ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2012. No. 4(120)
ELEKTRONIKA IR ELEKTROTECHNIKA

T 170 ELECTRONICS
ELEKTRONIKA

Design and Implementation of Neural Networks Neurons with *RadBas*, *LogSig*, and *TanSig* Activation Functions on FPGA

I. Sahin
Department of Electronics and Computer Education, Faculty of Technical Education, Duzce University,
Konuralp Yerleşkesi, 81620 Duzce, Turkey, phone: +90 0 380 5421133, e-mail: ibrahimsahin@duzce.edu.tr

I. Koyuncu
Department of Control and Automation, Duzce Vocational High School, Duzce University,
Uzun Mustafa Mh, 81010 Duzce, Turkey, phone: +90 0 380 5240099, e-mail: ismailkoyuncu@duzce.edu.tr

http://dx.doi.org/10.5755/j01.eee.120.4.1452

Introduction

Artificial Neural Networks (ANNs) are the computational models which copy the human brain's learning and decision making abilities. Before they are used, they are gone through a learning process and later are utilized in decision making processes. Their decision making ability and correctness of the decisions depend on several key factors such as the structure of network, type of activation functions in the neurons, period of learning, and number of stimuluses used during the learning process [1].

ANNs are utilized in several areas such as controlling electric motors [2], signal and image processing [3], prediction [4], classification [5], etc. ANNs are usually implemented as software tools. But in some cases when software implementations do not give desired performance, hardware implementations are utilized. Hardware implementations of the ANNs can be classified in three groups which are Application Specific Integrated Circuits (ASIC) [6], Digital Signal Processing (DSP) [7], and Field Programmable Gate Arrays (FPGA) [8] implementations. Although, ASIC implementations have a performance advantage over the other hardware choices, they are of a major drawback. Once an ANN is implemented as an ASIC, it is fixed, and the network configuration cannot be modified. Since the DSP chips are a kind of processors, they perform given task in a serial fashion. Calculations in ANNs can be performed in parallel, but this parallelism cannot be exploited in DSP. FPGAs offer solutions to both ASIC's being fixed problem and DSP's serial processing problem [8].

In this study, we designed and implemented a total of 18 different neurons, 2, 4 and 6-input biased and non-biased with each having three different activation functions. The activation functions we selected are *Radial-Basis* (*RadBas*), *Logarithmic-Sigmoid* (*LogSig*), and *Tangent-Sigmoid* (*TanSig*). The common feature of these functions is that they require the calculation of e^x.

The motivation behind this study is to show the possibility of implementing neural networks with exponential activation functions on current FPGA chips and measure the performance of the neurons. The results showed that up to 10 neurons can fit in the smallest *Virtex-6* chip and the network can be clocked up to 405MHz.

The rest of the paper is organized as follows. Second section briefly introduces the FPGA chips, ANNs, and some related studies. Structure of the designed neurons and details of the activation functions are given in section three. Section four presents the implementation result. Finally, the paper ends with a conclusions section.

Background

A. FPGA Chips. FPGAs are a type of chips that are completely prefabricated and ready for customization. The users of these chips can implement digital circuit designs by uploading the configuration file. Since the configuration time of these chips is very small, circuit designs can be realized very quickly compared to Application Specific Integrated Circuits (ASIC) implementations. A typical FPGA device contains three configurable parts. These parts are an array of logic cells called Configurable Logic Blocks (CLBs), a programmable interconnection network, and programmable Input/Output Blocks (IOBs). The CLBs are the most important parts of the FPGA device. They include Lookup Tables (LUTs), programmable flip-flops and several programmable multiplexers. The LUTs are function generators, capable of implementing combinational logic functions [9].

B. Artificial Neurons and Neural Networks. A neuron of an ANN is composed of three sections. The first section contains a number of synapses each having its own weight or strength and the second section contains adder. The last section is the squashing or activation function section. The input signals to the neuron's synapses are first multiplied by the related synapse's weights. Later, all multiplied (weighted) inputs are summed together and a single value is produced in the second section. If available, the bias is also added to this value. In the final section, the result of the adders is passed through an activation function. The purpose of the activation function is to limit the amplitude range of the output signal to some finite value. Several different types of activation functions have been defined in the literature. In the proposed work, we focused on *RadBas*, *LogSig*, and *TanSig* activation functions.

C. Related Works. Several research studies have been conducted to implement ANNs on FPGAs. Here, some of the selected studies are presented briefly.

Himavathi et. al. proposed an new realization technique of the ANNs on FPGA. Instead of realizing the whole network on FPGA, they only implemented the largest layer of the network and reused it for the other layers with the help of a controller block. They claimed that their technique is very effective in reducing the hardware cost of the ANNs with a moderate overhead on the speed [8].

Gompert et al. presented the development and implementation of a parameterized FPGA-based architecture for feed-forward multilayer perceptron with backpropagation learning algorithm. They also presented a new method for calculation of the sigmoid function. They are able to reduce the size of the look-up table used in the calculation of sigmoid function by applying a linear interpolation technique [10].

Overall design of the neurons

In this study, a total of 18 neural cells were designed and implemented on Xilinx's *Virtex-6* FPGA chip. In this section, first, the general structure of the neurons and multiplication and addition sections are explained. Finally, logic designs for the activation functions of *RadBas*, *LogSig*, and *TanSig* are given.

A. General Structure of the Neurons. Generalized top level block diagram of the neurons is shown in Fig. 1. Each neuron has a *Weight/Bias* input, a number of *Data Inputs*, and some synchronization inputs. Since the weight registers are serially connected in the neurons, there is only one *Weight/Bias* input defined in each neuron. It is assumed that the weights and bias values (if available) are feed in to the neurons only once while the neuron is being initialized. The *Shift* input is used for initializing these registers. *InputReady* signal is used to fire the neuron when a valid set of data is available at the inputs of the neuron. Neurons have only two outputs. These are *FOut* which is the neuron's functions output and *ResultReady* output which is asserted when a valid result is available at *FOut*. *InputReady* and *ResultRead* signals are designed to let more neurons to be connected together to form a neural network. In such a case, each neuron can automatically activate the neurons coming after it.

A total of six different types of neurons were designed. These types are 2, 4, and 6-input biased or non-biased neurons. The neurons can accept one set of input data at each clock cycle. The input data is first multiplied by the weights and the multiplication results are summed and forwarded to the activation function units. By changing the activation functions in each 2, 4, and 6-input biased or non-biased neurons a total of 18 different neurons were designed.

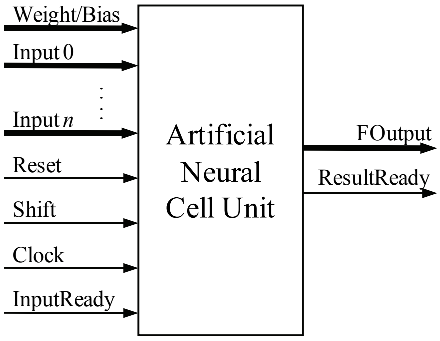

Fig. 1. Common top level block diagram of the neural cells

Fig. 2 shows the block diagram of the 6-input non-biased neurons. Since the multipliers and adders used in these designs are fully pipelined, to balance the data flow in the neurons, a delay unit is used and it is ensured that data at the outputs of the adders 1 and 2 reaches at the same time to the adder 3. In the biased version of the 6-input neuron due to the bias, another adder is used instead of the delay unit. 2 and 4-input neurons were designed similar to the 6-input neurons.

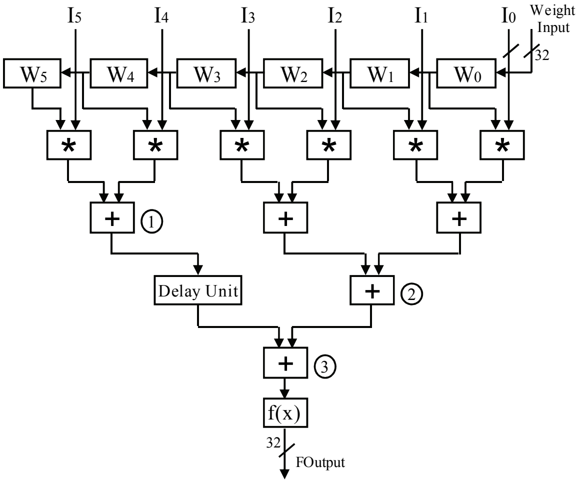

Fig. 2. Block diagram of the 6-input non-biased neuron

All multipliers and adders used in the neurons were designed as pipelined units using Xilinx IP Core Generator and their latencies are 8 and 12, respectively. They were designed to process standard 32-bit single precision floating-point data.

B. Design and Implementation of the Activation Functions. Through the course of this study, logic designs for three different types of activation functions which are *RadBas*, *LogSig*, and *TanSig* have been done. The common feature of the activation functions is that all of them require the calculation of e^x value. In literature, several approaches were used to calculate e^x [11] such as using look-up tables. In the proposed work, we designed a

COordinate Rotation DIgital Computer (CORDIC) based exponent calculator. Equation (1) shows the relationship between the trigonometric and exponent functions. CORDIC is able to calculate hyperbolic sine and hyperbolic cosine functions between –π/4 and π/4. We added the Sinh(x) and Cosh(x) output of the CORDIC using fixed-point adder

$$e^x = Sinh(x) + Cosh(x)\,. \tag{1}$$

RadBas, *LogSig*, and *TanSig* activation functions are given in Equations (2), (3), and (4), respectively. Using our CORDIC-based exponent calculator, we designed these activation functions:

$$f(x) = e^{-(x^2)}\,, \tag{2}$$

$$f(x) = 1/(1+e^{-x})\,, \tag{3}$$

$$f(x) = 2/(1+e^{-2x})-1\,. \tag{4}$$

Fig. 3, 4, and 5 shows the logic designs for the activation functions. Each design accepts and produces 32-bit floating-point data. Since the CORDIC works on fixed-point numbers, the data is converted between floating-point and fixed-point before and after CORDIC in calculation flow. 32-bit floating point number has up to 8 digit precision. Due to the conversions between number systems in the designs, the results' correctness are degraded up to 6 digits. This much precision is acceptable in many neural network applications.

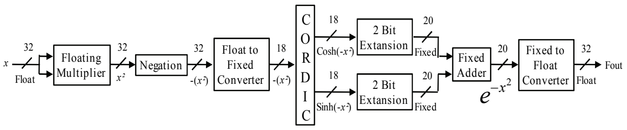

Fig. 3. Block diagram of *RadBas* activation function

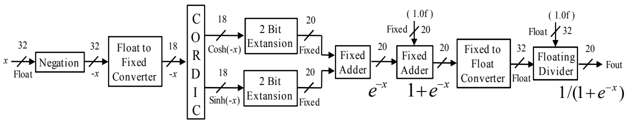

Fig. 4. Block diagram of *LogSig* activation function

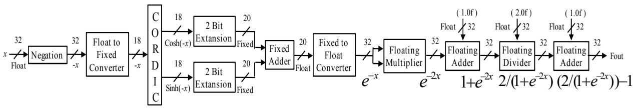

Fig. 5. Block diagram of *TanSig* activation function

Implementation results

The activation functions were designed and coded in VHDL and mapped to *Virtex-6* chip (xc6vlx75t, speed grade -3) using Xilinx's ISE WebPack 12.1 EDA tool. Table 1 shows the mapping results in terms of clock period and latency. All *RadBas* and *TanSig* units have the same clock period. This is due to the fact that we used standard pipelined units designed by the Xilinx's IP Core Generator. The *LogSig* units have higher clock periods because of cascaded adders which are placed after the 2-bit extension units. The latency of the each unit depends on the number of elements used in the pipelined calculation flow. To reach higher clock rates in the calculations flows, we used highest latency design units.

Table 1. Timing statistics of the neurons

Function Type	Input Count	Bias	Clock Period (ns)	Latency
Radbas	2	No	2.469	62
Radbas	2	Yes	2.469	74
Radbas	4	No	2.469	74
Radbas	4	Yes	2.469	86
Radbas	6	No	2.469	86
Radbas	6	Yes	2.469	86
Tansig	2	No	2.469	114
Tansig	2	Yes	2.469	126
Tansig	4	No	2.469	126
Tansig	4	Yes	2.469	138
Tansig	6	No	2.469	138
Tansig	6	Yes	2.469	138
Logsig	2	No	3.263	82
Logsig	2	Yes	3.263	94
Logsig	4	No	3.263	94
Logsig	4	Yes	3.263	106
Logsig	6	No	3.263	106
Logsig	6	Yes	3.263	106

Response time is the interval between a set of stimulus is received by a neuron and a result is produced by the same neuron. Fig. 6 shows the response time of each neuron in nanoseconds. These response times imply that the neurons can process from 2.89 to 6.53 million sets of stimulus per second.

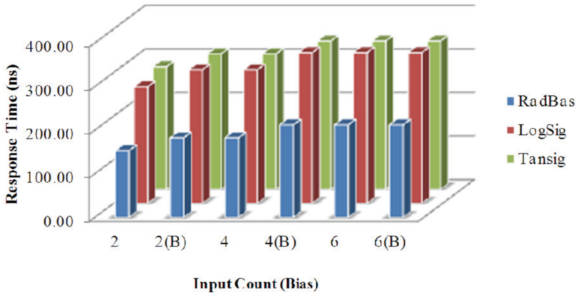

Fig. 6. Response time of each neuron

Table 2 lists the hardware requirements of the neurons on *Virtex-6* chip.

Table 2. FPGA chip statistics of the neurons

Function Type	Inp. Cnt.	Bias	Slice Regs. (%)	LUTs (%)	Occupied Slices (%)	Bounded IOBs (%)	Neurons to Fit in *Virtex-6*
Radbas	2	No	4	8	10	45	10
Radbas	2	Yes	5	9	11	45	9
Radbas	4	No	7	13	15	63	6
Radbas	4	Yes	8	14	16	63	6
Radbas	6	No	10	18	21	81	4
Radbas	6	Yes	10	19	22	81	4
Tansig	2	No	7	12	14	45	7
Tansig	2	Yes	8	13	16	45	6
Tansig	4	No	10	17	20	63	5
Tansig	4	Yes	10	18	20	63	5
Tansig	6	No	12	22	25	81	4
Tansig	6	Yes	13	23	29	81	3
Logsig	2	No	5	9	10	45	10
Logsig	2	Yes	6	10	12	45	8
Logsig	4	No	8	14	16	63	6
Logsig	4	Yes	8	15	17	63	5
Logsig	6	No	11	19	22	81	4
Logsig	6	Yes	11	20	22	81	4

When the IOB utilizations of the units are considered, it seems that only one copy of each neuron can fit in to a *Virtex-6*. However, while forming a neural network, since the neurons share the same inputs, the IOB requirement does not increase linearly as the number of neurons increases in the network. So, we only considered the number of Occupied Slices when deciding neurons counts that can fit in to an FPGA chip. The last column of the table lists the number of neurons that can fit in to *Virtex-6*. It is observed that as the number of inputs of the neurons increase, neurons' hardware requirements are also increase because of additional multipliers and adders in the first part of the neurons. Using these neuron designs, a neural network containing up to 10 neurons of the same kind can be formed in xc6vlx75t chip. Using a larger chip such as xc6vhx565t larger networks can be easily formed.

Conclusions

In this research work, we designed and implemented a total of 18 different FPGA-based neurons, 2, 4 and 6-input biased and non-biased with each having three different activation functions. The implementation results showed that ANNs containing 10 or more neurons can easily be constructed on FPGAs and these ANNs can be clocked up to 405 MHz. The activation function section of the neurons requires the calculation of e^x. We utilized Xilinx's CORDIC design which is able to calculate e^x between -π/4 and π/4. As a result the neurons have to be used with normalized values. In the future, to eliminate this limitation, a new exponent calculator can be designed and integrated to the neurons.

References

1. **Nirmaladevi M., Mohankumar N., Arumugam S.** Modeling and Analysis of Neuro–Genetic Hybrid System on FPGA // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 8(96). – P. 69–74.
2. **Reyneri L. M.** Implementation Issues of Neuro–Fuzzy Hardware: Going Toward HW/SW Codesign // IEEE Transactions on Neural Networks, 2003. –Vol. 14. – No. 1.
3. **Paukštaitis V., Dosinas A.** Pulsed Neural Networks for Image Processing // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 7(95). – P. 15–20.
4. **Rutka G.** Prediction Accuracy of Neural Network Models // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 3(83). – P. 29–32.
5. **Raudonis V., Narvydas G., Simutis R.** A Classification of Flash Evoked Potentials based on Artificial Neural Network // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 1(81). – P. 31–36.
6. **Juang J. G., Chien L. H.,Lin F.** Automatic Landing Control System Design Using Adaptive Neural Network and Its Hardware Realization // IEEE System Journal, 2011. – Vol. 5. – No. 2.
7. **Yonggang W., Junwei D., Zhonghui Z., Yang Y., Lijun Z., Bruyndonckx P.** FPGA Based Electronics for PET Detector Modules With Neural Network Position Estimators // IEEE Trans. on Nuclear Science, 2011. – Vol. 58. – No. 1.
8. **Himavathi S., Anitha D., Muthuramalingam A.** Feedforward Neural Network Implementation in FPGA Using Layer Multiplexing for Effective Resource Utilization // IEEE Trans. on Neural Networks, 2007. –Vol. 18. – No. 3.
9. **Sahin I.** A 32–bit floating–point module design for 3D graphic transformations. – 2010. – Vol. 5(20). – 3070 p.
10. **Gomperts A., Ukil A., Zurfluh F.** Development and Implementation of Parameterized FPGA–Based General Purpose Neural Networks for Online Applications // IEEE Trans. on Industrial Informatics, 2011. – Vol. 7. – No. 1.
11. **Jamro E., Wiatr K., Wielgosz M.** FPGA Implementation of 64–Bit Exponantial Function for HPC // International Conference on FPL and Applications. – 2007. – P. 718–721.

Received 2011 08 15
Accepted after revision 2011 10 11

I. Sahin, I. Koyuncu. Design and Implementation of Neural Networks Neurons with *RadBas*, *LogSig*, and *TanSig* Activation Functions on FPGA // Electronics and Electrical Engineering. – Kaunas: Technologija, 2012. – No. 4(120). – P. 51–54.

Artificial Neural Networks (ANNs) are utilized in several key areas such as prediction, classification, motor control, etc. When high performance is needed, FPGA realizations of the ANNs are preferred. In this study, we designed and implemented a total of 18 different FPGA-based neurons, 2, 4 and 6-input biased and non-biased with each having three different activation functions requiring the calculations of e^x. Our purpose was to show the possibility of implementing neural networks with exponential activation functions on current FPGAs and measure the performance of the neurons. The results showed that up to 10 neurons can fit in to the smallest *Virtex-6* and the network can be clocked up to 405MHz. Ill. 6, bibl. 11, tabl. 2 (in English; abstracts in English and Lithuanian).

I. Sahin, I. Koyuncu. Neurioninių tinklų neuronų su *RadBas*, *LogSig* ir *TanSig* aktyvinimo funkcijomis projektavimas ir įdiegimas į FPGA // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2012. – Nr. 4(120). – P. 51–54.

Dirbtiniai neuroniniai tinklai (DNT) naudojami keliose srityse – prognozavimo, klasifikavimo, variklių valdymo ir t. t. Kai reikia didelės greitaveikos, tinkamiausias būdas – įdiegti DNT. Sukurta ir įdiegta 18 skirtingų neuronų FPGA pagrindu, kurių kiekvienas turi tris skirtingas aktyvinimo funkcijas, reikalingas e^x skaičiuoti. Siekta parodyti galimybę įdiegti į FPGA neuroninius tinklus su eksponentinėmis aktyvinimo funkcijomis ir išmatuoti neuronų našumą. Rezultatai rodo, kad iki 10-ies neuronų gali tilpti į mažiausią *Virtex-6* ir tinklas gali veikti iki 405 MHz dažniu. Il. 6, bibl. 11, lent. 2 (anglų kalba; santraukos anglų ir lietuvių k.).