FPGA Implementation of Neural Network for EEG Signal Processing

[1,2]Thiago Pontes, [1]Rodrigo Braga, [1]Carla Becker,[2]Eduardo Costa, [2]Sérgio Almeida,
{thiago0b12, rodrigobraga.fuzz, becker.diniz}@gmail.com,{ecosta, smelo}@ucpel.tche.br

[1]Laboratório de Engenharia Biomédica – LEB
[2]Laboratório de Microeletrônica e Processamento de Sinais - LAMIPS
Universidade Católica de Pelotas – UCPel

Abstract

The brain computer interface (BCI) offers an alternative to improve life quality in patients which had lost the ability to control his body, with impairment of locomotion. It is possible with the BCIs systems, to control devices as wheel chairs or computer systems. Usually the electroencephalographic (EEG) signals are used tocommand these systems. In this paper, it is proposeda low-cost use of field programmable gate arrays (FPGAs) to process EEG signals for a Brain-Computer Interface. As a preliminary study, this work shows the implementation of a Neural Network for EEG signal processing. The preliminary tests with the proposed architecture for the activation function proved to be feasible both in terms of the requirement precision as well in processing speed.

1. Introduction

People around the round suffer from diseases that compromise and restrict their movements,causing perishing on a wheelchair or on a bed for the rest of their lives. The development of self-care equipment which can be controlled using EEG signals from their brains, can reduces the medical labor load,facilitates the patient's autonomy and consequently improves the quality of life. It is proposed in this paper, the hardware implementation of the BCI system proposed in [1], aiming at the processing of biological signals, more specifically, the classification and processing of EEG signals using artificial neural networks (ANNs).

In applications where area occupied and processing speed are critical, the use of field-programmable gate array (FPGA) devices has been gradually replacing the Digital Signal Processors (DSPs), mainly due to the great power of parallel processing, large capacity and cost competitiveness, that new FPGAs have [2]. The literature shows that the implementation of ANNs in FPGAs devices is feasible and very interesting [9], both in terms of capacity and cost.

2. The ProposedBCI System

The proposed BCI system [1], represented in fig. 1, processes data used from the BCI Competition II [3]. The dataset was recorded from a healthy subjectduring a feedback session. The task was to control a feedback bar in one dimension by imagination of left- or right-hand movements. The ANN algorithms implemented [1] in MATLAB obtained a mean accuracy of 84.3 % for the classification of the EEG signal between the left and right hand classes.

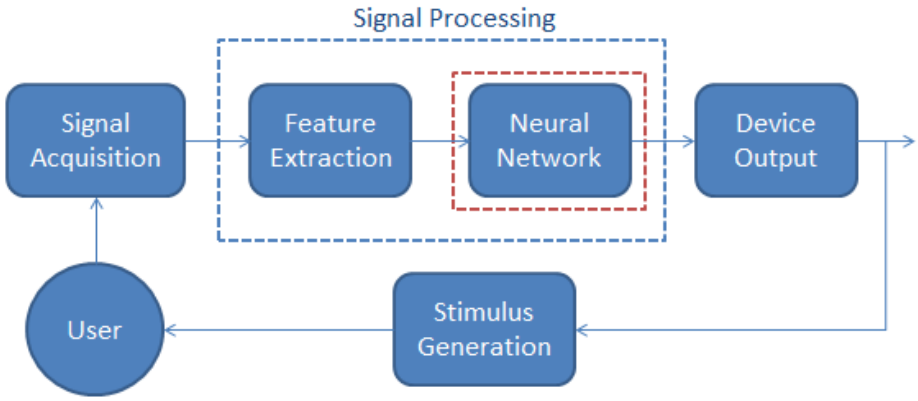

Fig.1 - Block diagram of a BCI system.

The work proposed in this paper is focused on the implementation of the Neural Network block, marked in fig. 1 on an FPGA device. The architecture used in its implementation and the results obtained are discussed in the flowing sections.The literature proposes the use of data with 8-bit floating point, as in [4]. However, this approach has as main drawback the imprecision of the results, which carries a high number of false positives ones [5]. To overcome this problem, this paper proposes the use of floating point representation with 32 bit-

width of precision, aiming at a more accurate comparison between the expected results and those obtained through simulations with MATLAB.

3. Network Architecture

In order to find the balance between speed and area used, the hardware implementation of the artificial neural network is being done with the elements that are shown in tab.1. All these elements are Xilinx IP Blocks [6, 7], available in their development tool, ISE.

Tab. 1 - Neural Network elements.

Element	Quantity
Adders	6
Multipliers	4
Dividers	1
ROMs	4

All the circuit elements work in 32-bit single precision floating point, then it is possible to obtain greater precision. However, this precision is obtained at the cost of increases in area occupied by each component, as well the total processing time. But the FPGA used can easily deal with this type of request and further permits greater flexibility in some network points. As will be explained below, the processing may occur in parallel in some points(increasing performance) and other ones it processes sequentially (decreasing the total area).For the proposed network architecture, it is possible to process in parallel, two of the first layer neurons (N1 to N10 as presented in fig. 2) at a time, and all six adders can work in parallelto accumulate the inputs of the output layer neurons (N11 and N12 as presented in fig. 2).

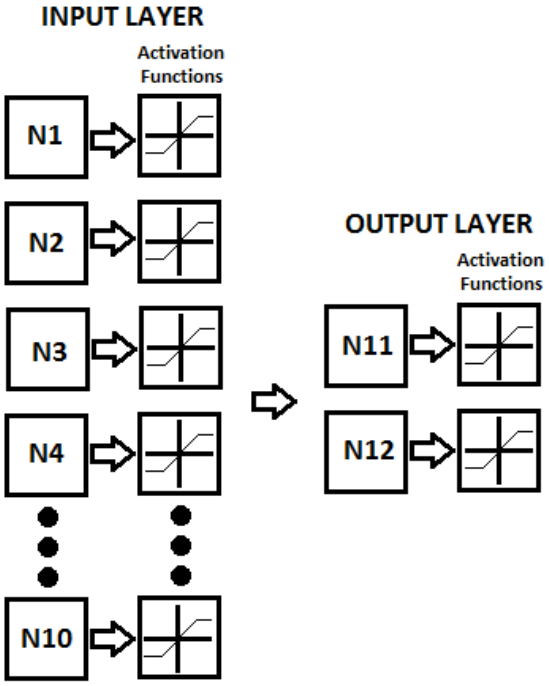

Fig. 2 - Block diagram of the proposed neural network.

3.1. Neuron Architecture

The architecture of a neuron is shown in fig. 3. It consists of: (i) two 32x24 ROM memories, and (ii) two multipliers and one adder. In the multiplier blocks a signal RDY indicates that there exist new operands to be processed. The underflow and overflow signals indicate errors during the operation. The signal invalid_op indicates that the operands are no valid numbers to be processed such as zero or a number greater than 32 bit-width. The same signals are used in the adder block. The signal SCLR clears the outputs when the results are obtained.

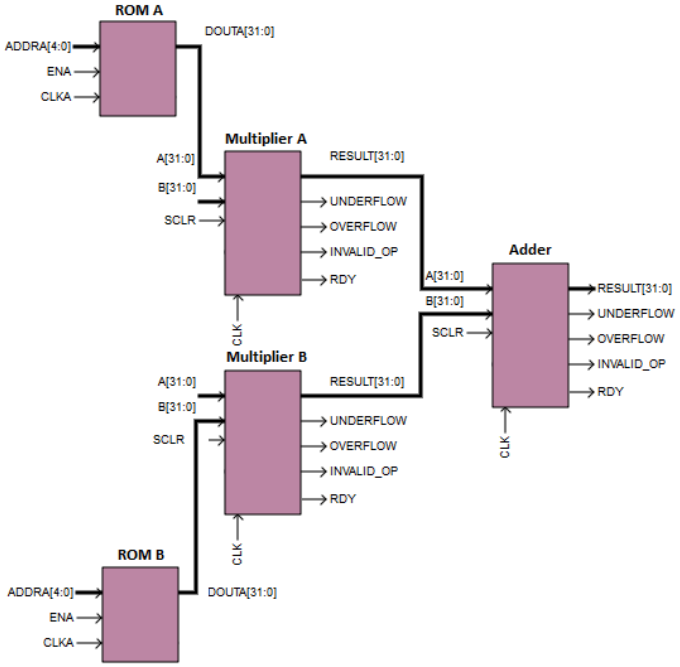

Fig. 3 - Block diagram of a single neuron.

3.2. Activation Function

The activation function chosen for this implementation was the log sigmoid [8], shown in eq. 1, mainly due to its range of results that are between zero and one, which facilitates binarizing the output. To implement this activation function, it was decided to an approach that generates the greatest possible accuracy over speed and area occupied;this approach is an iterative algorithm for calculating the exponential function, which requires the use of a divider circuit to be implemented.

(1)

The use of a divider ends up increasing considerably the total area of the circuit, but through resources available in Xilins tool ISE, it is possible to reuse some of the hardware resources, so that the divider circuit occupies an area equivalent of a normal multiplier, and the only drawback is the increasing number of clock cycles demanded to the circuit to show their result.

4. Results

The results obtained so far are concentrated on the implementation and testing of the activation function of the network. The results are shown in tab. 2 and 3. These tests were performed in the Spartan 3E – Starter Kit development board model XC3S500E, family 4fg320 from Xilinx. The results were promising, since it was used an FPGA, whose technology and area are inferior that the one that will be used to implement the network (Xilinx Virtex ® -5 XC5VLX110T FPGA).

As expected the area occupied by the circuit was about 28% of the used device, which provides the possibility to use the rest of the area for the implementation of other stages of the network, improving the final result of the tests.

The activation function was tested using data obtained from the simulation in MATLAB, proposed in [1]. In the tab. 3 one can observe that, in the worst case found by the activation function (higher value of the signal after preprocessing, 239.82, multiplied by the greater weight of 3.572 and, added with the higher bias, 1.041) requires about 200 clock cycles for the calculation. All other tests were: (i) the lowest computational cost (number equal to 0) and (ii) the higher computation cost value (-856.68) with itsreversed sign. In these further tests the number of necessary cycles were, as expected, about 12 for the number 0, due to the need of a single iteration to the exponential converges to 1, and about 240 clock cycles to the number-856.68. These additional 40 cycles are due to signal verification stages inside the algorithm, which consumes two more cycles per iteration. All these tests were performed using the standard board clock of 50 MHz, which causes, in the worst case presented, arefresh rate greater than 288 KHz.

Tab. 2 – Activation function implementation used resources.

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	3,785	9,312	40%
Numberof4 input LUTs	2,561	9,312	27%
NumberofoccupiedSlices	2,693	4,656	57%
Total Number of 4 input LUTs	2,622	9,312	28%

Tab. 3 - Activation function implementation timing results.

Activation Function input	Clock Cycles
856.68	~200
0	~10
-856.68	~240

5. Conclusions

This work shows the implementation of a Neural Network for EEG signal processing. The preliminary tests with the proposed architecture for the activation function, proved to be feasible both in terms of the requirement precision as well in processing speed. However, tests and adjustments must be done on the network, so that the optimum point between area occupied, precision and speed can be reached. As future work we intend to implement all the neural network and the pre-processing of EEG signals,in order to compare the obtained results in both approaches, software and hardware. It will enable improvements in both approaches until a fully embedded system can beachieved, which will allow further studies of visually evoked potentials and the development of a BCI.

6. References

[1] R.B. Braga, “Processamento de Sinais de EEG para uma Interface Cérebro-Computador.”, Technical Report. UCPel, Pelotas, 2011.

[2] An Independent Analysis: The Evolving Role of FPGAs in DSP Applications, 2007, BDTI (www.BDTI.com).

[3] BBCI, “BCI Competition II, dataset III description”. 2003. Available at:<http://www.bbci.de/competition/ii/Graz_description.doc>

[4] Sahin, S. Becerikli, Y. Yazici, S., “Neural Network Implementation in Hardware Using FPGAs”, *LECTURE NOTES IN COMPUTER SCIENCE*,2006, NUMB 4234, pages 1105-1112

[5] Omondi, Amos R.; Rajapakse, Jagath C. “FPGA Implementations of Neural Networks”, (Eds.)2006, XII, Ed. Springer.

[6] Xilinx, “LogiCORE IP Floating-PointOperator v5.0 ProductSpecification,” San Jose, March 1, 2011.

[7] Xilinx, “LogiCOREIP Block Memory Generator v6.1 ProductSpecification,” San Jose, March 1, 2011.

[8] Kwan, H.K.: ‘Simple sigmoid-like activation function suitable fordigital hardware implementation’,Electron. Lett., 1992, 28, (15),pp. 1379–1380.

[9] M. Krips, T. Lammert, and Anton Kummert, “FPGA Implementation of a Neural Network for a Real-Time Hand Tracking System”, Proceedings of the first IEEE International Workshop on Electronic Design, Test and Applications, 2002.