On-Line Arithmetic Based Reprogrammable Hardware Implementation of LVQ Neural Network for Alertness Classification

Mohamed BOUBAKER[1], Khaled BEN KHALIFA[1], Bernard GIRAU[3], Mohamed DOGUI[2] and Mohamed Hédi BEDOUI[1].

[1]TIM Project; Biophysics Laboratory, Medical Faculty of Monastir
Rue Ibn Sina - Monastir - 5000, TUNISIA
[2]NVAP Project; Physiology Laboratory, Medical Faculty of Monastir
Rue Ibn Sina - Monastir - 5000, TUNISIA
[3]Cortex Project ; INRIA Lorraine Research Groups, LORIA, Technopole de Nancy-Brabois -Campus Scientifique 615, rue du Jardin Botanique - BP 101- 54602 Villers-les-Nancy Cedex-FRANCE

Summary
The current study presents the hard implementation of a learning Vector Quantization (LVQ) neural network. Starting from the spectral EEG analysis, we suggest an LVQ serial on-line architecture implementation on a Field programmable Gate Array (FPGA) circuit. Our concern was mainly to get a light, easy-to-wear system for the classification of vigilance levels in humans using EEG signals. The results of these classified states by LVQ mode are presented in this paper. Furthermore, the highly-satisfactory performances of our implementation in terms of area speed and delay are described.

Keywords: LVQ neural network, on-line arithmetics, FPGA, EEG and vigilance.

1. Introduction

Our study is aimed at developing an embedded system able to detect hypovigilance from a single EEG derivation. The developed system is to be used in an ambulatory manner with real-time computation. The approach proposed in this paper has two parts. The first part is focused on the classification of the states of the awakening-sleep transition by using the topological properties of self-organizing maps. This connectionist unsupervised approach will be summarized in this paper and is fully described in [1]. The second part deals with the implementation of an LVQ neural network on an FPGA programmable circuit for a future exploitation in an embedded system.

As for the first part, many investigations have led to the development of neural network-based systems to assess vigilance levels using electroencephalogram (EEG) signals [2, 3, 4 and 5]. All these systems ensure the vigilance quantification more or less successfully with a monitoring analysis capability. In [2], T.P Jung and colleagues suggested a method based on the spectral component analysis and a multilayer neural network, where 10 subjects were studied over epochs of 1.29 s. The aim was to study on the one hand the correlation between the EEG signal spectrum and the vigilance level quantified by an auditory test, and on the other hand, the automatic classification of vigilance states from the spectrum of the same EEG signal performed by a neural network.

In [4], a Radial Basis Function (RBF) neural network made it possible to classify the vigilance levels in 12 healthy subjects from five-second EEG portions. The considered parameters are the coefficients of an autoregressive model (AR). Kohonen Self-organizing maps were used to make the cartography of the awakening-sleep transition over EEG epochs of 1.28 s while taking into account the artifacts [5].

In a study considered as the most recent and complete one, Vuckovic [3] exploited three different models of neural networks; a feed-forward multilayer network using a supervised training with the Levenberg-Marquardt (LM) algorithm, a linear neural network with supervised training using the Widrow-Hoff rule and an LVQ. These three algorithms were used to separate only two physiological states (alertness and drowsiness) from one-second EEG epochs recorded with 14 electrodes in 12 subjects. This classification did not take into account the artefacts which had been previously eliminated referring to two experts' opinions. It should be noted that the EEG recordings were carried out in the particular conditions of darkness while forcing the subjects to close their eyes. In this paper, we used an LVQ neural network to classify a state of vigilance.

In the second part, various hardware implementations of Kohonen neural networks on integrated circuits have been presented. They may be divided into two main categories:

- Analog implementations of the Kohonen neural networks on dedicated integrated circuits have been designed (for example [6]). These supports are technically limited as they lack precision. Their performance greatly depends on the technology used.
- Implementations of Kohonen neural network on ASIC digital circuits (neuroprocessors) have also

*Corresponding author : **Khaled BEN KHALIFA,** TIM Project, Biophysics Laboratory, Medical Faculty of Monastir, Rue Ibn Sina - Monastir - 5000, TUNISIA Tel +216 73 46 22 00; Fax +216 73 46 07 37*

been designed [7]. This is now the most used category of VLSI for the integration of neuromimetic algorithms. However, these devices are costly and lack flexibility.

The above shortcomings of both types of implementation devices may be avoided thanks to reprogrammable circuits, such as Field Programmable Gate Arrays (FPGA). These circuits offer high-performance, high-speed and high capacity programmable logic solutions that enhance design flexibility while reducing time-to-market. Configurable hardware appears well-adapted to obtain efficient and flexible neural network implementations.

Several special-purpose hardware implementation of LVQ neural network on FPGA have been proposed. In [8] the authors present the implementation of an LVQ with four 16-bit inputs and 9 neurons on the output layer on an FPGA Altera EP20K1500EBC652 for the recognition of odors. For distance computation, the L1 norm was adopted and Learning was done off chip.

In [9] the authors propose the implementation of a 64-neuron LVQ with 16 inputs on ACEX1K100. The used arithmetic was parallel. Each input was represented with 16 bits. The distance was computed with the L1 norm and learning was done off chip.

In [10] the authors describe the implementation of the SOM and LVQ algorithms with 23 inputs of 8 bits each and 25 neurons on the output layer on an FPGA Xilinx XCV1000BG for the classification of vigilance. For computing the distance the author adopted the L2 norm and learning was done off chip.

In this context, and in order to ensure an embedded implantation of a real time LVQ neural network, we adopted a serial approach to reduce the area of implantation. The parallelism of artificial neuronal networks and particularly the LVQ, as well as the great number of inputs and outputs, justified the choice of the serial approach.

In this paper, the implementation of an authentic LVQ architecture to classify human alertness is proposed. FPGA circuits have been chosen for the implementation taking advantage of their rapid prototyping at a low cost.

This paper is organized as follows: Section 2, presents our experimental setup and its major issues: subjects, recordings and spectral processing of EEG, Vigilance level classification by LVQ neural network and LVQ implementation on FPGA. Section 3, describes the results of our implementation: speed, area and delay. Therefore, we implement an on-line serial arithmetic with the MSBF mode (Most Significant Bit First) using a redundant number representation system. Finally, we conclude in section 4.

2. Materials and methods

2.1 Subjects:

This study was concerned with a control group of four healthy male medical students, aged 18 to 23. The recruitment was made by direct contact and voluntary membership. Each subject had three 24-hour recordings fortnightly with a15-day interval.

Recordings: The equipment in use is an ambulatory long-duration recording system with 8 channels, OXFORD MEDILOG 9000 model. The analogical recording was made on a magnetic tape. The analogical recordings were digitized and visualized by a second reading system.

Each recording contains two EOG channels, an EMG channel of the chin and five EEG channels. For the EOG, the active electrodes are placed at the level of the external canthus (on the right and on the left) with the reference at the level of the contra lateral mastoid. The EMG is recorded by a bipolar diversion connected to two 2 cm distant electrodes placed on the cowlick and the chin. The EEG is recorded by bipolar diversions (F3-F4; C3-P3; C3-01; C4-P4 and P4-O2).

All the registered signals were sampled at 128 Hz. Four noisy recordings are eliminated. A 24-hour recording, for every subject, is selected (four 4-hour recordings are used in our application). These recorded signals are exploited by an expert in EEG and polysomnography interpretation to label two vigilance levels (sleep and awakening).

2.2 Pre-processing:

In our approach, we wish emphasize on a realistic design, including the hardware implementation as discussed below. In order to allow a portable, easy-to-wear system, we have tried to find a compromise between as few electrodes as possible and acceptable performances, which is an important drawback with regard to existing approaches. More precisely, we have opted for only a right parieto-occipital EEG derivation (P4-O2). The choice of the derivation P4-O2 helps to avoid the ocular frontal derivation artefacts and allows to get an alpha activity of a posterior topology, a characteristic of the calm awakening with closed eyes.

The spectral pre-processing applied on this derivation (P4-O2) consists of a Short Term Fast Fourier transformation STFFT with 4-second portions and a 512-point Hamming window ponderation type. For this purpose, 23 bands of 1 Hz, normalized from 1 to 23Hz, are used:

$$PSP_i = \frac{PS(i \quad to \quad (i+1)Hz)}{TPS} \times 100$$

$$i \quad from \quad 1 \quad to \quad 23 \ Hz$$

PSP_i = Percentage of the power spectrum of the corresponding i band.
TPS = Total power Spectrum.
PS_i = Power Spectrum of the corresponding i band.

Our corpus is presented in table 1:

Table 1: The corpus structure

Epochs				PSP				Expert label
Epoch 0	PSP0 1	PSP0 2	PSP0 3	.....	PSP0 21	PSP0 22	PSP0 23	Sleep
.	.	.	.	.....	.	.	.	.
.	.	.	.	.....	.	.	.	.
Epoch j	PSPj 1	PSPj 2	PSPj 3	.....	PSPj 21	PSPj 22	PSPj 23	awakening
.	.	.	.	.....	.	.	.	.
.	.	.	.	.....	.	.	.	.
Epoch N	PSPN 1	PSPN 2	PSPN 3	.....	PSPN 21	PSPN 22	PSPN 23	awakening

After this treatment and the choice of a signal band coding, connectionist treatments are applied, as described below.

2.3 Vigilance level classification:

In this work, an LVQ connectionist model was used for the vigilance state classification. This model of neural networks was created by Kohonen [12], and it is a method of training neural networks for pattern classification. Figure 1 illustrates this network's architecture, which includes:

- The **input** (called input layer) represents the space of inputs X of dimension NR,
- The **competition layer** models the space of the inputs,
- The **linear layer of decision** allows to make decisions.

Each class is referred to a vector of weights that, in turn, represents the center of the clusters defining the decision hypersurfaces of the classes.

The linear layer of decision makes decisions thanks to the weights $w_{ci,j}$ of the matrix Wc in the following way: $w_{ci,j} = 1$ if the neuron J is associated class I, else $w_{ci,j} = 0$.

The learning of LVQ network consists in best positioning the prototypes by employing Kohonen rule [12].

Algorithm 1 describing the four main steps of the LVQ neural networks is given below:

Algorithm 1: LVQ (Learning Vector Quantization)

1 Select the nearest prototype to the input vector X

$$C = \text{argmin } \{|| X - wi ||\}, \text{ for } 1 \leq I \leq NR$$

2 Modify the WC vectors

$$WC(t+1) = WC(T) + \alpha(T) * [X(T) - WC(T)]$$

if X and WC belong to the same class

3 Modify the WC vectors

$$WC(t+1) = WC(T) - \alpha(T) * [X(T) - WC(T)]$$

if W and WC do not belong to the same class

4 $wi(T) = wi(T)$ for $I \neq C$

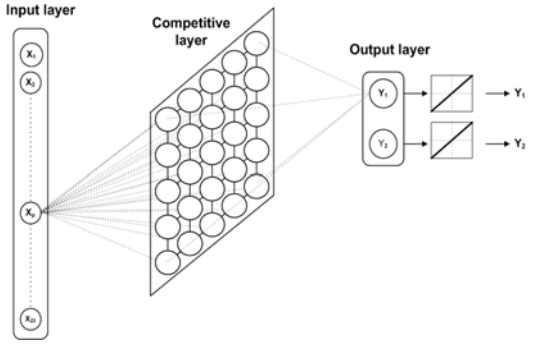

Figure 1: Structure under layers of the LVQ neural network

2.4 Implementation performance criteria:

In order to validate the concepts described above, we implemented the decision process of the LVQ neural network on FPGAs (using pre-trained parameters obtained from software simulation). Such a work has to take into account several parameters and useful characteristics such as power consumption according to the external clock frequency rate, the number of inputs/outputs, integration surface, or neural parallelism (the various neurons may work concurrently). Some of these parameters are difficult to estimate before synthesis.

Computation speed is sometimes another essential criterium. In our case, the decision (hypovigilance detection) is done in real time without difficulty thanks to a completely parallel implementation that does not require a sequential use of the resources or dynamic reconfigurations of the FPGA. Therefore, speed does not stand as a real constraint for the technological choices that must be made. On the contrary, the assumption of an embedded circuit in an ambulatory system requires a very low-power implementation. Among the parameters quoted above, the number of inputs/outputs, and above all the level of neural parallelism have a direct influence on the obtained implementation consumption. A fully parallel implementation is a real challenge, taking into account the

size of the neural networks and the FPGA.

2.5 Implementation environment**:**

Xilinx FPGAs [11] have been chosen because they offer a high memory/logic ratio: they are well-suited for the implementation of serial and on-line arithmetic operators, and their computation grain fits well the parallelism of neural computations. The configurable logic blocks (CLBs) can be connected using a configurable routing structure. In Virtex FPGA, CLBs can be efficiently connected to the neighboring CLBs as well as the CLBs of the same row or column. The configurable communication structure can connect external CLBs to input/output blocks (IOBs) that drive the input/output pads of the chip.

Our implementations are based on a Celoxica RC1000-PP board with a Xilinx Virtex XCV1000E-BG560 FPGA. Each CLB in the Virtex-E FPGA family corresponds to 4 configurable logic cells. It must be pointed out that current FPGAs already out perform the capacity of the Virtex XCV1000E: such an FPGA contains 27648logic cells, to be compared with the 73008 ones of the largest Virtex-E, and with the 125136 logic cells of the largest current Virtex-II Pro.

The board includes 8MB of SRAM directly connected to the FPGA in four 32-bitwide memory banks. This memory may also be addressed from the host CPU across the PCI bus so as to share data between the host CPU and the FPGA. The RC1000-PP board suggests a programmable clock frequency between 400 KHz and 100 MHz.

2.6 Technological choices:

Because of the great number of data involved in a neural network application (particularly LVQ neural network), the number of wires in hardware implementation might represent a critical problem. This problem was overcome opting for serial communications. Which take advantage of the development of various serial arithmetics.

In terms of implementation area and delay, the bit-level pipeline parallelism and the neuron-level parallelism (23input neurons and 5x5 output neurons simultaneously handled) lead to better performances with a serial arithmetics than with solutions that are based on parallel arithmetics.,.

In a preliminary work [10], we have described the implementation of a Kohonen neural network on a Virtex-type FPGA, using an LSBF (Least Significant Bit First) serial arithmetic: it mostly uses arithmetical operations that may be performed in an LSBF way (subtraction, addition, squaring) except for comparisons, for which a buffer reverses digit streams from LSBF to MSBF. This work has led to very satisfactory performances (the whole LVQ being implemented in parallel on the Virtex XCV1000E FPGA). In order to adapt our solution to smaller devices, we also needed to reach smaller implementation areas. That is why we have developed another LVQ, based on an L1 norm so as to get rid of quadratic operators. Nevertheless, numerous MSBF operators are added: absolute value computations. Therefore, in this paper, we handle only radix-2 MSBF serial arithmetics: the on-line arithmetics [13].

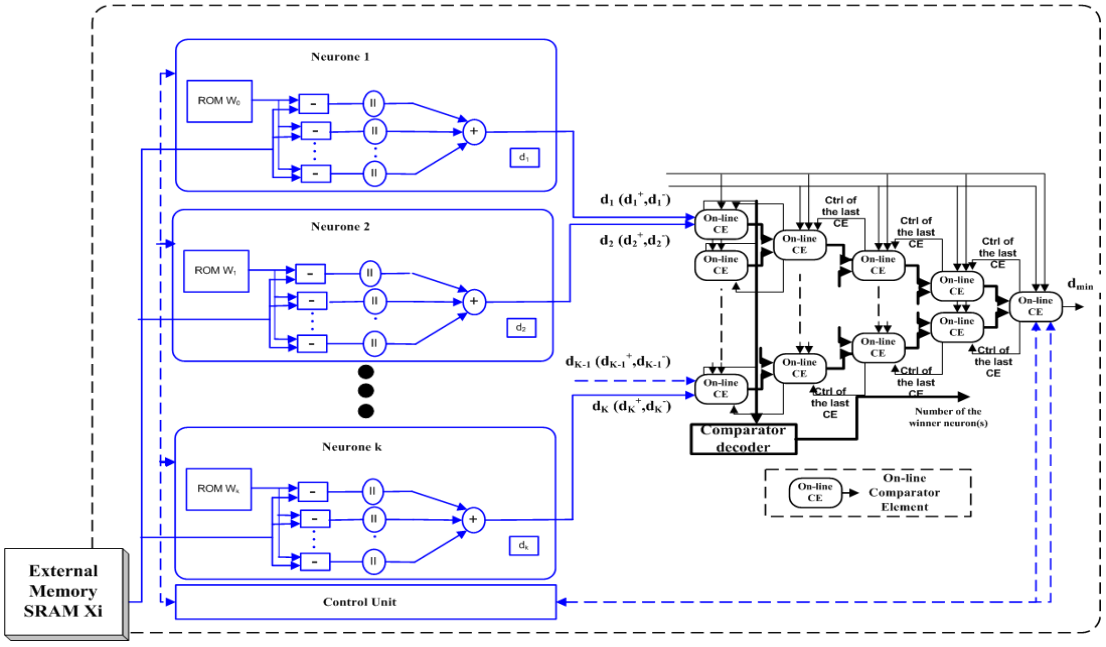

Figure 2: Data path architecture of the LVQ neural network.

2.7 LVQ neural network serial implementation:

The global architecture of the LVQ neural network consists of 3 blocks (figure 2):

1- A block of internal ROM memories used to store fixed LVQ weights.
2- A column of K neurons (K=25). For each neuron j of the LVQ output layer, the L1 norm distance d_j must be computed ($d_j = \sum_{i=0}^{N-1} |X_i - W_{ij}|$, N is the number of inputs of the LVQ, in our case N=23). The architecture of operator $|Xi - W_{ij}|$ is formed with a serial on-line subtracter and a serial on-line absolute value processing. The global architecture of the distance computation is given in figure 2. The rather small area of serial operators allows to perform all computations simultaneously by means of a column of N subtracters followed by a column of N absolute value processing. It provides N outputs that are connected to the N inputs of a simple tree of serial adders. Weights W_{ij} are fixed, and they are stored in the internal ROM of the FPGA..
3- A serial on-line comparer to select minimal distance d_i and the corresponding winner neuron.

The architecture of the K-inputs comparer of the third block is specific to the requirements of an LVQ: it simultaneously extracts the minimal distance and the number of the winner neuron in serial on-line architecture (figure 2).

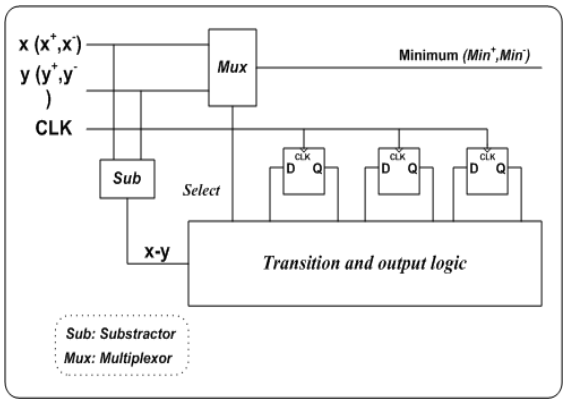

Figure 3: Schematic Implementation of on-line Comparator Element (CE)

We used on-line Comparator Elements (CE) capable of extracting the minimum of two redundant radix-2 numbers. An on-line algorithm with zero on-line delay is developed [14]. Based on this algorithm a finite state transducer, with 5 states and15 transitions, is derived. Figure 3 shows a schematic implementation of our on-line radix-2 comparator.

To carry out a comparison of the 25 inputs, we used the on-line CE arranged in stages; with the output of neurons connected to the first stage on-line CEs. The output of the first stage on-line CEs are connected to the inputs of the second stage CEs and so on figure 2. The last stage has a single on-line CE whose output is the minimum distance. To extract the winner neuron number, we decode the output controls of the on-line CE first stage. This extraction is performed after minimal distance processing. The global comparator is composed of stages with delay 5.

3. Results and discussion

3.1 Vigilance level classification**:**

The results presented below and related to the application of the neuronal tools on EEG signal portions recorded in the various subjects, are described and analyzed in order to obtain the best approach to quantify the various states of vigilance.

The LVQ network architecture includes 23 units on the input layer, which represent the 23 spectral bands, and 25 units on the output layer that characterize the two states (sleep and awakening).

The global performance is first computed for all subjects. One LVQ network is learned with the training corpora of the four subjects and is globally tested with all the test corpora. This experiment yields a total success rate on the test corpus of 76.73% with a recognition rate of the Sleep-Wakening states of 72.28% and 81.19% respectively (see table 2). The performance computed for each subject is presented in table 3.

Table 2: LVQ global performances for all subjects (task: awakening and sleep state recognition)

	TRAINING CORPUS		TEST CORPUS	
	WAKENING	SLEEP	WAKENING	SLEEP
SUCCESS RATE (%)	95.36	88.17	81.19	72.28
TOTAL SUCCESS RATE (%)	92.01		76.73	

Table 3: LVQ performances for four subjects (awakening and sleep states)

		Subject 1	Subject 2	Subject 3	Subject 4
Training corpus	Awakening	97.42	96.14	100	98.23
	Sleep	93.84	97.48	100	100
	Total	95.63	96.81	100	99.11
Test corpus	Awakening	100	92.70	100	93.70
	Sleep	100	85.78	100	100
	Total	100	89.24	100	96.85

3.2 Computation time and implementation area on FPGA:

Firstly, hardware performances were independently evaluated for each module of the network architecture. Then the whole network has been studied. Table 4, lists results in terms of computation time and implementation area for the different arithmetical operators, control blocks, data storage blocks and the whole network. The results are given according to Xilinx ISE 9.1 synthesis tool with a 25-MHz frequency.

Table 4: Results of the LVQ implementation

Block type	Delay (cycle clock)	CLB use rate		
		Slices (2 per CLB)	LUT	FF
On-line substracter/adder	2	5	8	4
On-line 23-inputs adder	6	56	98	18
On-line absolute value	1	2	4	-
On-line Comparator Element	-	17	32	3
On-line 25-input Comparator	5	425	797	124
On-line $\|Xi-Wij\|$	4	7	12	8
On-line neuron	10	241	401	214
parallel neuron	3	1390	1936	586
ROM (W_{ij} storage)	-	3	4	-
LVQ: 23 inputs, 5×5outputs [LSBF]	34	12286	14557	17659
LVQ: 23 inputs,5×5outputs [On-line]	23	6190	10701	5275

In the previous work we have evaluated parallel and LSBF serial implementations [10]:

1- Using standard 8-bit **parallel** arithmetics, each neuron requires 1390 slices so that the Virtex FPGA available on the RC1000-PP board is too small to fully implement the LVQ neural network: at most 9 neurons may be simultaneously implemented. Furthermore, such arithmetics would require a too large number of I/O pads for the Virtex.

2- A serial arithmetics appeared as a solution to reduce the implementation area. We develop a solution using a standard serial **LSBF** arithmetic in [10], still using the standard L2 norm for the LVQ. The results are quoted in the last row of table 4. This solution has allowed a fully parallel implementation of the whole network with 12886 slices. It uses the whole FPGA surface. Decision time *Texec* is equal to 1.37 μs with a 25 MHz clock frequency. The approach requires the implementation of an internal SRAM memory block to reverse the bit propagation order before comparison (MSBF mandatory).

This has led us to develop a solution using **on-line** arithmetic operators. Nevertheless, the LVQ model includes operations that must be performed in an **MSBF** mode (the comparator). Therefore, choosing to develop a version that uses the L1 nor*m* for the LVQ, and to implement it with on-line arithmetics has two advantages: unlike the LSBF solution, the implementation area is reduced by half. Bsides, the pipeline is fully exploitable (for consecutive input vectors), whereas the above LSBF solution uses a buffer that breaks the pipeline.

The results of the three solutions (parallel, LSBF and serial MSBF) are quoted in table 4.These results characterizing hardware performances were independently evaluated in terms of computation time and implementation area for the different arithmetical operators, control blocks, data storage blocks and the whole network. They are given according to Xilinx ISE 9.1 synthesis tool with a 25-MHz frequency.

It can be concluded that the on-line LVQ implementation requiring 6190 slices and taking 0.92 µs at a 25-MHz clock frequency is the best solution

4. Conclusion

This current paper describes an efficient parallel LVQ neural network implementation (used to separate vigilance states in humans from EEG) on FPGA using on-line serial arithmetic operators. L1 norm was adapted to measure the distance, thus reducing the computational complexity. Therefore, the main advantage of this implementation is the strong relationship between the neural architecture and its hardware realization. The generated circuit contains only the necessary operators and communications whatever the neural network size parameters are. The choice of an FPGA device is justified by its flexibility, parallelism and reduced consumption. The required implementation had to minimize the inputs/outputs pin, and maximize internal

parallelism to lead to a very low consumption.

Furthermore, this solution enables us to deal with more accurate numbers, whereas the precision problem is critical for analog implementations. The use of on-line arithmetics also reduces the circuit size and is compatible with all operators of the chosen LVQ model. The used technological choices are aimed at reaching area-saving and an entirely parallel solution where all the neurons function and treat all their input simultaneously. The performances obtained by our implementation have largely achieved our aims (very low-power parallel implementation).

Having acquired some experience in this field, we are now working on implementing the Learning Vector Quantization (LVQ) used to detect artefact states and to determine thereafter the physiological state of the subject (awakening or sleep states). We are following the same approach developed in this paper except that the precalculated weights have been modified. Once this phase is achieved, our embedded system is fully amenable.

References

[1] K. Ben Khalifa, M.H. Bédoui, M. Dogui,F. Alexandre, 2004. Alertness States Classification By SOM and LVQ Neural Networks International Journal of Information Technology IJIT- Vol (1) N°4, 228-232.

[2] T.P. Jung, S. Makeig, M. Stensmo and T. Sejnowski, 1997. Estimating alertness from the EEG power spectrum. IEEE Transactions on Biomedical Engineering, 44:60–69.

[3] A. Vuckovic, V. Radivojevicb, A.C.N. Chena, and D. Popovica, 2002. Automatic recognition of alertness and drowsiness from EEG by an artificial neural network. Med Eng Phys, 24(5):349–360.

[4] S. Roberts, I. Rezek, R. Everson, H. Stone, S.Wilson, and C. Alford, 2000. Automated assessment of vigilance using committees of radialbasis function analysers. In Proceedings of MEDSIP-2000.

[5] S.L. Joutsiniemi, S. Kaski, and T.A. Larsen, 1995. Self-organizing map in recognition of topographic patterns of EEG spectra. IEEE Transactions on Biomedical Engineering, 42:1062–1068.

[6] D. Macq, M. Verleysen, P. Jespers, and J.D. Legat, 1993. Analog implementation of a Kohonen map with on-chip learning. IEEETrans. on Neural Networks, 4(3):456–461.

[7] N. Vassilas, P. Thiran, and P. Ienne, 1996. On modifications of Kohonen's feature map algorithm for an efficient parallel implementation. In Proceedings of ICNN96, pages 932–937.

[8] T. Nakamoto, T. Ozawa, Y. Shibata, 2003. Improvement of odor recognition chip, Proceedings of IEEE Sensors, vol. 2, 22-24 Oct, 2003, pp. 1203-1208

[9] M. Kugler, M. Lopes, 2007. A Configware Approach for the Implementation of a LVQ Neural Network", International Journal of Computational Intelligence Research, vol.3, no.1, pp. 21-25.

[10] K. Ben Khalifa, B. Girau, F. Alexandre, M.H. Bedoui, 2004. Parallel FPGA implementation of self-organizing maps ICM 2004 Proceedings. The 16th International Conference on Microelectronics, 709 – 712.

[11] Xilinx editor, 2002. The Programmable Logic Data Book. Xilinx.

[12] T. Kohonen, 2001. Self-Organization Maps. Springer.

[13] M.D. Ercegovac, 1984.On-line arithmetic: an overview. In SPIE, Real Time Signal Processing VII, pages pp 86–93.

[14] A.M. Nielsen, 1997. Number systems and Digital Serial Arithmetic. PhD thesis, Odense University.

Mohamed BOUBAKER received the diploma degree in electrical engineering from ENI of Tunis, TUNISIA in 1991. In 2002, he received the Master degree in Computer Sciences from ENI of Sfax, TUNISIA. Currently, he is a PhD student. His research interests include a connectionist system able to discriminate vigilance states with a good accuracy and the implementation of many models of neural networks on FPGA.

Khaled BEN KHALIFA received his MSc in Physic Microelectronic and his DEA in Materiaux et Dispositive pour l'electronique and a PhD diploma in Physics-Electronics from the Science Faculty of Monastir, Tunisia, in 1997, 1999 and 2006, respectively. His current research interests include hardware realization for neural networks and heterogeneous multiprocessor architectures. He is working in collaboration with LORIA Laboratory (CORTEX Project) , Nancy Cedex France.