Extended Abstracts of the 1994 International Conference on Solid State Devices and Materials, Yokohama, 1994, pp. 394-396

ARAMYS-A Bit-Serial SIMD-Processor for Fast Parallel Nearest Neighbor Search and Associative Processing

Andreas KÖNIG, Peter WINDIRSCH, and Manfred GLESNER

Institute for Microelectronic Systems
Darmstadt University of Technology
Karlstrasse 15, 64283 Darmstadt,Germany

In this paper we report on the VLSI-implementation of a processor for competitive neural networks and its prototype implementation. The basic element of our SIMD-processor architecture are dedicated bit-serial processing elements that allow vector comparison using binary or integer metrics and support fast parallel minimum search. An ASIC with 32 processing elements was implemented. Systems with up to several thousand neurons can be implemented with our architecture for applications as image coding, classification and associative processing.

1. INTRODUCTION

Lateral inhibition in competitive neural algorithms is commonly formulated as a search for the minimum distance or maximum correlation between an input pattern vector and the corresponding neuron weight vector. In Kohonen's self-organizing feature map (SOFM)[1], for instance, competition between neurons is computed with Kohonen's short-cut algorithm, determining the nearest neigbour neuron N_j to stimulus X^g is by:

$$N_j = min_{i=0}^{m}(||X^g - W_i||) \qquad (1)$$

Here $||X^g - W_i||$ stands for an arbitrary vector norm, e.g., the Euclidean distance, the Manhattan distance, or for binary vectors the hamming distance. This computation of vector distance followed by minimum distance search is of interest for several important applications, e.g. vector quantization for image and speech coding, nearest neighbor classification in pattern recognition, neural networks and neural associative memories. Implementation by bit-parallel processors seems attractive as a fast solution but is hampered by two features of this approach. First, the limited carrier pin count is an obstacle for the integration of a significantly large number of processors on one chip. Second, the minimum search will be a bottle neck, consuming considerable time of the overall computation. Alternatively, processor architectures with a large number of bit-serial processing units can be considered.

2. ARAMYS-ARCHITECTURE

In our work we have developed an efficient bit-serial processor that offers a viable alternative to the bit-parallel approach. The basic element of the processor is a dedicated up/down counter that can be used for two binary metrics, Hamming distance (EXOR) and correlation (AND), and one integer metric, Manhattan distance (s. Fig. 1).

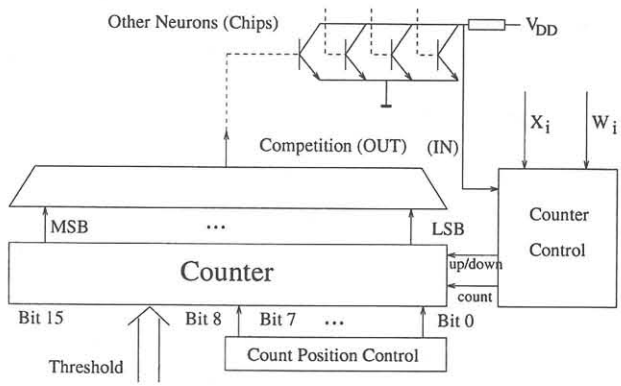

Fig.1 Processing element (neuron) with counter for metric computation. Manhattan distance is computed from MSB to LSB counting at corresponding counter stage 7 to 0 activated by global *Count Position Control*. The principle of the competition mechanism is sketched in the figure.

This counter cell computes in a single bit-serial step the vector component difference, the absolute of this difference and accumulates it to the overall distance.

Count actions are carried out observing the incoming bit pairs X^g_{ik} and W_{jik} as given in Table 1. The computation starts from MSB to LSB, observing the first pair with $X^g_i \neq W_{ji}$, which determines whether $X^g_i > W_{ji}$ or vice versa. For $X^g_i > W_{ji}$ column 3 is used subsequently for count actions, for $X^g_i < W_{ji}$ column 4 is used. Thus, in all cases $|X^g_i - W_{ji}|$ is computed and accumulated to $||X^g - W_j||$. No further registers are required.

X^g_{ik}	W_{jik}	Count action	Reversed action
1	1	No count	No count
1	0	Up count 2^k	Down count 2^k
0	1	Down count 2^k	Up count 2^k
0	0	No count	No count

Table.1 Count actions for observed inputs X^g_{ik} and W_{jik}.

The minimum search of eq. 1 is also carried out by a bit-serial procedure, but in parallel for all neurons. Each neuron is equipped with an additional flag indicating participation in competition. Initially all neurons are participating in competition. Starting from the most significant counter bit for all neurons in the SIMD-computer a wired-OR value of all bit values at the current bit position is computed. Every participating neuron compares its local bit value with this global bit value. In case of a global "0" and a local "1" the corresponding neuron will no more participate in competition, as there is at least one neuron with a smaller activation or distance value. Thus, after proceeding to the least significant counter bit, only those neurons having the minimum distance value are still in competition and marked active. Active neurons are identified by a priority encoder, which also gives a tie breaking rule for the order of selection if more than one neuron is active. By repetitive competition the k-nearest neighbors of an input vector X^g can be computed. A neuron is blocked from participating in competition when an overflow occurs during distance computation. This mechanism can be exploited by setting an initial counter value as global or local threshold, thus limiting the distance computation to well defined hyperspheres. This can serve to implement a rejection threshold for nearest neighbor classifiers or for the implementation of the RCE (Restricted-Coulomb-Energy) classifier[2]. Additionally, counter values can be read out and transferred to the system controller for further processing.

3. ARAMYS-II ASIC

We have implemented this architecture by a standard cell chip comprising 32 of these simple neurons. Weight memory is off-chip in standard RAM chips or modules. The chip is packaged in a 68 pin CLCC carrier and was manufactured with 1.0μm process, featuring an area consumption of $\approx 56\ mm^2$, 13783 cells, 33449 gates, and 133799 transistors (s. Fig. 2). The chip runs at 20 MHz

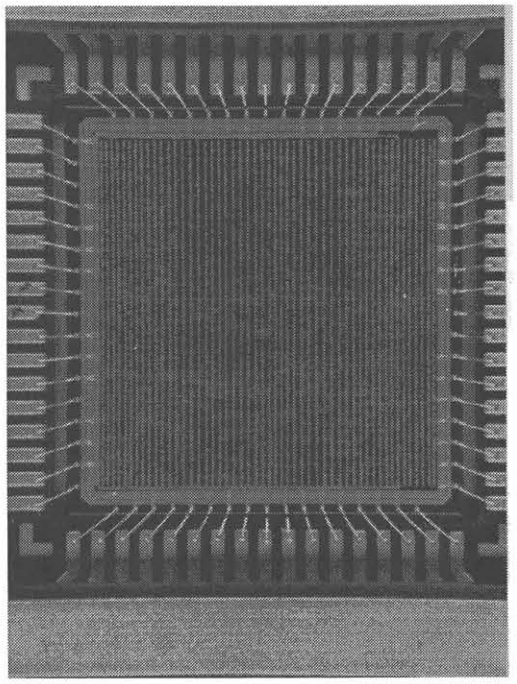

Fig.2 ARAMYS-II standard cell chip with 32 bit-serial processors.

(maximum 23 MHz) and can be cascaded to a SIMD-computer of an arbitrary size. Currently, four chips are integrated in a prototype system (s. Fig 3.) developed as a research tool and demonstrator for neural networks and neural associative memories, coined ARAMYS (Autonomous Real-time Associative MemorY System). Referring to the prototype system, the chip is denoted as ARAMYS-II. The chip was tested in the prototype system and current work focuses on the extension of the prototype system as a demonstrator, e.g. for vector quantisation in image coding, classification in pattern recognition, parallel template matching, and image inspection in automated visual industrial quality control. The ARAMYS system controller is based on a PC-board and the system works as a neural coprocessor for the PC host. Our architecture offers a basis for the efficient imple-

mentation of vector quantizers[3], k-nearest neighbor classifiers ($k \geq 1$)[4] [5], hypersphere

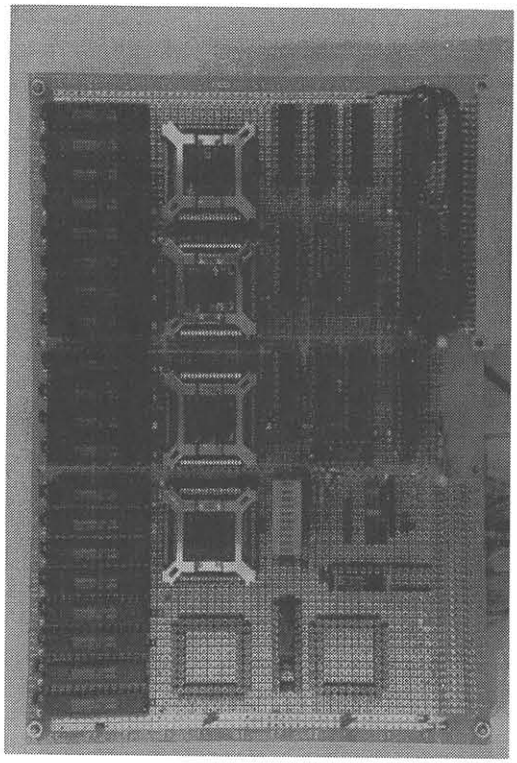

Fig.3 Module board of the ARAMYS-prototype system with four ARAMYS-II chips and the corresponding weight vector memory.

classifiers[2], and SOFMs[1]. Application domains for classification and pattern matching are, for instance, mechatronic and visual industrial quality control tasks. Real-time constraints can be met by adding chips and processing modules. For instance, processing speed can be doubled when two input vectors are processed in parallel doubling the number of processing elements and chips. For domains of significant real-time demands, e.g. image coding, an improved version of our architecture was designed as a full-custom implementation, denoted as ARAMYS-III. First simulations showed a speed of $\approx$ 80 MHz for a neuron cell, thus coming well in the range for real-time vector quantization of video phone images.
The architecture of the neuron allows an enhancement to implement a simplified version of SOFM with on-chip learning. The learning rule of Kohonen's SOFM algorithm can be simplified to:

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

A box function is assumed for the neighborhood function. The learnrate and the neighborhood width are globally computed and broadcasted along with the winner index to the individual neurons. Each neuron performs a local computation determining whether it is inside or outside the the box centered at winner N_c. This can be accomplished using the counter for computation. The multiplication in the learning rule is replaced by shift right operations ($\alpha(t,r(t)) < 1$). This also can efficiently be carried out by our processing element, as shift right can be accomplished by starting the count algorithm at a lower counter bit position, e.g. for a right shift of 3 bit positions we start the serial computation at position $2^{7-3} = 2^4$ instead of 2^7. Thus, with very little additional overhead our neuron can be extended to a SOFM neuron with on-chip learning. We currently model such a parallel SOFM-architecture using VHDL.

4. ACKNOWLEDGEMENT

The authors wish to express their gratitude to Mr. Geng-Xia for his work towards the ARAMYS-III full-custom chip implementation and to Mr. Martin Gumm for the implementation of the ARAMYS-II chip presented in this paper, which was fabricated with financial support by DFG, SFB 241 "IMES".

5. REFERENCES

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

2) C. Elbaum D. L. Reilly, L. N. Cooper. A Neural Model for Category Learning. Biological Cybernetics 45(1982) 35.

3) A. König and M. Glesner, An Approach to the Application of Dedicated Neural Network Hardware for Real Time Image Compression. Proc. Intern. Conf. Artificial Neural Networks ICANN'91 2(1991) 1345.

4) T. M. Cover and P. E. Hart, Nearest Neighbour Pattern Classification. In *IEEE Trans. on Information Theory* 13(1967) 21.

5) P. E. Hart, The Condensed Nearest Neighbour Rule. In *IEEE Trans. on Information Theory* 14(1968) 515.