Extended Abstracts of the 1994 International Conference on Solid State Devices and Materials, Yokohama, 1994, pp. 349-351

KOHSIP - A Dedicated VLSI-Processor for Kohonen's Self-Organizing Map

Andreas KÖNIG, Jochen REIMERS*, and Manfred GLESNER
Institute for Microelectronic Systems
Darmstadt University of Technology
Karlstrasse 15, 64287 Darmstadt,Germany

Our work targets on the implementation of a versatile VLSI-Implementation of Kohonen's self-organizing feature map that shall serve as an accelerator or neural coprocessor to a standard host computer. The processor consists of a full-custom ASIC with four parallel processing units and a FPGA control unit. The complete prototype system is integrated on a PC-board. Along with a set of evaluation and visualization tools the Kohonen processor will serve as fast and flexible device for applications as image coding or data analysis.

1. INTRODUCTION

Kohonen's self-organizing feature map (SOFM)[1] is an unsupervised neural network paradigm, devised according to biological evidence observed in the brain cortex, which offers interesting solutions for many applications, e.g. for exploratory data analysis and knowledge extraction, visualization, process control, image coding, and classification. In the majority of applications SOFM is applied with a two-dimensional arrangement of neurons in the processing layer. Finding the neuron most

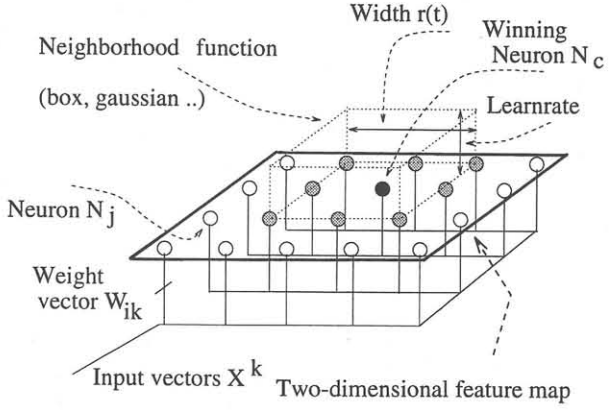

Fig.1 Two-dimensional feature map of size 5×4.

similar to the input stimulus and adapting the corresponding weight vector of this neuron and those of a time dependent neighborhood are the elementary computation tasks for the SOFM-algorithm. Using Kohonen's short-cut algorithm the nearest neighbor neuron N_c to stimulus X^g is determined by:

$$N_c = min_{i=0}^{m}(\sum_{k=0}^{l}(x_k^g - w_{ik})^2) \qquad (1)$$

In a following learning step adaptation takes place for all neurons within a neighborhood N_c according to:

$$w_{jk}^{new} = w_{jk}^{old} + \alpha(t,r)(x_k^g - w_{jk}^{old}) \qquad (2)$$

Input patterns are repeatedly presented while learnrate $\alpha(t,r)$ and neighborhood width $r(t)$ decrease with t. The neighborhood function, that determines the number of neurons to be adapted in the *ordering phase*, can be modelled using for instance a Gaussian function, a linear function, or a box function. In the *convergence phase* $r(t)$ was diminished to zero and only winner neuron N_c will be adapted. The time dependent decay function for learnrate and neighborhood can also be chosen from a variety of functions. The choice of these functions comprises a trade-off between obtainable quality of results and allowable simplifications for implementation.

SOFM carries out a vector quantization of the sample set along with a topology preserving mapping from the l-dimensional sample vector space to the two-dimensional map. The algorithm can be exploited by implementation on conventional general-purpose hardware but the inherent massive parallelism of the algorithm is thus wasted. Especially the training phase of the SOFM

*) Jochen Reimers is with Deutsche Bundespost Telekom, Forschungs- und Technologiezentrum (FTZ), 64295 Darmstadt, Germany.

consumes considerable computation time. Codebook generation in image coding may require the training and adaptation of a map with 32 × 32 neurons[2]. Also realistic examples of exploratory data analyses and visualization require for instance map sizes of 64 × 64 or 128 × 128 neurons[3]. Therefore, for exploratory data analysis as well as for real-time tasks, e.g., image coding, a dedicated SOFM hardware implementation is of significant interest.

2. KOHSIP-ARCHITECTURE

In this work a dedicated VLSI-processor was developed for the Kohonen algorithm and a prototype system with this ASIC was designed as a neural coprocessor or accelerator for a standard host computer (PC). A data path for the Kohonen algorithm was designed (s. Fig. 2) according

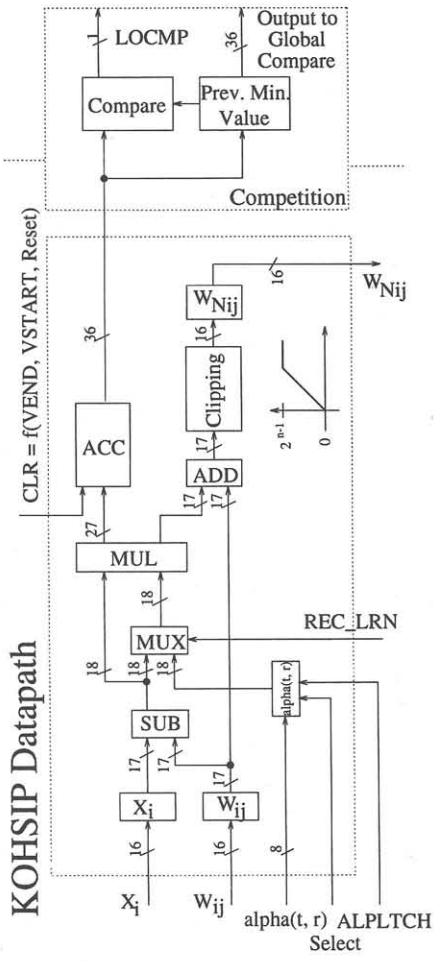

Fig.2 Block diagram of KOHSIP-data path.

to eq. 1 and eq. 2 using fixed-point arithmetic. Accuracy requirements were determined by extensive simulations in prior work[2]. The data path is devised as a SUMAC-structure (Subtract-Multiply-Accumulate), referring to the MAC-structure of traditional signal processors. For that reason we coined the processor KOHonen-SIgnal-Processor. Four data paths are integrated in one KOHSIP-ASIC (s. Fig. 3) exploiting the inherent paral-

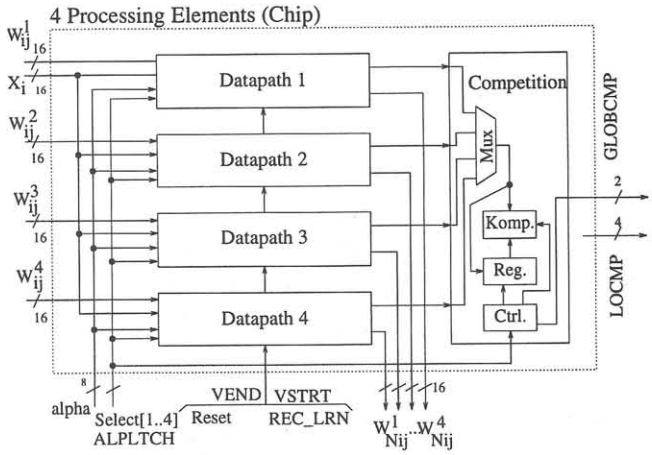

Fig.3 Block diagram of the KOHSIP-ASIC with four data paths.

lelism of the algorithm. The search process for the winner neuron N_c corresponding to a stimulus X^k takes place concurrently for four weight vectors. Each data path carries out a local comparison concurrently to distance computation determining the local winner neuron. Thus, only one final comparison for the global winner is required after searching the whole map. The chip provides parallel processing for different vectors but does serial processing of vector components. The ASIC can be cascaded for more demanding problems by adding an additional level of external comparison for, e.g. four chips. The KOHSIP-ASIC (s. Fig. 3) will be manufactured in 0.8μm technology using a full-custom approach with dedicated macro cells of the German Telekom. A single data path comprises ≈ 20000 transistors and the whole chip a total of ≈ 85.000 transistors consuming an area of $2.5mm^2$ per slice. The ASIC will be clocked with 70 MHz maximum, providing one ASIC with the power to train a SOFM of, e.g. 64 × 64 neurons for four-dimensional iris data with 15000 training steps in ≈ 3 seconds. The design of the ASIC is completed (s. Fig. 4) and manufacturing will take place fall 1994. The KOHSIP-system is concurrently designed and a prototype is implemented as PC-board based neural accelerator (s. Fig. 5 and Fig. 6). We expect the prototype to run at 20

MHz. The prototype was devised for flexibility allowing the simulation of feature maps of arbitrary sizes. A maximum of 512 vectors components and 2048×2048 neurons are supported providing a flexible allocation of available memory resources. The neighborhood and decay functions are stored in a LUT-memory of the controller. Thus arbitrary function forms can be used for the training process.

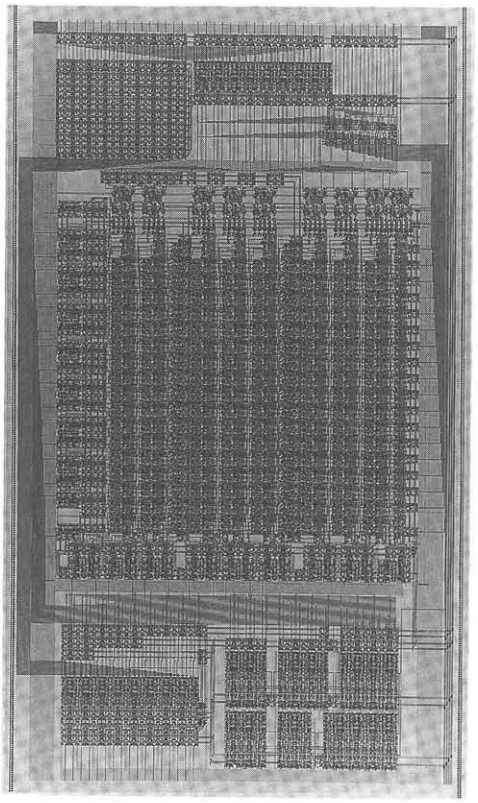

Fig.4 Layout of one KOHSIP data path.

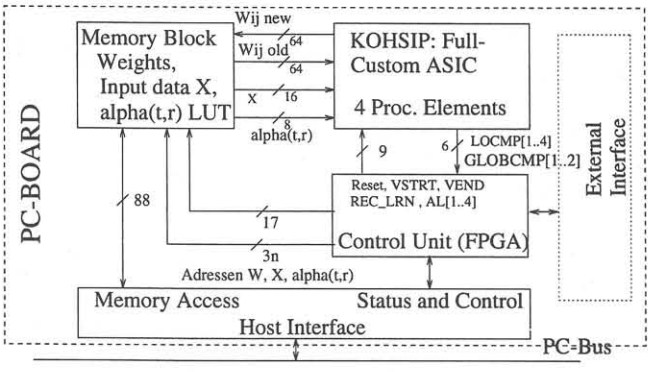

Fig.5 Block diagram of KOHSIP prototype system.

The long term research goal is the integration of the neural accelerator unit KOHSIP with a UNIX-PC under X-WINDOWS with a set of analysis and visualization tools developed for exploratory data analysis and general pattern recognition problems, e.g. finding structure in data or selecting significant features for classification problems.

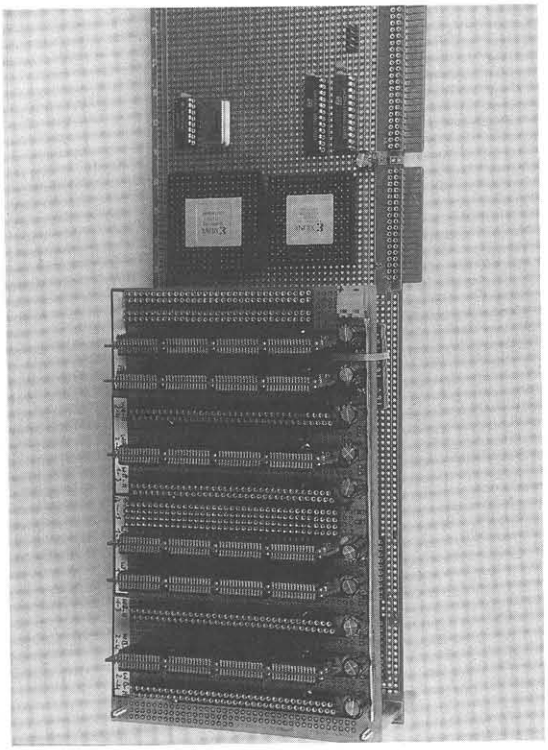

Fig.6 KOHSIP prototype system.

3. REFERENCES

1) T. Kohonen, *Self-Organization and Associative Memory.* Springer Verlag Berlin Heidelberg London Paris Tokyo Hong Kong, 1989.

2) A. König, X. Geng, and M. Glesner. Hardware Implementations of Kohonen's Feature Map by Scalar and SIMD-Array Processors. *Proc. Intern. Conference on Artificial Neural Networks ICANN'93*, Amsterdam, 1993.

3) A. Ultsch, H. P. Siemon. Kohonen's Self-Organizing Feature Maps for Exploratory Data Analysis, *Proc. Intern. Neural Networks*, Kluwer Academic Press, Paris, 1990