Proc. of the ProRISC/IEEE Benelux workshop on Circuits, Systems and Signal Processing, Mierlo(NL), March 23-24, 1995

An associative processor architecture for pattern regognition

Philippe Thissen

Michel Verleysen

Jean-Didier Legat

Université catholique de Louvain,
Microelectronics Laboratory - DICE,
3 Place du Levant,
1348 Louvain-La-Neuve, Belgium
tel: +32 (10) 472551, fax: +32 (10) 478667

email:thissen@dice.ucl.ac.be

Abstract

This paper describes a parallel SIMD architecture dedicated to classification and pattern recognition applications. The SIMD architecture is built around a bit-serial word parallel associative processor and contains many elementary processors (EPs) working in parallel. This SIMD architecture has been designed in order to speed up the main operations involved in standard classification algorithms (distance, extremum computations, ...); some specialized hardware resources have been added in this purpose too.

The parallel architecture described in this paper is the data path of a more complex autonomous system dedicated to classification. It is programmable and many algorithms can be implemented only by writing their associated microprogram. The computational power of the architecture is fully exploited by neural-like algorithms for classification where many independent parallel computations must be realized and only a few global operations involving the output of all EPs are needed.

Such a processor, operating at a frequency of 100 MHz and built with 1024 EPs, is able to process up to 10 billions of 8 bits additions per second and to classify 400 000 vectors in 2 dimensions with neural network algorithms like LVQ or RCE.

This VLSI associative processor is being implemented in 1 μm technology with full custom design.

1 Introduction

In this paper, we describe a massively parallel processor adapted to classification by neural-like algorithms. The processor is based on a SIMD bit-serial, word parallel associative architecture and represents the data path of a general classification machine. This associative processor consists of many simple processing units working in parallel. Each unit is able to realize simple operations like bit-serial additions or comparisons. Some specialized cells, taking as inputs the outputs of all elementary processing units, have been added to execute collective computations needed in the algorithms; these dedicated ressources have been chosen in order to speed up classification tasks. We insist on the fact that this associative architecture is multi-purpose and many different algorithms can be runned on this machine only by writing the related microprogram; of course, the architecture and the basic simple and collective operations are particular adapted to classification algorithms.

This operative part of the processor is currently under design in the 1 μm ES2 technology. The elementary ALU and the static memory cell have been successfully simulated at frequencies up to 100 MHz. The size of the different blocks of this operative part allows to build a parallel processor formed by 128 elementary processing units, each unit provided with 256 bits of memory, in only 6×6 mm. The parallel processor is expandable and larger networks can be realized by connecting several circuits together.

In this paper, we first briefly present the functional description of the processor; the different parts of the elementary processing unit and the operations realized by these parts are described. Results are given about the performances of this architecture for simple mathematic operations. The second part of this paper deals with the implementation of neural classification algorithms; as an example, we focus our attention on the RCE algorithm, and present its learning and classification phases on our architecture.

2 Functional description of the architecture

The associative architecture described in this paper is built around a SIMD bit-serial, word parallel associative structure taking benefit of its high computational power and relatively low hardware cost. Figure 1 presents the whole structure of this processor. It is composed of many elementary processing units working in parallel; the memory addresses and the data to be processed are identical for all processing units and come from outside the circuit.

The elementary processing unit, illustrated at figure 2, is composed of 3 parts : a memory, a 1-bit ALU and a status register.

- The memory is a static random access memory with two busses: one for reading information and another for writing in the cell. This double access allows to execute simultaneously reading and writing operations in two different cells during the same clock cycle. This useful property is fully exploited as explained in a next section.
- The 1-bit ALU (EP) executes classical operations such as additions, subtractions and comparisons. Specialized hardware has been added to speed up comparison operations and extremum searches. The ALU processes data depending on the value of its status register.
- The status register (S) is a one bit register. It memorizes the state (active or not) of the related processing unit. Of course, the result of a mathematical or logical operation can

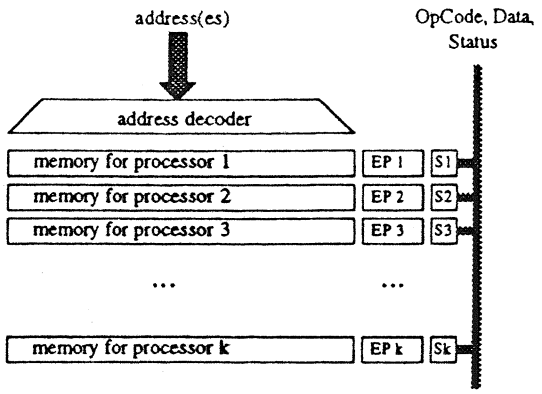

Figure 1: Architecture of a bit-serial, word-parallel associative processor.

be put into this status register; the status information can also be stored in memory or retrieved from it at any address.

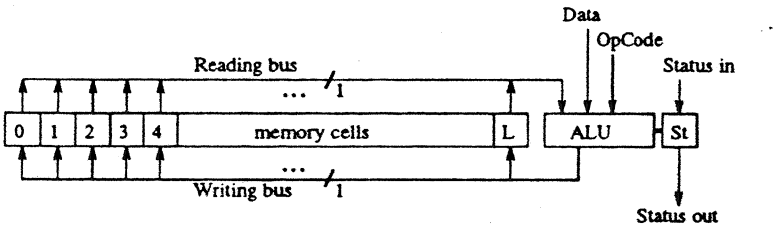

Figure 2: Schematic representation of the elementary processing unit.

Three different operations can be realized at the same time in one clock period on different data. These operations are:

- reading one data bit in the memory and loading it into the input buffer of the ALU;
- arithmetical or logical operation on data contained in the input buffer of the ALU, the result being stored in the output buffer;
- writing the value contained in the output buffer into the memory.

This pipelining virtually multiplies the clock frequency by 3 and increases the number of operations up to $3\,10^8$ per second for a clock frequency of 100 MHz.

Each processing unit is able to carry out simple computations, reading data from its associated memory and storing the results in it. Dedicated ressources have been added to this structure to execute collective functions operating on all elementary processing units together. Two functions are particularly recommended for classification tasks: they are *electing the first active EP* and *summing all their outputs.*

- The first function is very useful for extremum searches or to read sequentially into the memory of each processing unit. This combinatory block sends a token which goes through the status register of all processing units. To avoid large delays due to very long chains of EPs, some tricky ressources have been designed to bypass the normal way of the token.

- The second function sums the outputs of all active processing units. It is built around a full-adder tree and the data go through this tree in a pipelined way. This function is used to count the number of active processing units and also to realize sums.

This architecture is multi-purpose and expandable: many different algorithms can be runned only by writing their associated microprogram, and several chips can be connected together to increase the number of EPs.

Each processing unit is able to compute bit serial additions without any interaction with other processors and, by extension, subtraction, multiplication and division [3]. Floating point operations can also be executed on this processor. Table 1 gives the number of operations that can be processed on this architecture with 1024 elementary processing units clocked at 100 MHz; the results are given in million of operations (addition, subtraction, multiplication, division) per second (MOPS). For the floating point operations, the 32 bit IEEE format was chosen.

	integer			*real*
	8 bits	*16 bits*	*32 bits*	*32 bits*
addition				
$X + A = C$	10240 MOPS	5687 MOPS	3012 MOPS	80 MOPS
$B + A = C$	5687 MOPS	3012 MOPS	1552 MOPS	80 MOPS
subtraction				
$X - A = C$	9310 MOPS	5390 MOPS	2925 MOPS	80 MOPS
$B - A = C$	5390 MOPS	2925 MOPS	1527 MOPS	80 MOPS
multiplication				
$X \times A = C$	397 MOPS	122 MOPS	35 MOPS	57 MOPS
$B \times A = C$	317 MOPS	92 MOPS	25 MOPS	42 MOPS
division				
$X/A = C$	132 MOPS	40 MOPS	12 MOPS	20 MOPS
$B/A = C$	112 MOPS	35 MOPS	10 MOPS	17 MOPS

Table 1: Computational power of the massively parallel processor formed by 1024 elementary processing units clocked at 100 MHz; one MOPS is equivalent to one million of operations per second.

The number of comparisons that can be carried out per unit time is also very impressive; the complexity of comparisons is the same as for subtractions.

3 Implementation of RCE algorithm on this architecture

Neural-like classification algorithms are particularly adapted to this architecture. In fact, in such algorithms, a lot of computations (for example, distance computations) can be done in parallel without any interaction between the different processing elements; this kind of algorithm is thus perfectly adapted to the proposed architecture. To illustrate this principle, we propose to study the implementation of a particular neural classification method on the processor: the well-known RCE algorithm. It has however to be kept in mind that this algorithm has only been chosen as an example, and that the architecture is also well adapted to other algorithms like LVQ, Bayesian classifiers, ...

3.1 The RCE algorithm

The RCE algorithm [2] is an incremental neural network belonging to the class of Region Of Influence (ROI) algorithms.

The network contains several decision units (prototype neurons) defining hyperspheric regions of influence in the input space. Each decision unit is first characterized by the coordinates of the center of the region of influence in the input space, and secondly by its radius. Any input falling within these regions activates the neuron(s) attached to the region(s); the neurons are linked to a class.

During the learning process, if new classes are encountered, new decision units (neurons) are created. If misclassification occurs, new exemplar units are formed and/or weights are modified.

In some real time applications, hardware accelerators may be necessary for RCE or similar algorithms; parallel implementations are particularly adapted [1].

3.2 The learning phase

The learning principle consists in successively presenting the patterns of the learning set and in verifying if they belong or not to one or several ROI already defined (if their distance to the prototype neurons are less than the radius of the associated hyperspheres).

Each neuron of the network is associated to one elementary processing unit. The different steps of the learning phase are:

1. **distance computations:** Each elementary processing unit computes the Manhattan distance between the input vector and the coordinates of the center of the ROI; these coordinates are of course only loaded once in the associated memory of the processing unit. All distances can be computed in parallel without interaction between elementary processing units.
2. **activation of neurons:** We compare the calculated distance with the radius of the influence region, also loaded in the memory. The only active neurons are the one whose influence region radius is greater than the calculated distance. This operation can also be executed in parallel on all processing units.
3. **modification of existing radii and/or creation of new neurons:** We must modify the influence region radius of neurons leading to misclassification, that is activated neurons who have a different class that the input vector; the new radius is fixed to the calculated distance between the input vector and the stored prototype. Secondly, we must control that the input vector is well classified by, at least, one neuron; if not, a new neuron is created, i.e. an unused processing unit is allocated and memorizes the coordinates of the input vectors and an a priori radius.

The left part of figure 3 illustrates the number of vectors that can be learned per second with the RCE algorithm; these results represent a worst case study.

Simulations of our architecture on the RCE algorithm have been carried out on a bidimensional database containing 1000 elements divided into two separated classes (a circle and a external ring). The data accuracy has been truncated from 5 to 32 bits and the number of vectors learned per second is about 25 percent higher than the results presented in figure 3.

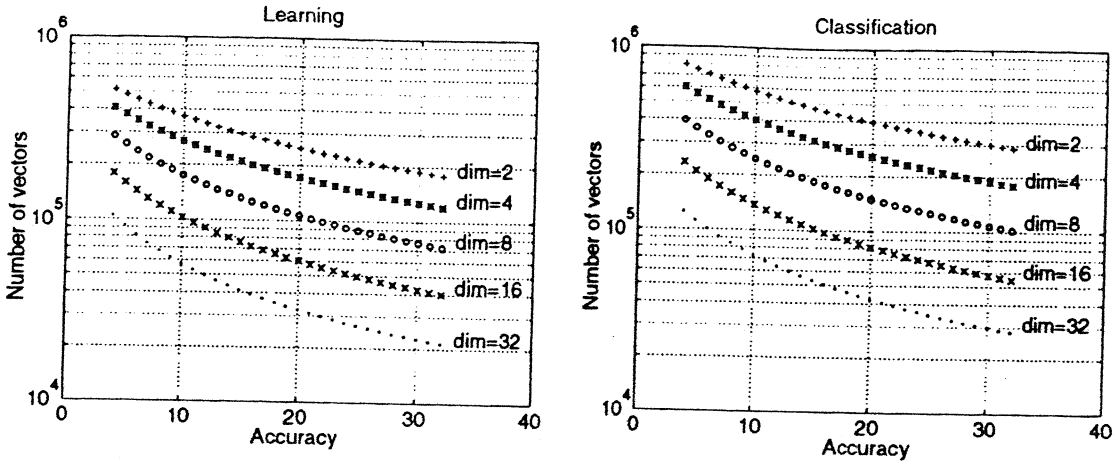

Figure 3: Number of vectors learned (left part) and classified (right part) by the associative processor clock at 100 MHz with the RCE algorithm.

3.3 The generalization phase

During the generalization or classification phase, an unclassified vector is presented to the network. The following operations occur:

1. to compute distances between the input vector and all stored centroids;
2. to set active the neurons which influence region includes the input pattern (the calculated distance is smaller than the stored radius of the ROI);
3. to fix the class of the input vector to the class of one randomly chosen active neuron and to verify than there is only one class proposed.

The two first operations, which are the most time consuming, are the same than in the learning phase. The right part of figure 3 illustrates the number of vectors that can be classified per second with the RCE algorithm. Of course, no modification of radii or creation of new neurons occurring in the classification phase, the number of vectors that can be classified is thus greater than the number of vectors that can be learned in the same conditions (space dimension and accuracy of data).

4 Conclusion

This paper describes a parallel digital architecture dedicated to classification tasks. A SIMD processor is built around a bit-serial word-parallel associative architecture; it is programmable and many algorithms can be implemented only by writing their associated microprogram. The computational power of the architecture is fully exploited by neural-like algorithms for classification. A full custom VLSI implementation of this processor is being developed.

5 Acknowledgments

Philippe Thissen is working towards the Ph.D. degree in microelectronics under an IRSIA (Institut pour l'Encouragement de la Recherche Scientifique dans l'Industrie et l'Agriculture) fellowship. Michel Verleysen is a Senior Research Assistant of Belgian National Fund for Scientific Research (FNRS).

References

[1] Intel Neural Network Group. Design and implementation of a recognition accelerator. In *Proceedings of the Canadian Conference on Very Large Scale Integration*, 1993.

[2] D.L. Reilly, L.N. Cooper, and C. Elbaum. A neural model for category learning. *Biological Cybernetics*, 45(1):35–41, 1982.

[3] I. Scherson, D. Kramer, and B. Alleyne. Bit-parallel arithmetic in massively-parallel associative processor. *IEEE Transactions on Computers*, 41(10):1201–1210, 1992.