# FPGA Implementation of Artificial Neural Networks

*SAMI EL MOUKHLIS* [1], *ABDESSAMAD ELRHARRAS* [1] and *ABDELLATIF HAMDOUN* [1]

[1] Department of Electronics and treatment of information,
UNIVERSITE HASSAN II MOHAMMEDIA,
Casablanca, Morroco

**Abstract**

*In this paper, a method of classification of handwritten signature based on neural networks, and FPGA implementation is proposed. The designed architecture is described using Very High Speed Integrated Circuits Hardware Description Language (VHDL).*
*The proposed application consists of features extraction from handwritten digit images, and classification based on Multi Layer Perceptron (MLP).*
The training part of the neural network has been done by using MATLAB program; the hardware implementations have been developed and tested on an Altera DE2-70 FPGA.

***Keywords:*** *ANN, VHDL, FPGA,MLP.*

## 1. Introduction

In the last decades, the electronic devices production field has witness the birth of the FPGA (Field Programmable Gate Array). These platforms are the optimum and best choice for the modern embedded digital systems. The FPGAs offer low cost, powerful software development tools and true parallel implementations [1], these characteristics put them as a viable solution to implement Artificial neural networks.

Artificial neural networks (ANN) are widely used in different fields such as image recognition, their fast processing are based on a parallel architecture.

The important characteristics [2] of the network depend on its structure, the activation function and the learning mechanism.

The most commonly used algorithm of neural networks for pattern classification tasks is the backpropagation algorithm [3], which represents multilayer perceptron.
The main focus of this paper is to present a combined method for classification of handwritten signature which consists of three steps. A) A pretreatment step whose role is to extract characteristics from handwritten signatures by using MATLAB [4]. B) An off line training step where we calculated the weights of our ANN by using MATLAB neural network library for in order to calculate. C) The hardware implementation step which describes the design of A digital system architecture realizing a feedforward multilayer neural network, The network is implemented in Altera Cyclone II FPGA embedded in *Terrasic DE2-70 Board*[5].

## 2. Proposed design

The proposed design consists of an MLP neural network two layers which is one of the most architecture used in pattern recognition each layer is fully connected with its adjacent layers.
Each layer contains a number of neurons which is the processing element of the ANN, and activation function which is the most important arithmetic operation required for neural networks.

### 2.1 Architecture of the neuron

A Neuron can be viewed as processing data in three steps; the weighting of its input values, the summation of them all and their filtering by sigmoid function.
Figure 1 shows the block diagram of the designed neuron, the MAC unit which accepts a serial processing of weights and parallel inputs pairs, each pair is multiplied together and added to the next multiplication until the end. The result is fed in the activation function.

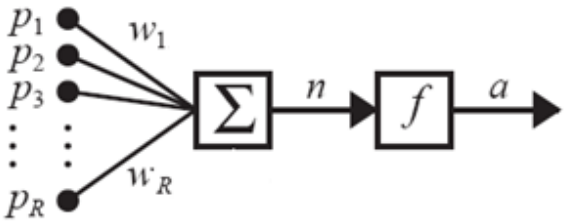


Fig. 1 Architecture of neuron.

The implementation of the trained network was made by VHDL language. This design is synthesized by Quartus. The figure 2 shows the block diagram of a neuron, it s composed of multiplier that multiply the weight and the input and an accumulator that mad the sum of all the multiplication and at the end the result is put in the activation block which validate or not the result.

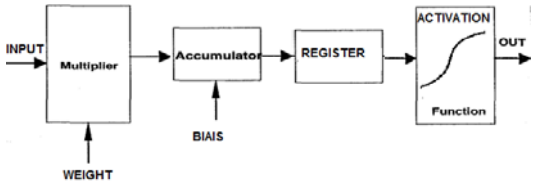


Fig. 2 Block diagram of a neuron

We implement our circuit on Cyclone II FPGA from ALTERA, the implementation of the neurons has been done by using 16-bit MAC (Multiply Accumulate) circuit, the sigmoid function is implemented both in the hidden an output layer.

## 2.1 Activation function

The activation function [7] is used to limit the value of the neuron output. It's one of the most important parts of an ANN. The sigmoid function (1) is the most frequently used activation function in backpropagation neural networks applications. It is not suitable for direct implementation because it consists of an infinite exponential series. For its implementation function on FPGA, we use Piece-Wise Linear Approximation which consists of a linear approximation of logsig function; the mathematical representation is shown in equation 2. This method has sufficient precision for our implementation.

$$\theta(v) = \frac{1}{(1+e^{-v})} \tag{1}$$

$$\theta(v) = \begin{cases} 1 & for \quad v \geq 4 \\ 0.0625v + 0.75 & for \quad 0 < v < 4 \\ 0.5 & for \quad v = 0 \\ 0.0625v + 0.25 & for \quad -4 \leq v < 0 \\ 0 & for \quad v \leq -4 \end{cases} \tag{2}$$

The figure 3 shows the RTL hardware circuit of the activation function, it gives an idea about the complexity and calculation that needed in the implementation of this function.

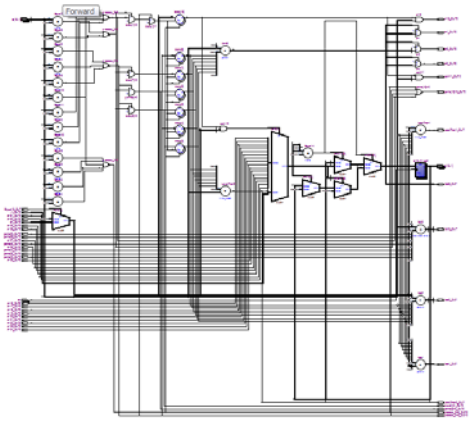

Fig. 3 RTL circuit of the sigmoid

## 3. Results

In this work the architecture of feedforward neural network used (20-6-1) layers; the network is composed of 20 inputs, the hidden layer[6] with 6 sigmoid neurons and the output layer with 1 sigmoid neuron. The network uses the parallelism and the rapidity of the FPGA to perform a parallel processing of the data.

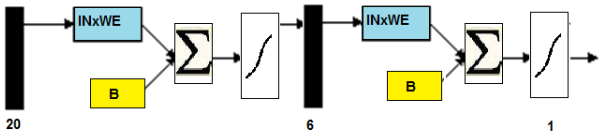


Fig. 4 The global network

For the neurons of the hidden layer the product of signed inputs and signed weights form an accumulated result to which we applied the activation function, the final outputs

of each neuron (6 outputs) are used as inputs of the next layer (the output layer), the same process is repeated and the final result is the output of the network that indicates whether or not the handwritten signature belong to the group.

The RTL design of the circuit is shown in the figure 5.

Fig. 5 The RTL description of the network

The figure 6 summarize the characteristics of the designed circuit including resource use and performance, we can see that our design occupied 39% of the chip and used only 5% of the embedded Multiplier, this results shows the advantages of using FPGA in the implementation of ANN.

| Flow Summary | |
|---|---|
| Flow Status | Successful - Fri Jan 03 20:00:32 2014 |
| Quartus II 32-bit Version | 11.1 Build 216 11/23/2011 SP 1 SJ Web Edition |
| Revision Name | r1052 |
| Top-level Entity Name | r1052 |
| Family | Cyclone II |
| Device | EP2C70F896C6 |
| Timing Models | Final |
| Total logic elements | 26,835 / 68,416 ( 39 % ) |
| Total combinational functions | 25,327 / 68,416 ( 37 % ) |
| Dedicated logic registers | 4,271 / 68,416 ( 6 % ) |
| Total registers | 4271 |
| Total pins | 322 / 622 ( 52 % ) |
| Total virtual pins | 0 |
| Total memory bits | 0 / 1,152,000 ( 0 % ) |
| Embedded Multiplier 9-bit elements | 14 / 300 ( 5 % ) |
| Total PLLs | 0 / 4 ( 0 % ) |

Fig. 6 The results of the designed circuit

To validate our design we made a simulation by using SIMULINK, considering the processing of the neural network. We feed the network with two vectors, one of a signature that belong to the trained set and on that not, the simulation results of the overall network behavior are shown in figure 7.

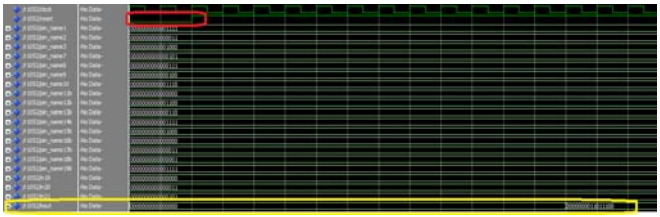

Fig. 7 The simulation

## 4. Conclusions

This paper has presented a design solution of neural networks by FPGAs. We implemented a single neuron and proposed a solution for connecting neurons into a multilayer feedforward BP neural network.The proposed network architecture is modular, being possible to easily increase or decrease the number of neurons as well as layers.

An important obstacle in this work was the hardware implementation for the approximation of sigmoid activation function.

The aim of this work is to be able to use the weights calculated in a software environment (MATLAB) directly in a hardware solution implemented in FPGA.

The result of implementation shows that we still have enough resources for implementing other circuit like the image processing algorithm used to extract features from the handwritten images.

## References

[1] R. Omondi, C. Rajapakse," FPGA Implementation of Neural Networks", Springer U.S., 2006,ISBN 10-0-387-28485-0.

[2] M.Gopal, Digital Control and Static Variable Methods. Tata McGraw Hill,New Delhi, 1997.

[3] V. H. C. de Albuquerque, A. R. de Alexandria, P. C. Cortez, and J. M. R. S. Tavares, Evaluation of multilayer perceptron and self-organizing map neural network topologies applied on microstructure segmentation from metallographic images,NDT & E International, vol. 42, no. 7, pp. 644-651, Oct. 2009.

[4] R.Hunt, A Guide to MATLAB for Beginners and Experienced Users. L.Lipsman, M.Rosenber, Cambridge University Press, ISBN-I3 978-0- 511-07792-0, 2001.

[5] http://www.altera.com

[6] Sathish Kumar, Neural Networks: A Classroom Approach. Tata McGraw-Hill Publishing Company Limited, New Delhi, 2004.

[7] Alin Tisan, Stefen Oniga, Daniel MIC, Attila Buchman, “Digital Implementation Of The Sigmoid Function For FPGA Circuits”, ACTA Technica NAPOCENSIS Electronics and Telecommunication, vol.50, no.2, 2009.