

FPGA implementation of a synchronous and self-timed neuroprocessor

Raygoza-Panduro J.J., Ortega-Cisneros S., Boemo E.
Escuela Politécnica Superior, Universidad Autónoma de Madrid, España
jjraygoza@uicm.net, susana.ortega@uam.es

Abstract

This article presents the implementation of a neuroprocessor based on a self-organizing map (SOM) architecture. The processor presents a hybrid structure both synchronous and self-timed. Where the neuronal network blocks (SOM) are synchronized with a protocol of 4 phases, for the control of data flow.

The neuroprocessor was designed for the analysis and classification of tension deformation patterns of the knee ligaments. The circuit is programmable and recognizes different sequences of movement patterns for a knee joint with damage to the anterior cruciate ligament (ACL). This design is part of an electronic system for the rehabilitation of injuries to the ACL and the dynamic study of the knee. The circuit is implemented in an FPGA Virtex II.

1. Introduction

The neuroprocessor is designed to carry out the processing of signals from strain gauge sensors, dedicated to detecting the superficial tension deformation of the tendoligamentar structures of the knee joints of laboratory cats. The project is aimed at the analysis and study of injuries to the anterior cruciate ligament (ACL) [1,2]. The work of the neuroprocessor consists of classifying the patterns of tension deformation characteristic of each movement. The classification of the patterns is made in real time by the processor of neuronal architecture. The control of the data processing of the network is centralized in an asynchronous control unit that regulates the data flow by an SOM processor according to the requirements of each movement.

This work presents the architecture, implementation and results of the neuroprocessor designed with the purpose of utilizing a system that processes patterns with a neuronal structure. It is programmable so that it recognizes different movements and continuous sequences, and to maintain the classification of the

patterns in synchrony with the changes of position of the joint. In addition, reconfigurable circuits FPGAs were used, covering the necessity to have a modular circuit that is mobile, which counts on the benefits of a system of flexible recognition of patterns and on a unit of own programming.

2. Architecture of the neuroprocessor

The neuroprocessor that appears in this article is implemented in a FPGA virtex II. The circuit is divided into two main modules:

- Neuronal module SOM with self-timed synchronization.
- Control module for executing operations.

The execution of the microoperations of the neuroprocessor is made by means of a fetch cycle that establishes the sequence of general operations of the processor in agreement with a main program kept in a ROM memory.

The neuroprocessor incorporates in the neuronal structure a sequential control of instructions that allows it to program processing pattern routines, in function of a main program. It also includes an out port that feedback to an electrical muscular stimulator, to carry out the muscular compensation of unstable movements detected by the system of recognition of patterns or to control a routine of temporary impulses based on the patterns classified by the neuroprocessor.

Figure 1 shows the general diagram of blocks of the neuroprocessor. The circuit has 3 entrance buses, 2 of these are of 15 bits that correspond to the input datas of the patterns of movements (X1 and X2), the third bus is of 16 bits, functioning as an entrance port for connection to a keyboard or an interface of an external circuit, in addition it has 6 exits of 11 bits that show the levels of activity of the winning neurons (G1 to G6).

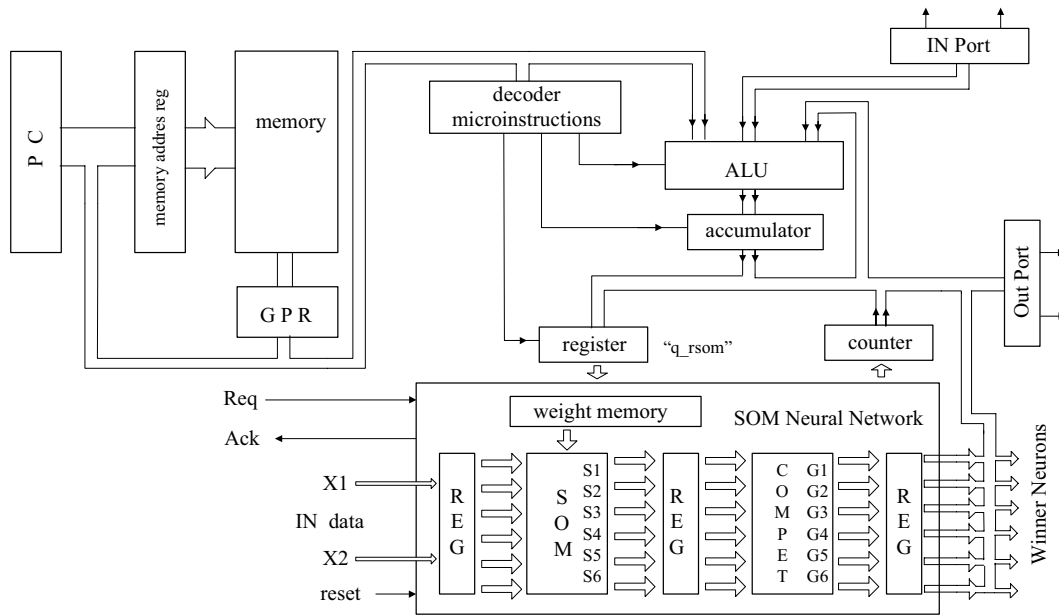


Figure 1. Block's diagram of the neuroprocessor.

Both main modules of the neuroprocessor can work in independent form and to interact together during the execution of a program of pattern classification.

3. SOM neuronal module with self-timed synchronization.

The architecture of SOM network consists of 2 neurons of entrance and 6 neurons in the exit layer. The matrix of weights " w_{ij} " is of the order of 2×6 . The Algorithm of SOM network [3,4] can be described in the following way: let us consider a vector of entrance " x " for each neuron " i ".

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \quad (1)$$

Each neuron of the network is connected by the weights array " w_{ik} ."

$$\mathbf{w} = (w_{1k}, w_{2k}, \dots, w_{ik}) \quad (2)$$

Where " i " is the total number of neurons of the entrance layer and " k " is the total number of neurons of the exit layer. In order to find the most appropriate value of the weights with respect to the vector of entrance " x "; $d(j)$ is applied to the equation of the euclidian distance:

$$d(j) = \sum_i (w_{ik} - x_i)^2 \quad (3)$$

The minimum distance for all the vectors of entrance is calculated and the $d(j)$ of the least value is selected and the weights are updated according to the following equation:

$$W_{ik} = w_{ik(\text{old})} + \alpha [x_i - w_{ik(\text{old})}] \quad (4)$$

Where " α " is the learning reason, which is updated, later the value of " α " slowly decreases based on the time (cycles) [5,6]. The training process can be observed through the deployment of the map of the matrix of weights, calculated in each iteration. If the initial values were near the correct values, the map of the matrix of weights will quickly order in orthogonal form. In the opposite case, the training process will require more calculation cycles until an orthogonal map is obtained.

$$w_{ik}(t+1) = .5\alpha(t) \quad (5)$$

The recovery function of the SOM network is carried out once the cycle of training has finished. With the matrix of weights obtained, the network is able to group the vectors of entrance by means of the calculation of the minimum range between the vector of entrance and the matrix of weights. The neuron with the minimum value activates and is considered the winning neuron. The entrance vectors closest to one another, are grouped and activated to the same neuron according to the patterns trained by the network. The frequency of winning neurons is entered with the following equation:

$$Nt_k = \sum_1^n Neu_k \quad (6)$$

Where " Nt_k " is the total number of times that neuron " k " wins, and " Neu_k " is neuron " k " of the exit

layer, that wins in each calculation. The architecture of SOM network is shown in figure 2.

The general diagram of the network implemented in the hardware is shown in figure 3. The entrance vectors $X1$ and $X2$ as well as each weight " w_{ik} " have a length of word of 15 bits. The general circuit has a request entrance "req" and an exit of recognition "ack", a common line of reset that initializes the calculation of the network for each input data.

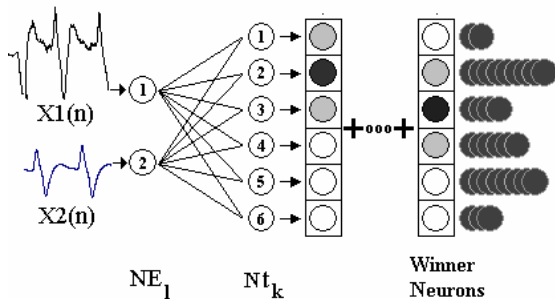


Figure 2. Architecture of the SOM neuronal network.

The network exit is composed of 6 channels of 11 bits of word length, each one of them represents the frequency of activity of the winning neurons in the exit layer of of the SOM network. The circuit is composed of 4 main blocks:

- Self-timed control that regulates the data flow of the network.
- Somdist Circuit that contains the arithmetical operations of the euclidian distance and the multipliers of the matrices of weights and input datas.
- Compet1 Circuit made up of comparison units of the partial results of the active neurons in the exit layer, and Compet2 which contains a second block of comparison units that establish the winning neurons of the exit layer in each processing cycle and the block of counters for the frequency of these.
- Ram memory block that keeps the weights corresponding to the training of the network of various patterns.

3.1. Self timed control block

The network control circuit SOM is developed using a self-timed (ST) [7] communication protocol, that controls the general data flow to the entrance and through the entire network, allowing signal processing according to the feeding speed of these. The network circuit SOM is segmented with registries at the exit of each processing module. The data transfer of the network is synchronized with the pulses "xi" that the

asynchronous control block generates, the sequence of pulses of the post layout simulation made in ISE of Xilinx is shown in figure 4, and each pulse is obtained as a result of the interchange of the protocol synchronization of 4 self timed phases [8]. The latency of the SOM network is 211 ns and each pulse "xi" has an approximated value of 40ps.

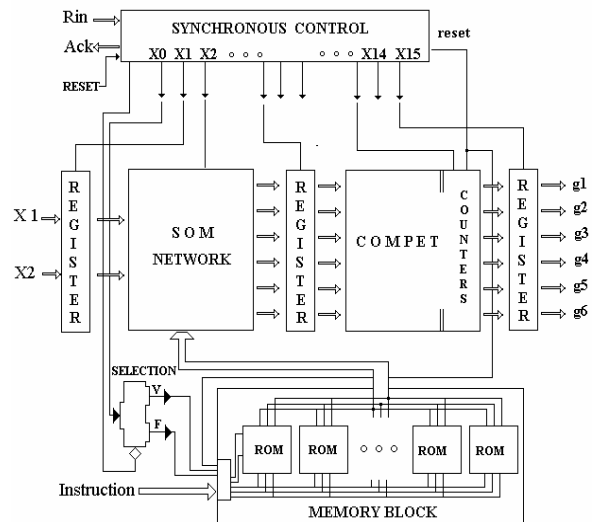


Figure 3. SOM ST neuronal network.

Each asynchronous control block is composed of storage elements RS, and logical ports, formed in a pipeline structure, that are synchronized by means of an intermediate delay between each asynchronous control block. This delay should be equal to the processing of the combinatorial circuit that it controls [9].

3.2. Somdist block

The somdist block contains the multipliers and adding circuits that make the calculations of the minimum range between the vectors of entrance and the matrix of weights. The calculation of each neuron is applied to the compet block 1 and 2 to determine the winning neuron of the exit layer of the network.

3.3. Compet blocks 1 and 2

The compet 1 and 2 modules compare the result of the 6 neurons of the exit layer of the network. These are implemented using comparators of 30 bits word length each. Its main function consists of determining which exit of the neurons has the minimum value of activation (euclidian minimum range). The neuron that obtains the minimum value is considered the winning

neuron and the frequency counter corresponding to that neuron is increased. In order to control the data flow through the circuit, in the entrances as much as the exits of each subcircuit, it is segmented with registries that allow the passage of the data in synchrony with signals “xi” of control ST.

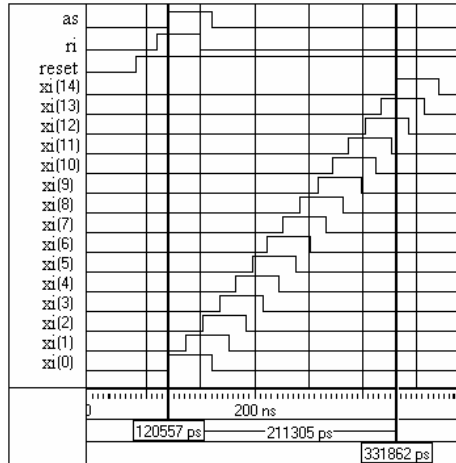


Figure 4. Activation pulses xi.

The 6 entrances F1 up to F6 represent the frequency of the neurons activated in the exit layer. The first level of comparison groups the 6 entrances in three pairs, and they are compared of the following way:

If $L1=L2$ then $L1 \rightarrow S1$, and If $L2 > L1$ then $L2 \rightarrow S1$.
 If $L3=L4$ then $L3 \rightarrow S2$, and If $L4 > L3$ then $L4 \rightarrow S2$.
 If $L5=L6$ then $L5 \rightarrow S3$, and If $L6 > L5$ then $L6 \rightarrow S3$.

The second level we have:

If $S1=S2$ then $S1 \rightarrow S4$ and If $S2 > S1$ then $S2 \rightarrow S4$.

The last stage of comparison consists of:
 If $S3=S4$ then $S3 \rightarrow Z$ and If $S4 > S3$ then $S4 \rightarrow Z$.

Where Z represents the exit of the circuit, and is minimum level of neuron activity.

The compet2 circuit compares the 6 levels of neuron activity against the minimum value of these, as shown in figure 5. The neuron with the minimum value is applied to each comparator and the one that fulfills the condition $L_n = Z$ causes the exit to be equal to “1” enabling the subcircuit of selection in the true exit,

allowing the passage of the request pulse “xi”, which increases the account of winning neuron “G” of the exit.

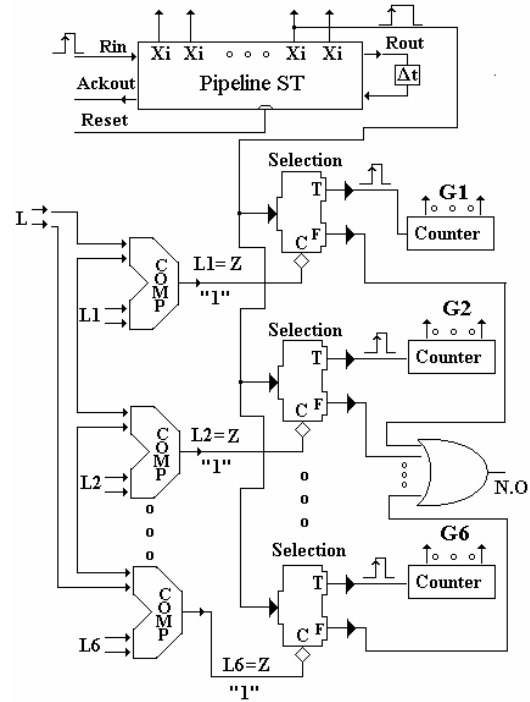


Figure 5. Compet2 circuit.

3.4. Weight memory block

The processing capacity of each SOM network is acquired during the training phase and in this period the weight matrices are generated that contain the required values to group the different data from the entrance vectors. In the SOM ST network, the memory block has the function of storing the weights of the matrices obtained from the different movement patterns. The block is composed of 12 RAMB16s18 memories of 16 x 1024, with 16 bits in such a way that a word in this bus directs all the memories and their corresponding exits to each one of the “ w_{ij} ” of the matrix.

Figure 7 shows the interconnection of the memory block of the neuronal SOM ST network, consisting of 12 buses of 0 to 14 bits length that are connected to the entrance of each network weight. With the incorporation of the memory block, the neuronal network can be programmed with different types of matrices and therefore it can change its capacity of pattern classification just by writing a word in the address bus of the memory block, without modifying any connection or any part of the network architecture.

4. Occupation of SOM RNA

The results of the implementation of the circuits in the FPGA Virtex II Xc2v1500-4ff896 are summarized in the figure 6. Showing the distribution of resources used by SOM network:

- Somdist module occupied 6% of slices, 6% LUTs and the total of equivalent ports for the design is 114656.
- Compet module is 3% of slices, 3% LUTs, 1% register; total of equivalent ports for the design is 4548.
- Asynchronous control ST is 3% slices, 1% LUTs, total of equivalent ports for the design not including macros is 180.
- Memory block is 1% slices, 1% LUTs, 25% of the memory blocks; total of equivalent ports for the design is 786507.
- Neuronal network SOM ST has 13% slices, 9% LUTs, 1% register, and total of equivalent ports for the design not including macros is 907268.

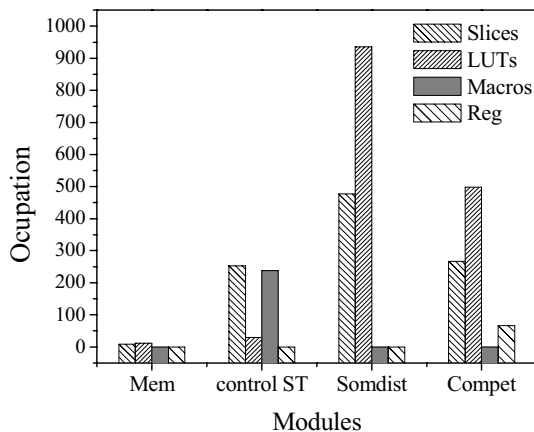


Figure 6. Occupation of network SOM.

5. Operations of control circuit

The unit of microinstructions executes the program of instructions and coordinates the process of the neuronal module. The memory and register unit store the executive monitor program and the temporary data, which are generated during the calculations. The arithmetic and input-output unit executes the addition, reading and writing operations of the ports.

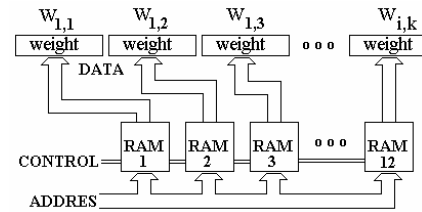


Figure 7. Weight memory.

In the RNA ST module, the transfer of data, the execution of the operations and the presentation of the results are done in a parallel way. The sequential control module executes in a cyclic way the instructions, which operate depending on the speed of a common clock line of the system, doing a cycle of search before carrying out each instruction. Consequently, this generates the existence of two modules with different systems of synchronization, which cohabit in a hybrid, self-timed and synchronized general circuit, implemented by a reconfigurable FPGA device.

The sequential control module is implemented using the following blocks:

- Control of microinstructions.
- Program counter.
- Codification register.
- Neurooperation register.
- Memory.
- Accumulator.
- ALU.
- In/Out ports.

The operation algorithm of the neuroprocessor is shown in figure 8. The sequential control module divides its operation into two main tasks:

- Execution of the fetch cycle.
- Implementation block of the neuroinstructions that interact with the RNA.

The instruction cycle consists of initializing the program counter to transfer the account to the address register, that stores the word and keeps it present in the address bus of the ROM memory and this automatically displays the command code stored in the signaled address. Simultaneously, the program counter is increased to provide the next address. The control block decodes the word of the memory in two parts, command code and data field.

The command code enables the function of operation by means of a decoder of several control lines that activate or deactivate the components of the processor. The field dedicated for data remains available in the accumulator for any arithmetical or logical operation.

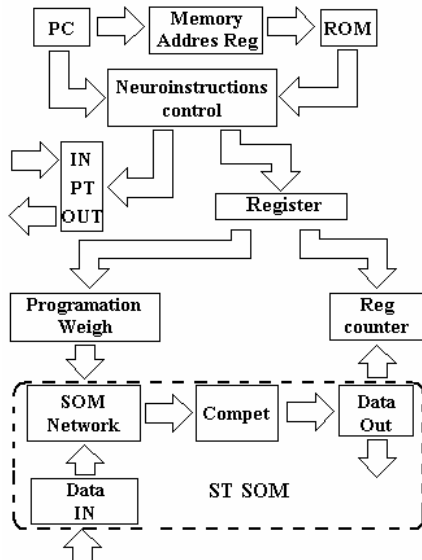


Figure 8. Flow chart of the neuroprocessor.

5.1. Neuroprocessor instructions

The neuroprocessor has two types of operational instructions: direct codification instructions and neuronal classification instructions.

The direct codification instructions are composed of a group of 13 instructions divided into: 2 input-output, 1 arithmetic, 1 logic, 2 jump, 2 register transference, 3 reference to neuronal network and 2 control. The neuronal classification instructions consist of up to 1024 different classes of processing of patterns. These correspond to each matrix of weights stored in the memory block of the ST SOM module.

5.2. Instructions referring to the RNA

The instructions of direct codification with reference to the neuronal network are those that carry out the interaction of the SOM network processing with the sequential control, in this way, allowing the classification of a series of recurrent patterns of tension-deformation or to interchange movements of continuous way. The incorporation of this type of instruction allows the RNA module to be programmed in a more flexible way for the development of the experimentation. The description of two instructions with reference to the neuronal network appears below.

1) The implementation of instruction “PRG_SOM” is shown in figure 9. The operation sequence begins with the loading of the accumulator with data that corresponds to the address of the weights matrix. The sequence is the following:

- (1) and (2) transfer of the data to the accumulator.
- (2) and (4) transfer of the accumulator to the programming register of the network.

Once the data is captured by the address register “q_rsom”, the control word remains fixed and the processing function of the network is established.

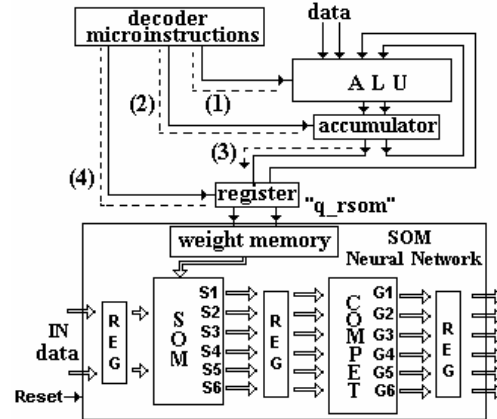


Figure 9. Neuroinstructions “prg_som”.

The implementation of the instruction “READ ACCOUNT” is shown in figure 10. The counter exit transfers the data “q_coun” to the accumulator. The sequence for the microinstruction block is the following:

- (1) Activate the multiplexor of the ALU to transfer the contents of the counter to the accumulator.
- (2) Capture the accumulator.

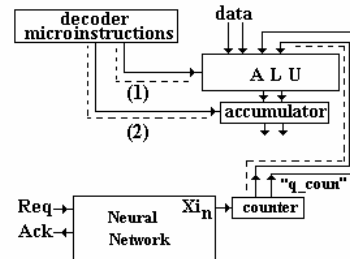


Figure 10. Neuroinstructions “read account”.

6. Results of the pattern classification.

The neuronal processor was trained to classify two classes of patterns, corresponding to movements of healthy knees and knees with an injured anterior cruciate ligament.

The registry of the sensors of deformation in the experiment is constructed by means of series of movements of the joint (flexion-extension and combined) which are made in independent form and in a series of continuous or paused repetitions. Part of the results is shown in figure 11. These patterns correspond to two series of 10 movements of a cat’s

anterior locker joint. (a) Response of Medial Collateral Ligament MCL of an injured joint of the ACL. (b) Movement of anterior locker of a healthy knee.

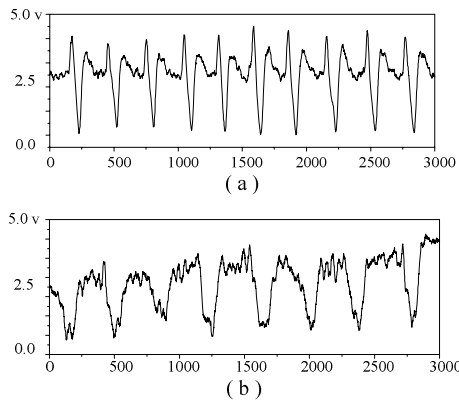


Figure 11. Patterns of tension deformation.

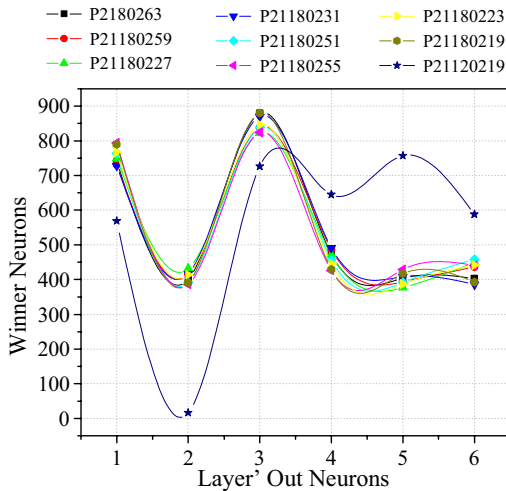


Figure 12. Result of the frequency of winning neurons of an injured joint.

These patterns represent the behavior of the tension deformation of the ligament structure in the healthy and injured joint.

From the results of the pattern classification using an SOM network; we found that of 100 processed patterns of a same type, 80 were recognized by the network and 20 were not grouped correctly, in accordance with the pattern characteristic of the exit of this movement. As observed in the graph of figure 12. The levels of activity of the winning neurons in the exit layer of the network represent the vector of behavior of the MCL ligament of the anterior locker of the injured joint. The results obtained from the SOM network with the matrix of weights of a stable ligament anterior locker correspond to 110 movement patterns.

- P300207
- P300211
- ▲— P300215
- ▼— P300209
- ◆— P302203
- ◆— P300257
- ◆— P300210
- P303203

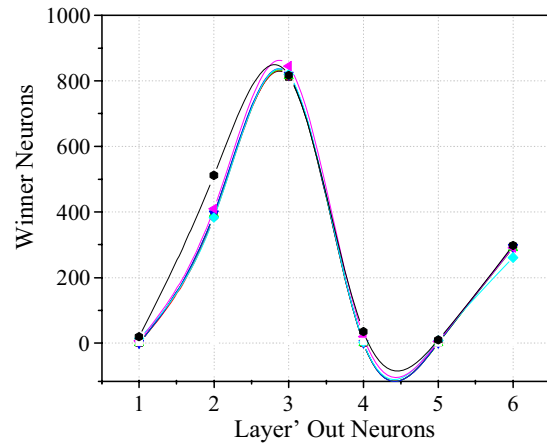


Figure 13. Results of the winning neurons in a healthy joint.

From the combined total movements of the same class, a correct classification of 85 patterns was obtained, in where the exit response of the winning neurons provided a correct result. In figure 13 we can see the levels of activity of the winning neurons of the exit layer for patterns of a healthy joint anterior locker classified by the network.

7. Results of the implementation.

The results of the occupation of the neuroprocessor are summarized in the table 1 corresponding to the FPGA Virtex II xc2v1500-ft896. As observed, it occupies 17% of the total slices, 12% of LUTs and 3% of the register.

| Circuito | Slices | LUTs | Reg | Gates |
|----------------|--------|-------|-----|---------|
| PC | 3 | 5 | 5 | 73 |
| GPR, MAR | 12 | 23 | 12 | 312 |
| Memory | 21 | 41 | 0 | 65,788 |
| Accumulator | 33 | 60 | 21 | 531 |
| Control | 96 | 76 | 116 | 1,395 |
| RNA SOM | 1,161 | 1,658 | 361 | 910,106 |
| Decoder | 34 | 37 | 0 | 315 |
| neuroprocessor | 1,368 | 1,928 | 551 | 978,853 |

Table 1. Occupation in the FPGA.

We can conclude that this device provides us with the capacity to increase the group of instructions.

Figure 14 shows the final layout of the SOM neuronal processor.

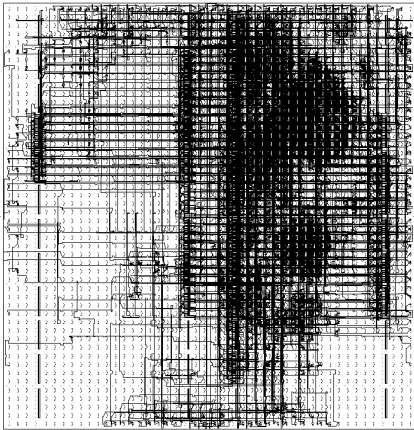


Figure 14. Layout of the neuroprocessor.

8. Conclusions

The implementation of the neuroprocessor, consists of a hybrid circuit of two modules with different synchronization (self-timed and synchronous), resulting in the combination of a sequential programming circuit that controls a ST circuit of parallel processing of neuronal architecture.

The architecture utilised in the SOM network hardware allows easy programming. This is obtained by simply changing the value of a register that contains the memory address of the weights of the new pattern.

The processor allows adjustments to be made to the network for different movements in continuous form maintaining the classification process of the patterns in continuous form. According to the main program kept in the ROM memory.

The asynchronous control of data transfer, allows an automatic adjustment of the data processing according to the speed required by the movements of the joint. Causing the circuit not to carry out any operation and only processes the data each time it is required through a request pulse.

9. Acknowledgment

This work has been financed by the National Advice of Science and Technology of México (CONACYT).

10. References

- [1] Beynnon B.D, Pope M.H., Wertheimer C.M. Johson R.J., Flemming B.C.,P.A, Nichols C.E. and Howe J.G.: “The Effect of Funtional Knee-Braces on Strain on the Anterior Cruciate Ligament in Vivo”, *The Journal of bone and Joint Sugery*, vol 74-A N° 9, October, 1992, pp. 1298-1312.
- [2] Shelburne kebin B., Pnady Marcus g., Anderson Frank C., Torry Michael R., “Pattern of anterior ligament force in normal walking”, *Journal of Biomechanics*, vol 37, 2004, pp 797-805.
- [3] Kohonen Teuvo, “Self-Organizing Map”, *Proceedings of the Institute of Electrical and Electronics Engineers*, vol 78, 1990, pp 1464 – 1480.
- [4] Kohonen Teuvo, “New Development and Applications of Self-Organizing Maps”, *Proceedings of the Institute of Electrical and Electronics Engineers*, 1996, pp 164 – 171.
- [5] The MathWorks, “Neural Network Toolbox 4”, *User’s Guide The MathWorks, Inc. MatLab*, www.mathworks.com. 2000.
- [6] Howard Demuth, Beale Mark., “Neural Network Toolbox” For Use with MatLab, *User’s Guide, Version 4*, 2004.
- [7] Beerel Peter A., “Asynchronous Circuits: Increasingly Practical Desing Solution”, *Proceedings of the International Symposium on Quality Electronic Desing, IEEE*, 2002.
- [8] Furber S. B. and Liu J., “Dynamic logic in four-phase micropipelines” *Advanced Research in Asynchrous Circuits and Systems, 1996. Proceedings.*, Second International Symposium on, March 1996, pp 11–16, 18-21.
- [9] Sutherland I. E., “Micropipelines”, *Communications of the ACM*, vol 32, No. 6, June 1989, pp. 720-738.