# A NEW METHOD FOR FPGA IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK USED IN SMART DEVICES

*Stefan Oniga*
Lecturer Eng
*North University of Baia Mare, Romania*

Smart devices development with learning capabilities and adaptive behavior is a need of these days. The implementation of such devices is possible using artificial neural networks (ANN). The present work shows a new, efficient and rapid method to design, train and implement in FPGA neural networks. System Generator tool for Simulink is used for ANN design using neural networks specific blocks, created by author. The Matlab is used to perform the off-chip learning task. System Generator also allows the easy generation of hardware Description Language (HDL) code from the system represented in Simulink. The VHDL design can then be synthesized for implementation in the Xilinx family of FPGA devices.

Keywords: smart, neural network, adaptive, learning, FPGA, VHDL

## INTRODUCTION

Nowadays the development of intelligent and more natural smart devices, without need of knowledge for parameters setting activity, is attracting the interest of many research groups worldwide. The need to have learning capacity and adaptive behavior for such smart devices can be satisfied using neural networks and FPGA implementation is an easy an attractive way for hardware implementation

Among possible applications are intelligent computer peripherals enabling people with any kind of handicap to use computer and communicate, as any kind of industrial or domestic device with learning and adaptive capabilities.

The goal of this work was to develop hardware-software codesign platform enabling the fast development of smart interfaces using:

- Application specific sensors,
- hardware modules that can be easily connected
- VHDL modules that can manage sensors,
- Artificial Neural Networks (ANN) used for example for features extraction, pattern recognition, etc.

Using this framework development of new smart devices needs only design and synthesis of new VHDL drivers for the new sensors and new application-specific ANNs.

## THE METHOD

The integrated hardware-software environment represents a new framework for hardware implementation of the Artificial Neural Networks and could be used for example for:

1. Development of ANN specific blocks in Simulink Xilinx blockset

2. Easy implementation in hardware of many types of ANNs
3. Development of new smart devices using Field Programmable Gate Arrays

## NEURAL NETWORK DESIGN

Neural networks could be realized using the specific modules created by author with blocks from Xilinx Blockset library of the System Generator tool for Simulink, presented in Figure 1.

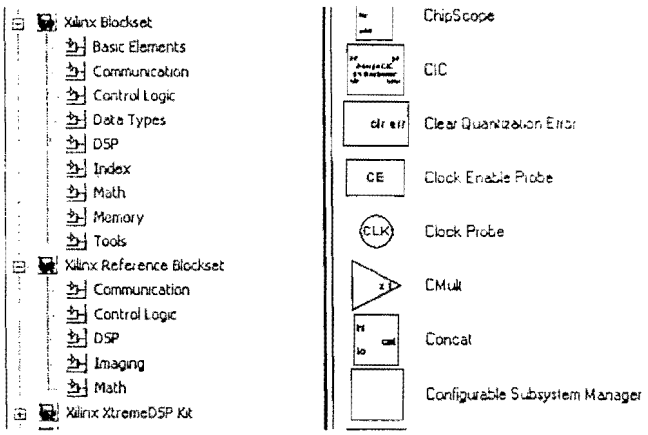


Fig. 1
Xilinx Blockset Libraries

An ANN designed using Xilinx Blockset modules and modules created by author are presented in Figure 2.

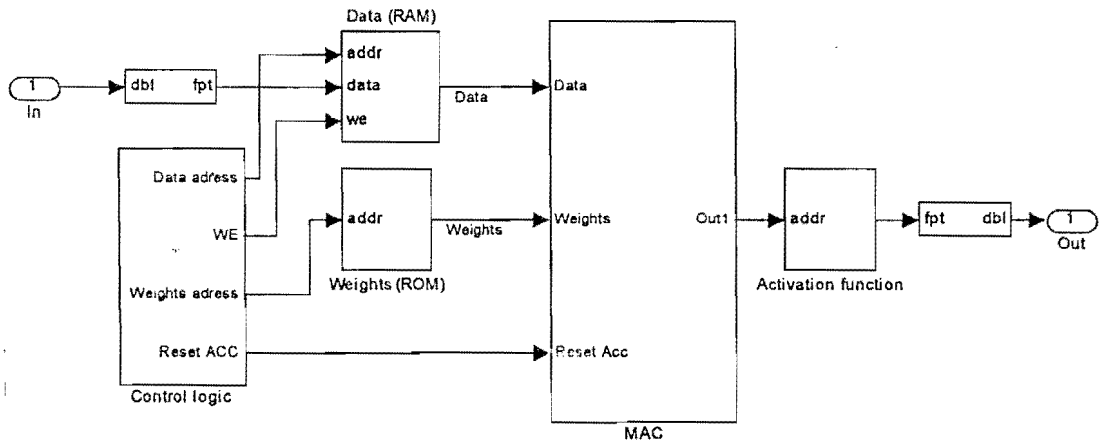


Fig. 2
Neural network model in Simulink

The main elements of neurons are:

1. The multiply-accumulate (MAC) block. This block could be implemented efficiently using existing dedicated multipliers in Virtex II, Virtex II Pro or

Spartan III FPGAs. For example XC2V1000 (a Virtex II FPGA) has 40 dedicated 18 bits MAC blocks. They can be implemented efficiently even in other FPGAs without dedicated MAC blocks, using Xilinx LogiCORE GeneratorBloc. Figure 3 presents a MAC block realized using blocks from Xilinx Blockset library.

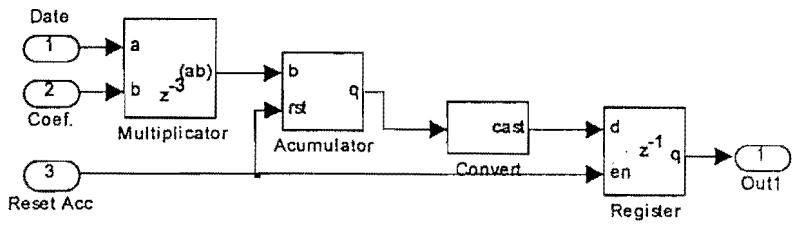


Fig. 3
Multiply-accumulate block

Resources used by the 16 bits multiply-accumulate block

Table 1

| Used resources | MAC implemented with | |
|---|---|---|
| | VIRTEX-II dedicated multipliers | Xilinx LogiCORE multipliers |
| Slices | 57 (29) | 91 (63) |
| Flip Flops | 59 (39) | 126 (106) |
| Block RAMs | 0 | 0 |
| Look-up tables | 66 (17) | 170 (121) |
| Dedicated multipliers | 1 (1) | 0 |
| % from a 50.000 gates Spartan-II | -- | 11,58 % |
| % from a 250.000 gates Virtex-II | 3.58 % | 5,79 |
| % from a 1.000.000 gates Virtex-II | 1,07% | 1,73% |

Between parentheses are shown resources used by the 16 bits multiply block.

2. Control logic block determines neural network architecture. For example determines number of neurons and the correspondence between inputs and weights. For simplicity we have considered that all neurons from a layer are connected to all neurons outputs from previous layer. In other cases the not necessary connections could be deactivated setting corresponding weights to zero.
3. ROM memory is used for storage of neurons inputs weights, and the RAM memory as a data buffer.
4. Transfer function is implemented using look-up tables.

The resources consumed by a very simple network with one layer of 7 neurons that uses one MAC bloc (implementing a layers parallelism) are presented in Table 2. Between parenthesis are shown resources used by the 16 bits multiply-accumulate block.

The hardware requirements are lowest here so this network can be implemented in a smaller FPGA. The multiply-accumulate operation is the bottleneck of ANNs FPGA implementation, because require a large amount of logic blocks.

The resources depend in a grate measure on the number of bits used to represent data and weights. The shown data are for 8 bits representation of data and 12 bits used for weights.

Resources consumed by a simple network (one MAC per layer)

Table 2

| Used resources | MAC implemented with | |
|---|---|---|
| | VIRTEX-II dedicated multipliers | Xilinx LogiCORE multipliers |
| Slices | 80 (57) | 114 (91) |
| Flip Flops | 77 (59) | 144 (126) |
| Block RAMs | 3 (0) | 3 (0) |
| Look-up tables | 103 (66) | 207 (170) |
| Dedicated multipliers | 1 (1) | 0 |
| % from a 50.000 gates Spartan-II | -- | 14,84% |
| % from a 250.000 gates Virtex-II | 5,20% | 7,42% |
| % from a 1.000.000 gates Virtex-II | 1,56% | 2,22% |

The same network implemented using one MAC block per neuron (consequently implementing node parallelism) uses more resources, but is much time faster.

Resources consumed by a simple network (one MAC per neuron)

Table 3

| Used resources | MAC implemented with VIRTEX-II dedicated multipliers |
|---|---|
| Slices | 534 |
| Block RAMs | 1 |
| Dedicated multipliers | 7 |
| % from a 1.000.000 gates Virtex-II | 10,43% |

Definition of system elements are made automatically using variables that are taken from Matlab workspace. In this way dimension of the memories, registers, counters, as constants and number of bits/word are automatically loaded in Simulink representation of the ANN after the simulation of the neural network in Matlab.

The System Generator tool for Simulink developed by Xilinx Inc. allow the easy generation of hardware Description Language (HDL) code from a system representation in Simulink. This VHDL design can then be synthesized for implementation in the Xilinx family of FPGA.

## RESULTS

The chosen application for testing the method was static hand gesture recognition system using:

- data glove equipped with optical fiber flex sensors, as fingers and hand position date source
- force sensing resistors, as data source related to contact force between hand and an object
- ADUC512 microconverter as core of the data acquisition system
- FPGA platform used for ANN implementation. It receives data in 8 bit parallel format and drive the 7 segment decoder that displays the recognized gesture number.

Figure 4 presents the implemented configuration for gesture recognition.

First block is a parallel port implementation and ensure the correct data transfer between data acquisition system and gesture recognition neural network.

RNA1 is Feed-Forward network that can be trained in many different ways but one of the most common methods is gradient based learning using back propagation. Other very used training method is Hebbian learning rule. We have tested both of them with good results. RNA 1 is used for input data preprocessing and is build from one layer of N1 neurons, where N1 represent the number of sensorial inputs.

The second network used for classification task is a simple competitive network with one layer of N2 neurons, one for each of N2 gesture to be recognized.

Last block is a BCD to 7 segment decoder and it displays the recognized gesture number.

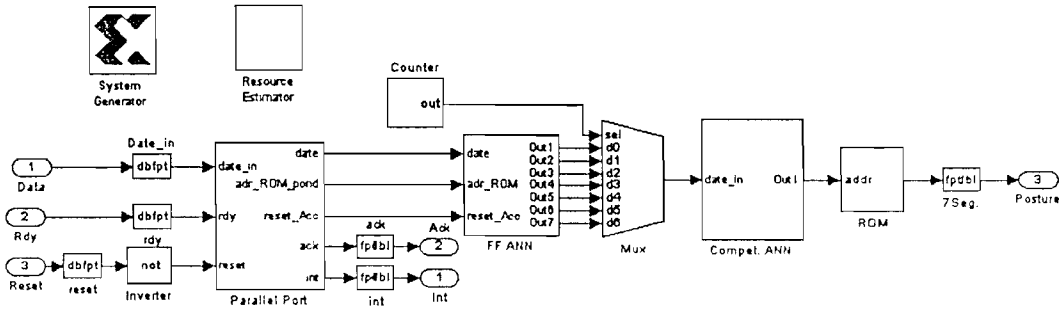


Fig.4
Gesture recognition network

Resources consumed by the gesture recognition system

Table 4

| Used resources | MAC implemented with VIRTEX-II dedicated multipliers |
|---|---|
| Slices | 1781 |
| Block RAMs | 2 |
| Dedicated multipliers | 22 |
| % from a 1.000.000 gates Virtex-II | 34% |

## CONCLUSION

The developed and tested methode contributes with folowing aspects:

- Creation of the framework that permit rapid development of smart devices with learning capability and adaptive behavior
- Development of the ANN implementation method using Xilinx Blockset modules
- Creation of neural network specific modules such as MAC units, activation functions etc.
- The proposed method permit to easily adapt the number of neurons per layer, the weight of each input and the activation function.
- A testbench was developed for application and it permit to implement different types of neural network with different kinds of architecture.
- The hardware – software enviroment developed in Matlab
- Different types of neural network with different kinds of architecture were tested such us:
  - Feed-Forward network trained in different ways such as gradient based learning using back propagation, Hebbian learning rule, etc.
  - Simple competitive network

Main applications for such smart devices with embedded intelligence are in the prosthetic, automotive, "domotic" and automation field where the trend is to produce easy-to-use devices.

REFERENCES

[1] Jihan Zhu, Peter Sutton, **FPGA Implementations of Neural Networks - a Survey of a Decade of Progress**, 2003.

[2] Ştefan Oniga, Virgil Tiponuţ, Attila Buchman, Daniel Mic, **Adaptive Interface Based on FGA implemented Artificial Neural Network**, Scientific Bulletin of the Politehnica University of Timişoara, Tomul 49(63), Fascicola 1, 22-23 octombrie 2004, pag.236.

[3] Dr. M. Turhan Taner, **Kohonen's Self Organizing Networks With Conscience**, Rock Solid Images, November 1997.

[4] Ştefan Oniga, Attila Buchman, **A New Method For Hardware Implementation Of Artificial Neural Network Used In Smart Sensors** The 10th International Symposium for Design and Technology for Electronic Modules - SIITME '04, Bucharest, Romania, 21-24 Sept. 2004, pp. 135-139.

[5] Rolf F. Molz, Paulo M. Engel, Fernando G. Moraes, Lionel Torres, Michel Robert, **Codesign of Fully Parallel Neural Network for a Classification Problem**, International Conference on Information Systems, Analysis and Synthesis, Orlando, USA, 2000.

[6] R. Gadea, J. Cerda, F. Ballester, A. Mocholi, **Artificial neural network implementation on a single FPGA of a pipelined on-line backpropagation,** Proceedings of the 13th International Symposium on System Synthesis (ISSS'00), pp 225-230, Madrid, Spain, 2000.