of the distribution of 50000 calculated weights during  $2 \cdot 10^8$  hardware clocks (starting with  $w_{12} = w_{21} = 0$ )

It was possible to achieve good results in signal separation by determining suitable hardware parameters, e.g.:

- 4096 hardware clocks per signal sample
- 12 bit counter length of integrative elements
- 15 bit counter length of adaptive elements
- $n = 2$  for learning function  $\phi(x)$  (see Figure 3)
- $n = 4$  for learning function  $\psi(x)$

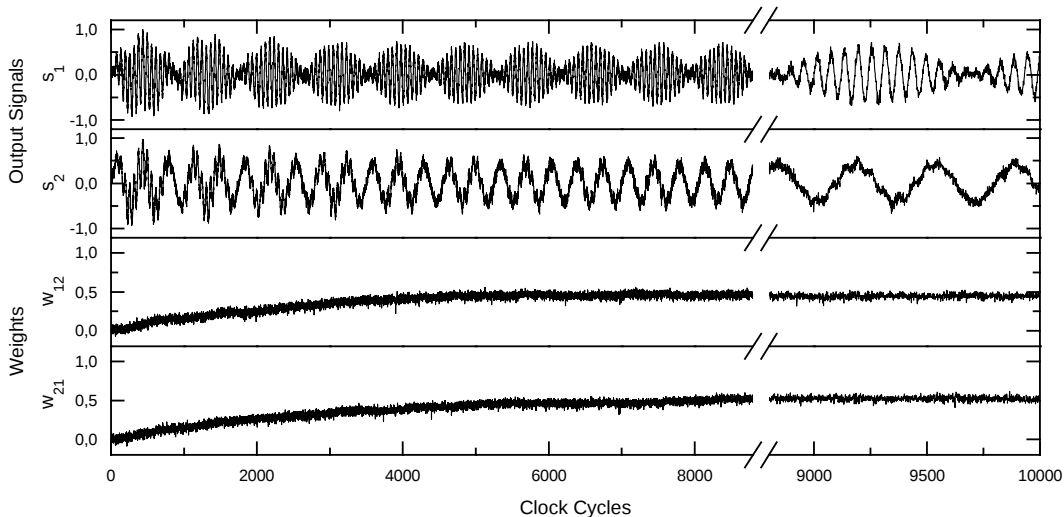


Figure 7: Time dependence of the output signals and the weights. The right side of the diagram is enlarged.

It should be emphasized that only such signals can be separated, where the synaptic weights have a solution in the range of  $[0,1]$ . This can be guaranteed in the case that the sum of all elements of each line of the mixing matrix  $\mathbf{A}$  is equal to 1:

$$\sum_j a_{ij} = 1, \quad \forall i$$

Following Herault and Jutten [6, 7] the output signals theoretically have the form

$$s_i = y_i - \sum_{k \neq i} c_{ik} s_k, \quad \forall i.$$

In this case however, the area of signals to be separated is diminished due to range limitations caused by the digital stochastic computing.

## 5. CONCLUSION

It has been shown that it is feasible to combine the features of digital stochastic computing with the requirements of blind source separation. In the case of two input signals the separation works quite well. Only for high accuracy applications the residual scattering of the weights gives rise to limitations.

Stochastic computing simplifies the processing elements significantly, and, thus, allows to implement the complete separation algorithm in digital VLSI technology. In principle, it is easy to scale the design to more than two input signals. However, the convergence of larger networks has still to be investigated.

## 6. REFERENCES

- [1] Comon, P. *Independent Component Analysis, A New Concept?* Signal Processing, vol. 36, pp. 287-314, 1994
- [2] Cardoso, J.-F. *Super-Symmetric Decomposition of the Fourth-Order Cumulant Tensor. Blind Identification of More Sources than Sensors.* Proc. IEEE ICASSP-91, Toronto, pp. 3109-3112, 1991
- [3] Cardoso, J.-F. *Source Separation Using Higher Order Moments* Proc. IEEE ICASSP, vol. 4, pp. 2109-2112, 1989
- [4] Vittoz, E. A.; Arreguit, X. *CMOS Integration of Herault-Jutten Cells for Separation of Sources* Analog VLSI Implementation of Neural Systems (ed. C. Mead, M. Ismail), Kluwer Academic Press, 1989
- [5] Cohen, M. H.; Pouliquen, P. O. and Andreou, A. G. *Silicon Implementation of an Auto-Adaptive Network for Real-Time Separation of Independent Signals* Proc. IEEE ISCAS-91, Singapore, pp. 2971-2974, 1991
- [6] Herault, J.; Jutten, C. *Space or Time Adaptive Signal Processing by Neural Network Models* AIP Conf. Proc., Snowbird, UT 1986, pp. 206-211
- [7] Jutten, C.; Herault, J. *Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture* Signal Processing, vol. 24, pp. 1-10, 1991
- [8] Riemschneider, K.-R. *Parallele Hardware für Backpropagation-Netze auf der Basis stochastischer Rechenwerke* Diss., Univ. der Bundeswehr Hamb., 1996, <http://www.UniBw-Hamburg.de/EWEB/TI/veroeffentlichungen.html>
- [9] Köllmann, K.; Riemschneider, K.-R.; Zeidler, H. Ch. *On-Chip Backpropagation Training Using Parallel Stochastic Bit Streams* IEEE Proc. of MicroNeuro, Lausanne, pp. 149-156, 1996
- [10] Massen, R. *Stochastische Rechentechnik - Eine Einführung in die Informationsverarbeitung mit zufälligen Pulsfolgen* Carl Hanser, 1977
- [11] Cichocki, A.; Unbehauen, R. *Neural Networks for Optimization and Signal Processing* Wiley, ISBN 0-471-93010-5 (Wiley), 1993