International Journal of Engineering Technology and Scientific Innovation

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

IMPLEMENTING A RECONFIGURABLE NEURAL BASED DEMODULATOR ON AN FPGA

Reyhaneh Pedram, Mohammad Reza Amini

Department of Electrical Engineering, College of Engineering,
Borujerd Branch, Islamic Azad University, Borujerd, Iran

ABSTRACT

In this study a universal demodulator for BPSK, BFSK, ASK signals is presented. This demodulator is based on neural network techniques and can be developed for any kind of modulation scheme. Regardless of modulation type used by transmitters, this demodulator can detect the transmitted signal bits by passing the signal samples through a probabilistic neural network. Each modulation scheme that is best similar and matched to the incoming signal samples will be determined as a detected pattern and its corresponding bit is detected as a received bit. The demodulator has its own architecture techniques to implement via VHDL code for speed and chip area optimization.

Keywords: FPGA; Probabilistic Neural Network; Demodulator; Genetic Algorithm; Block-Pulse Functions

I. INTRODUCTION

Neural networks have been considered recently and have been applied for many purposes such as pattern recognition, signal classification, variable prediction, parameter estimation and etc. [1-3]. Some neural network applications are designed to run on PCs platforms while others are to implement on digital signal processors or FPGA ICs. The Simulating and running of neural based systems on a PC platform are so different from implementing them on an FPGA since there are some limitations such as area of usage and processing speed when designing hardware architecture based on a field programmable gate array. Some studies have been done to demonstrate hardware implementation of neural networks but because of capabilities of FPGAs in parallel processing and performing cumbersome calculations in a minimum possible time, these ICs are beibg considered widely to be employed in systems utilizing artificial intelligence [4-8]. Many works have been done on implementing modulators and demodulators on FPGAs [9, 10] and most of them use CORDIC, lookup tables and Hilbert transform techniques [11-14]. Some

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

of them are also used reconfigurable structures [4, 15-18] since universal demodulator is considered in this paper, a combination of artificial neural networks and likelihood concepts along with implementation techniques should be used. This paper is organized as follows: next section introduces a brief overview of artificial neural network. Section three describes the idea for demodulating signals. The feasibility of design is investigated in section four and section five includes implementing process. Finally, some conclusions are given in section six.

II. ARTIFICIAL NEURAL NETWORK OVERVIEW

Artificial neural networks are a composition of one or more layers connected to each other and each layer consists of some elements called neurons. Each neuron can accept some inputs but outputs just one value. The simplest form of a neuron is shown in Fig.1 Inputs to a neuron usually are divided into ordinary type and bias type. In the Fig.1 which is shown for the neuron, and are ordinary and bias type respectively. Each input except bias type is weighted by its corresponding factor and then added to the bias and other weighted inputs. The result goes through the function called activation function which could be a linear or nonlinear function.

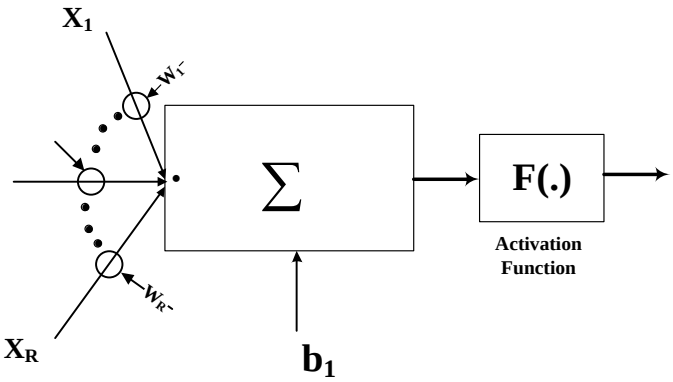


Fig. 1: A simple neuron model

Therefore, the output of a neuron can be in the form of following:

$$output = F((\sum w_i * X_i) + b_1)$$

The above formula is also written in a vector format, in this case there are an input vector and weight vector and a scalar bias [1, 4]. The mentioned simple neuron structure can be extended to a network of neurons connected via input, hidden and output of layers. Fig. 2 shows such a structure.

III. SYSTEM MODEL AND MAIN IDEA

In the wireless digital communication systems, best operated receivers always act based on the principle of similarity called

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

likelihood, which means the receiver should compute the similarity of each incoming signals with the all signals in the signal space of transmitter and chooses the signal that has a maximum similarity, or likelihood as a detected received signal. For example in the case of BFSK, BPSK and BASK which are the main schemes of digital modulations, the corresponding receiver has two paths (two branches) in its structure that each computes the correlation (likelihood) of received signal with one of the signal in the transmitter constellation space. Then the outputs of all paths are compared and the final decision about the detected bit or bits is made. Fig. 3 shows the correlation receiver for BFSK scheme. But the main idea of this paper is to incorporate a neural structure that can be used to demodulate many types of modulation schemes. This structure gives the transmitter the power that it can freely choose and change the modulation type within transmission. Furthermore, the receiver can easily be reconfigured and upgraded for the next generation systems and protocols with least modifications.

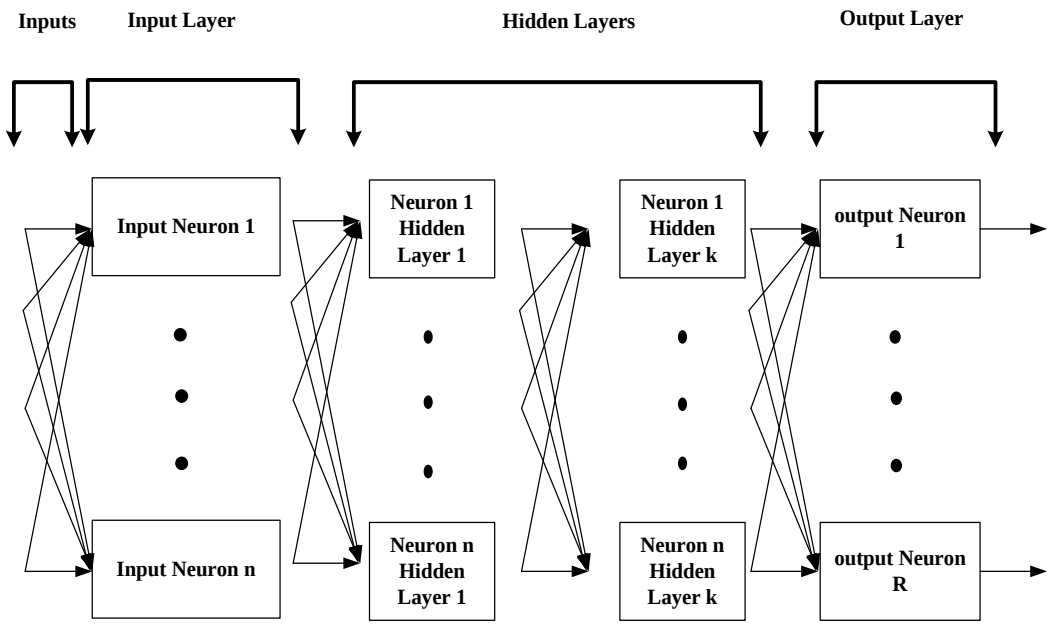


Fig. 2: Neural network structure

To apply the likelihood concept the special kind of neuron called probabilistic neuron is used. Probabilistic neural networks are frequently and efficiently used for signal classification and pattern recognition [19-23]. Fig. 4 shows this structure. In this structure, incoming signals are sampled in a symbol period and then feed the neuron as an input vector (see Fig.5).

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

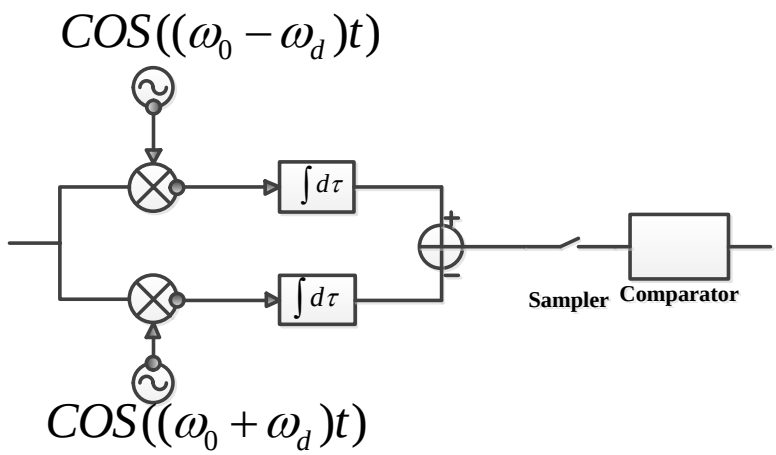


Fig. 3: BFSK demodulator

The neuron computes the distance between the neuron weight vector and input vector and this metric is passed through the activation function, which is a type of radial base function:

$$radial\ base(x) = e^{-x^2}$$

The radial basis function has a maximum of 1 when its argument is 0. As the distance between input vector and weight vector increases, its output decreases. Thus, a radial basis neuron acts as a detector that produces 1 whenever the input is identical to its weight vector. So the probabilistic neuron determines how much the incoming signal samples are similar to the weight vector or equivalently it produces the likelihood probability of the two vectors.

If the weights are set to the samples of special modulation types, say BFSK signal corresponding to bit zero, then the neuron computes the likelihood of incoming signal with that kind of bit-modulation signal. One of advantages of proposed neural demodulator is its reconfigurable structure that enables the receiver to be easily upgraded for the next protocols and standards. The ability that is important in the software defined radios [24, 25] and in the next generations called Adaptive-Intelligence (AI) and also in cognitive radio [26, 27]. Because they should have a flexible structure in order to interact with the environment and to adapt themselves in the case of transmitting power, modulation type, coding rate and so on. A clear example is the IEEE 802.11 standard that does this through adaptive modulation and coding feature (AMC). Furthermore, it can be used in the spectrum sensing part of cognitive radio to detect signals of primary users in the class of feature detection and even more, it can be used as a core unit for demodulating in the OFDM based transmission protocols (the promising protocols of high data rate in the future).

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

IV. FEASIBILITY OF DESIGN

The feasibility of design is tested by simulating the receiver in a noiseless scenario and its performance is shown in the noisy environment. To simulate it, firstly the signal stream of mixed modulation types is generated in the transmitter. To do this random bit stream are generated and each bit is modulated randomly by one of the schemes of BASK, BPSK and BFSK. Fig. 6 shows these processes.

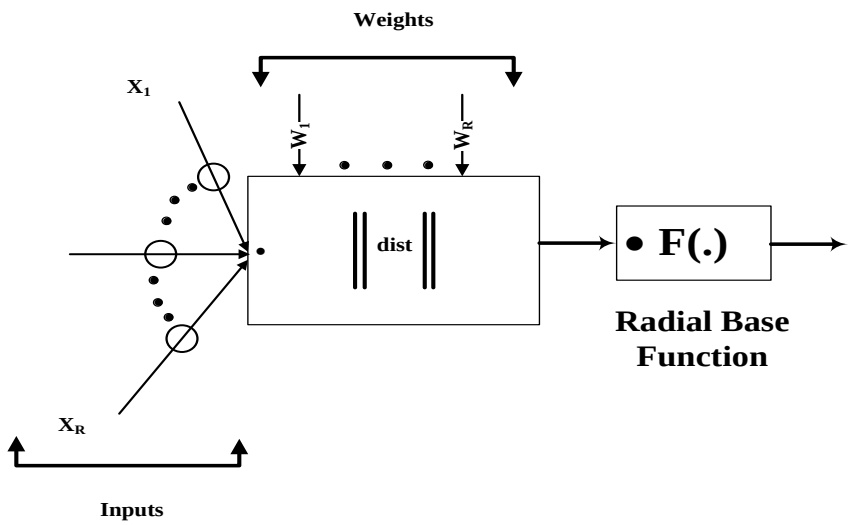


Fig. 4: Probabilistic neuron structure

In the receiver a network of six probabilistic neurons is constructed because there are 3 types of modulation schemes and for each there are 2 kinds of bits 0 and 1. The weights vector of each neuron is set by the values equal to signal samples of each signal type in the signal space of transmitter. Fig. 7 shows the signal samples which are set as each neuron weight vector. To easily determine the performance of receiver each neuron is assigned a class which is a number corresponding to a signal it computes the likelihood. These classes and the other properties of transmitted signals are shown in Table I.

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

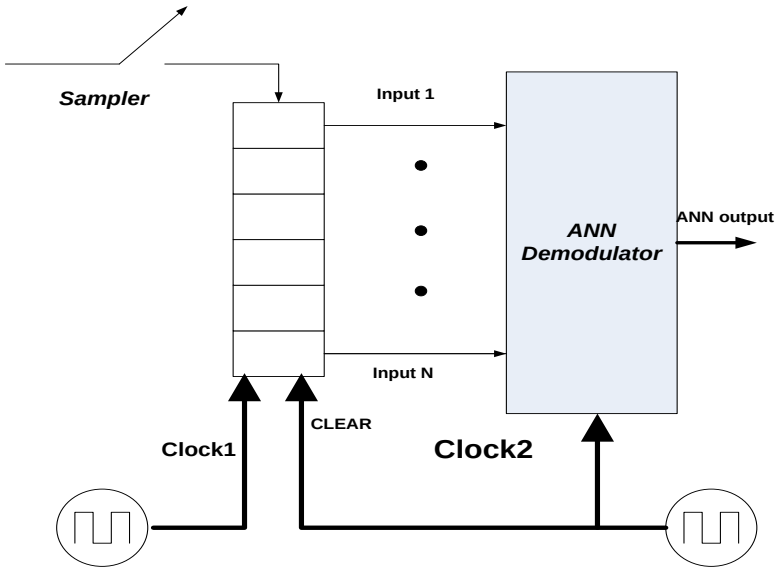
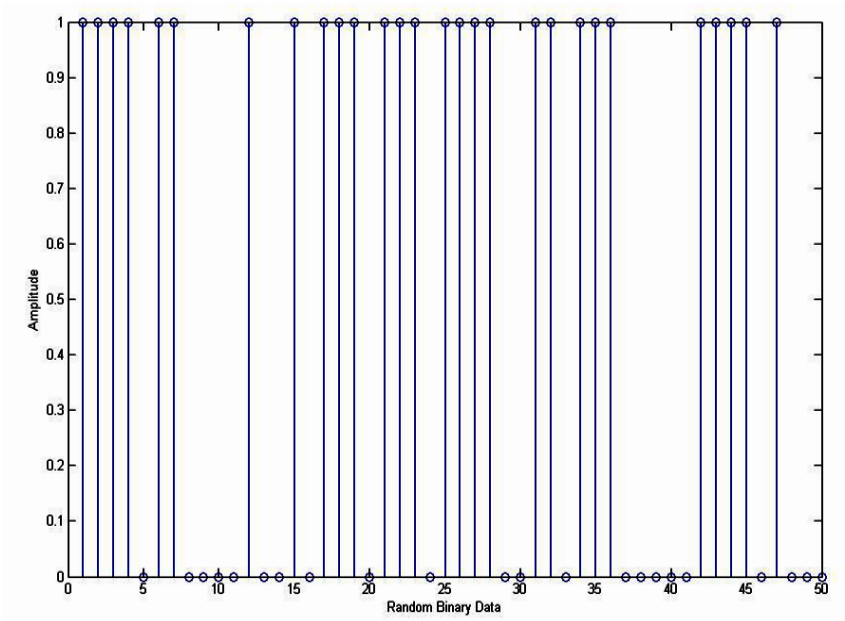


Fig. 5: Samples of signal fed to neural demodulator



ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

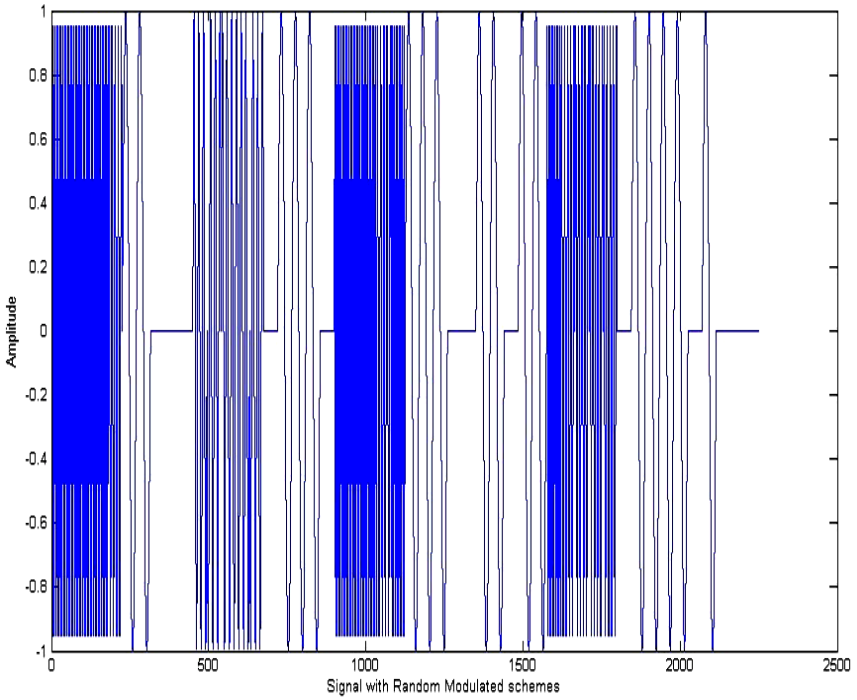


Fig. 6: Generated data bits and randomly modulating them a. random data bits b. randomly modulated bits

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

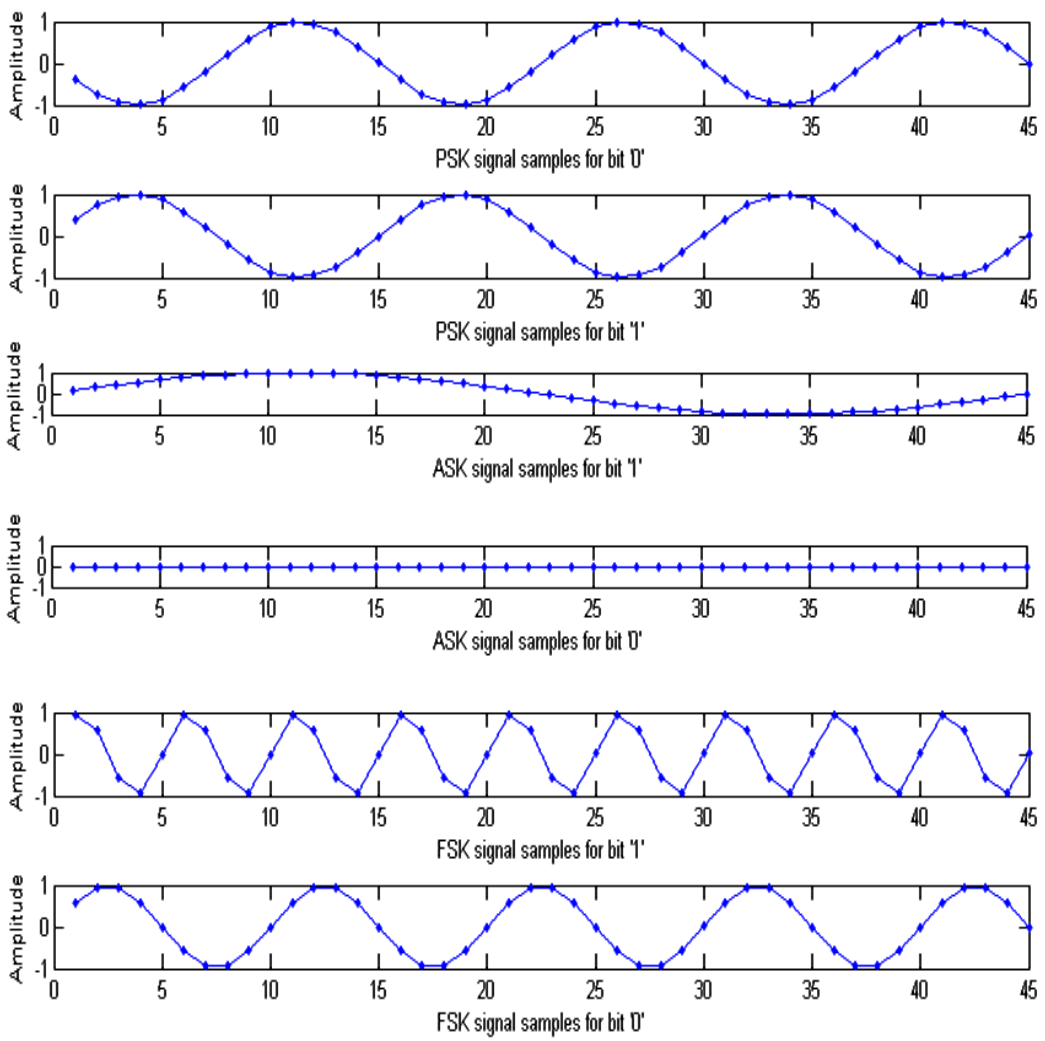


Fig. 7: Samples set as a weights vector

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

Table I: Modulated signals properties

Modulation Type	BFSK	BPSK	BASK
Sampling Frequency (sample/sec)	4500K	4500K	4500K
Carrier Frequency (KHz)	450	300	100
Bit Rate (bit/sec)	90K	90K	90K
Corresponding ANN output class	5 for bit 0	1 for bit 0	3 for bit 0
	6 for bit 1	2 for bit 1	4 for bit 1

In the noiseless environment, the demodulator acts perfectly and detect signal with no error. To simulate the effect of noise, white Gaussian noise is added to signal after modulating at the transmitter. The resulting noisy signal is then passed through the ANN demodulator and the effect of noise power on bit error rate is shown in Fig. 8

Implementing process

V. IMPLEMENTING PROCESS

To easily implement a digital hardware, it is better to divide the whole design into two main sections, data flow unit and control flow unit. The former performs the main digital processing that includes neuron hardware and compare and decide unit. The latter must control the timing process by activating the enable signals of each logic. Fig. 9 shows this architecture for the proposed receiver. The architecture consists of six neuron corresponds to each signal of modulation type. Remember that the architecture can be developed for any type of modulation by adding some extra neurons. The architecture of each neuron is shown in Fig. 10.

One of the challenging part of implementing probabilistic neuron is its activation function that is a nonlinear function. One method to implement a nonlinear function is to approximate it.

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

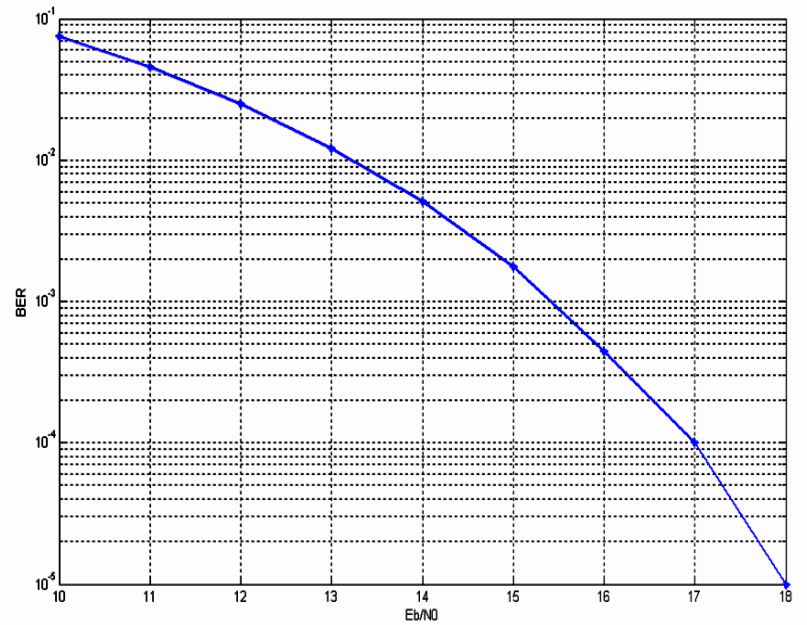


Fig. 8: BER curve versus E_b/N_0

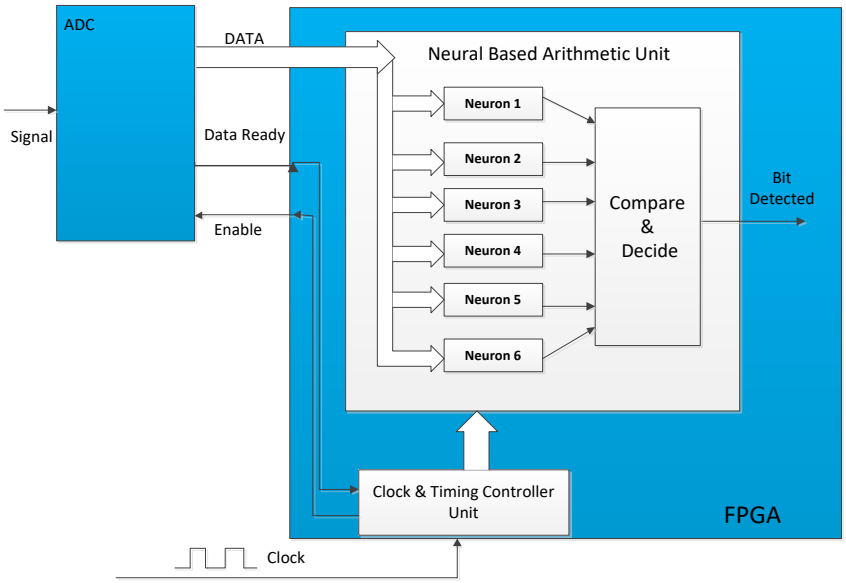


Fig. 9: Neural demodulator architecture

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

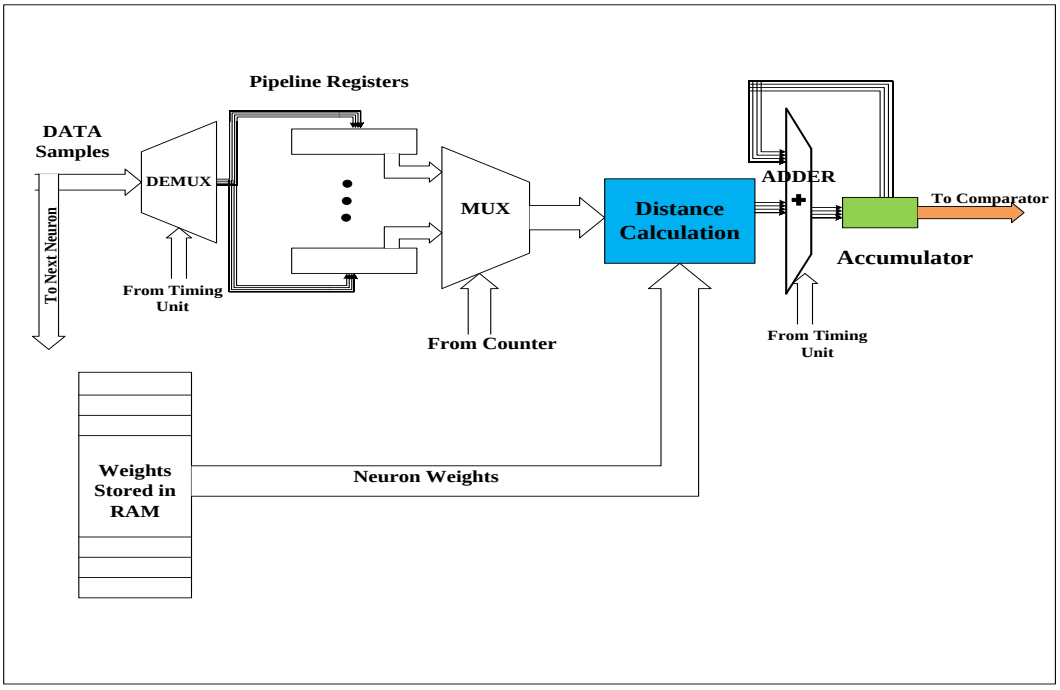


Fig. 10: Neuron architecture

The radial base function can be approximated by taylor series expansion that is shown below:

$$e^{-x^2} = \sum_{0}^{\infty} \frac{(-1)^n (x^{2n})}{n!}$$

In the Fig.11 the radial base function and its approximations up to 5th order are shown in the same axis to compare. To have a good approximation higher orders of taylor series should be employed which means the architecture needs implementing multipliers which in turn results in decreasing speed and increasing chip area usage. So we turn into another way of approximation which is used in calculus of variation called direct method [28]. In this method the radial base function is approximated using by hybrid Block-pulse and Chebyshev polynomials [29].

chevyshev polynomials are in the class of orthogonal sets.

The hybrid Block-pulse and Chebyshev functions $b(n,m,t)$ is defined as [18] :

$$b(n,m,t) = \begin{cases} T_m\left(\frac{2N}{t_f}t - 2n + 1\right), for\ t \in \left[\left(\frac{n-1}{N}\right)t_f, \frac{n}{N}t_f\right) \\ 0 \qquad\qquad otherwise \end{cases}$$

Where n and m are the orders of Block-pulse and Chebyshev functions and $T_m(t)$ are the well-known Chebyshev polynomials which are orthogonal in the interval

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

$[-1,1]$that satisfy the following recursive function:

$$T_{m+1}(t) = 2tT_m(t) - T_{m-1}(t) \;\; m = 1,2,\ldots$$

$$T_0(t) = 1, T_1(t) = t$$

A function $f(t)$ defined over $[0, t_f)$ can be approximated as:

$$f(t) \cong \sum_{n=1}^{N} \sum_{m=0}^{M-1} c(n,m) b(n,m,t)$$

Where

$$c(n,m) = \langle f(t), b(n,m,t) \rangle$$

In which <, > denote the inner product.

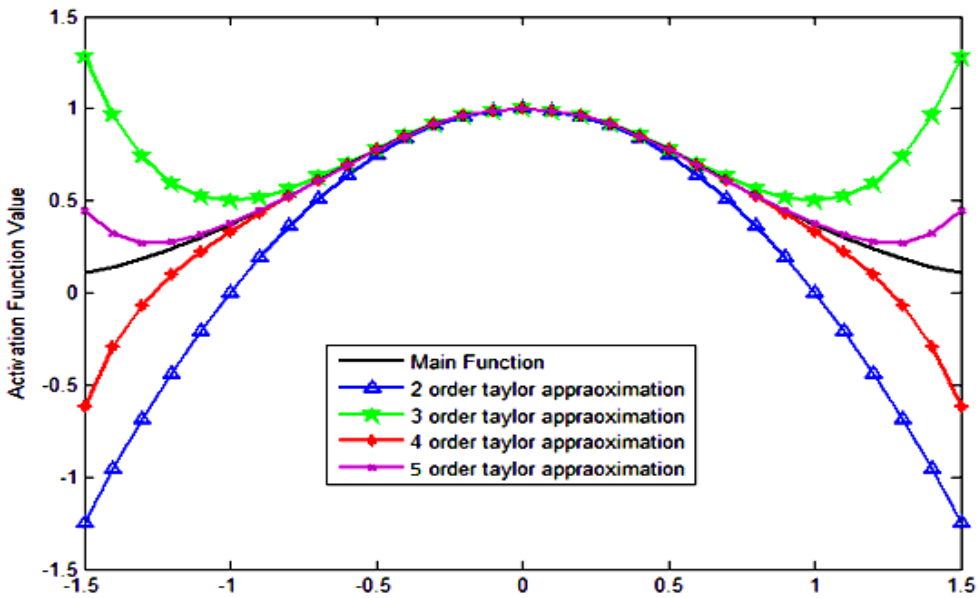


Fig. 11: Radial base function and its approximations by taylor series

Here the radial base function is approximated in the interval [-2, 2] but the subintervals for Block-pulse functions are not predefined. Since the fewer the number of subintervals lead to less amount of chip area usage, it is desired to approximate the function with lower orders of Chebyshev and fewer subintervals of Block-pulse. Because implementing the zero and first order polynomials are easy, just needed to use multiplexers and tri-state gate, we looked for the points x_1 and $x_2$and so that the function and its approximation has the least difference (square error) in the subintervals (by the symmetric property of radial base function) $[0, x_1]$ and $[x_1, x_2]$ and $[x_2, 2]$. Fig. 12 shows the typical approximation of this case. Therefore, the following optimization problem is reached.

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

$$\min_{\substack{0<x_1<x_2<2 \\ m_1,m_2,m_3\leq 0}} L$$

Where L is:

$$L = \int_0^{x_1} \left(e^{-x^2} - (m_1 \left(x - \frac{x_1}{2} \right) + e^{-\left(\frac{x_1}{2}\right)^2}) \right)^2 dx + \int_{x_1}^{x_2} \left(e^{-x^2} - (m_2 \left(x - \frac{x_1 + x_2}{2} \right) + e^{-\left(\frac{x_1+x_2}{2}\right)^2}) \right)^2 dx + \int_{x_2}^{2} \left(e^{-x^2} - (m_3 \left(x - \frac{x_2 + 2}{2} \right) + e^{-\left(\frac{x_2+2}{2}\right)^2}) \right)^2 dx$$

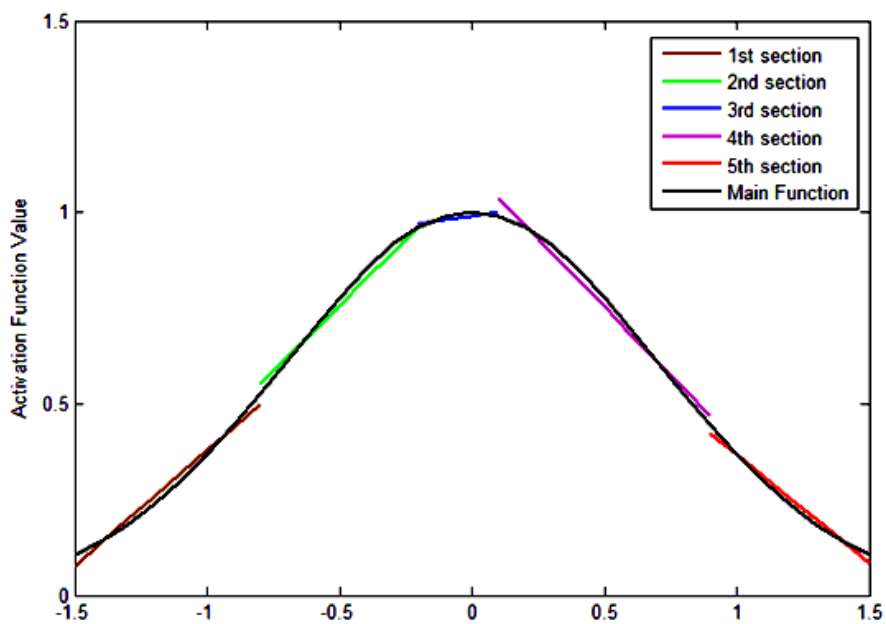


Fig. 12: Hybrid approximation

By employing Lagrange multipliers this optimization problem could be solved, but it is not tractable. Using evolutionary algorithms such as Genetic algorithm (GA) and by running a computer program in MATLAB, the solution is:

$$m_1 = -0.2150\,, \quad m_2 = -0.7652\,,$$

$$m_3 = -0.2283\,, \; x_1 = 0.2152\,, \; x_2 = 1.3010$$

Parameters of Genetic algorithm are shown in Table II and Fig. 13 shows the process of

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

convergence to the above solution. The objective (fitness) function of GA is defined the same as function L. To extract the synthesizable VHDL code the design is verified in MATLAB with HDL Coder toolbox and then the generated HDL code is passed through the ISE software for final rule checking and routing. The schematic of circuit for demodulator is shown in Fig.14. The control unit is designed by state flow diagram. Fig.15 shows the logic circuit of data control unit. Finally, the logic circuit for neurons is shown in Fig.16.

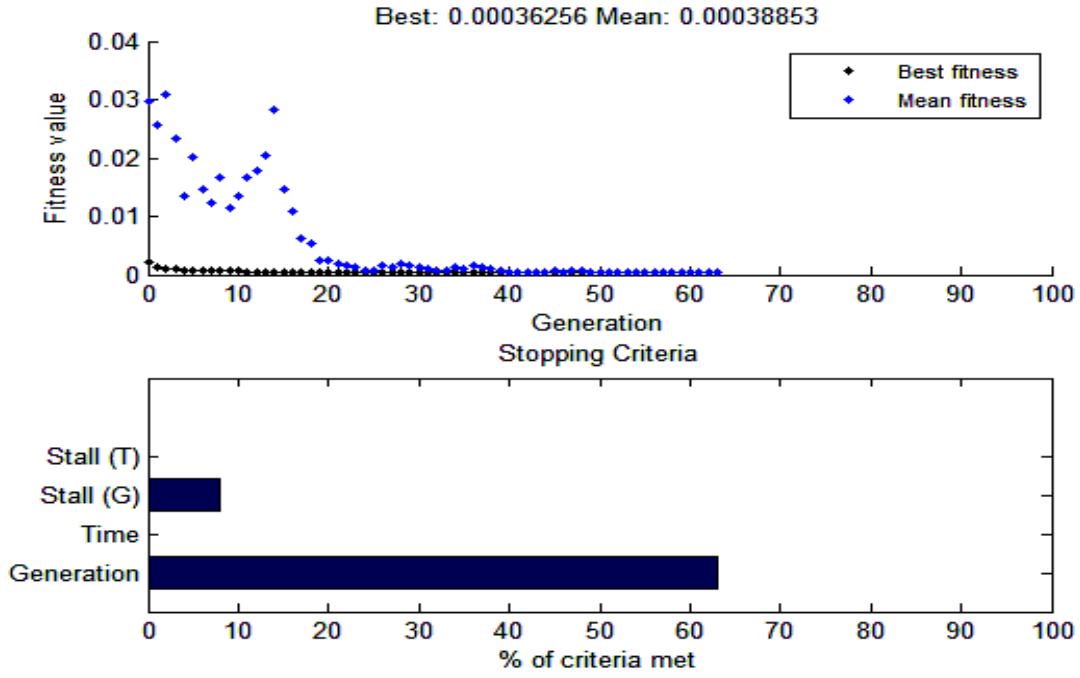


Fig. 13: Genetic algorithm convergence process

Table II: Genetic algorithm settings

Parameter name	Setting
Generations	100
Population Size	500
TolFun	1e-10
Fitness Limit	1e-6

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

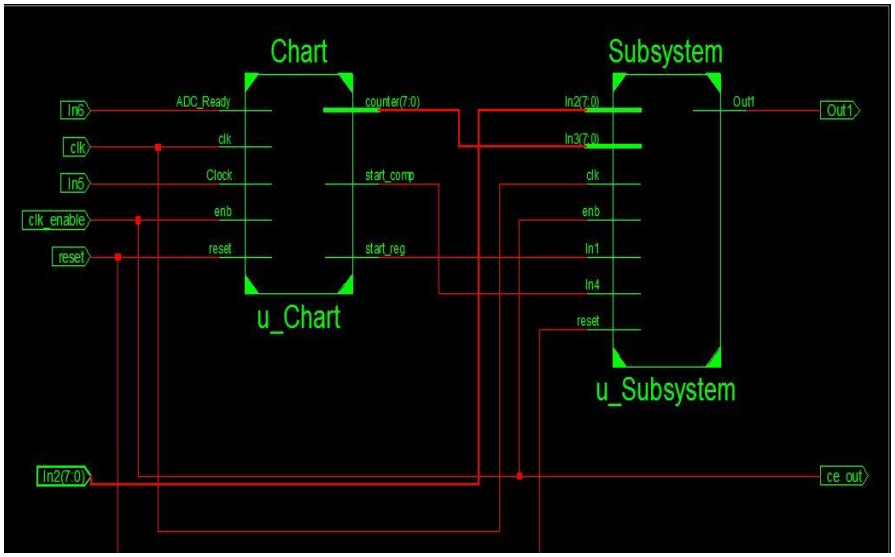


Fig. 14: Logic circuit schematic for demodulator

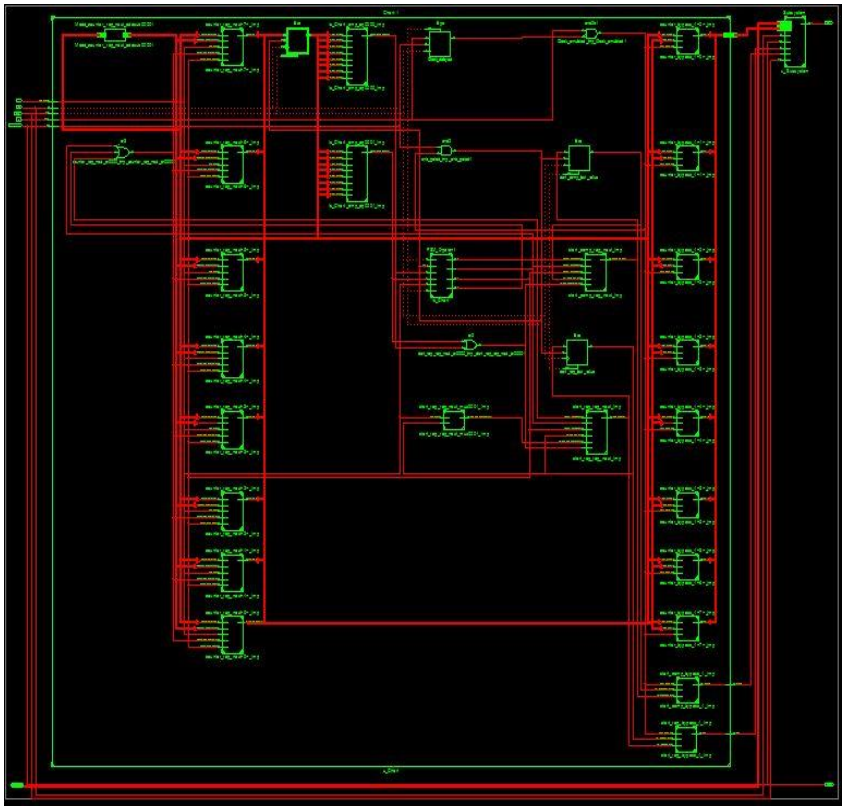


Fig. 15: Logic circuit of data control unit

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

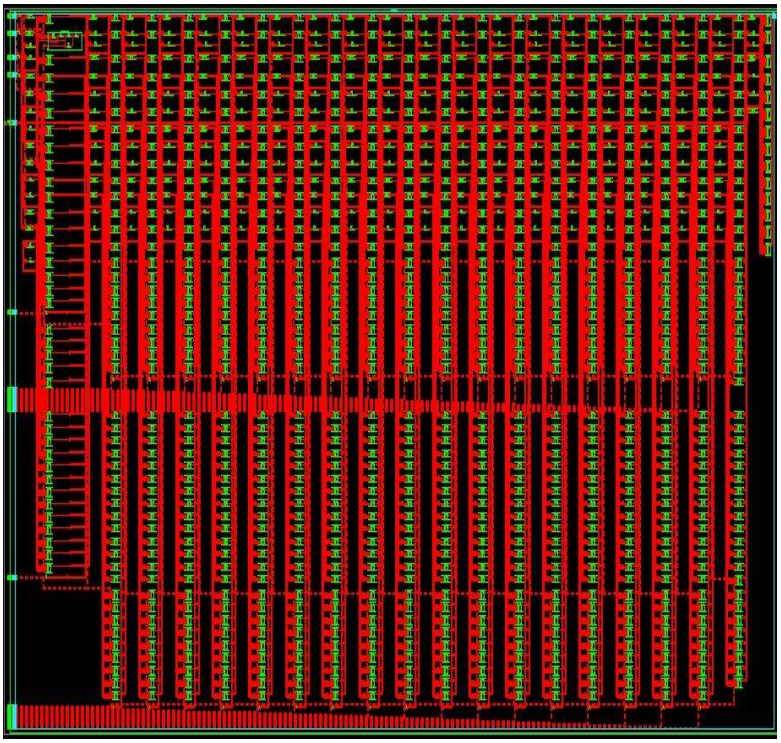


Fig. 16: Logic circuit of data control unit

Summary of device utility which is obtained by ISE is shown in Table III. The device used in this design is Virtex4.

Table III: Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	334	30,720	1%
Number of 4 input LUTs	4,877	30,720	15%
Number of occupied Slices	2,729	15,360	17%
Number of Slices containing only related logic	2,729	2,729	100%

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

Number of Slices containing unrelated logic	0	2,729	0%
Total Number of 4 input LUTs	5,058	30,720	16%
Number used as logic	4,877	-	-
Number used as a route-thru	181	-	-
Number of bonded IOBs	15	448	3%
Number of BUFG/BUFGCTRLs	1	32	3%
Number used as BUFGs	1	-	-
Average Fanout of Non-Clock Nets	2.36	-	-

VI. CONCLUSIONS

In this paper we have proposed a universal reconfigurable architecture using probabilistic neural network. Feasibility of design and implementation process along with challenging issues on implementing activation function are discussed. This architecture can be used in software defined radio and in its next generation cognitive radio because they must be flexible and adapt themselves with environment. The proposed structure can be easily extended for next standards and modulation schemes and can be used as a core in OFDM based systems.

Acknowledgements

This Study was supported by Islamic Azad University, Borujerd Branch, Iran. The authors would like to acknowledge staffs of university.

REFERENCES

[1] P. Koprinkova-Hristova, V. Mladenov, and N. K. Kasabov, Artificial Neural Networks: Springer International Publishing, 2015.

[2] L. Iliadis and C. Jayne, Engineering Applications of Neural Networks: Springer International Publishing, 2015.

[3] S. Bassis, A. Esposito, and F. C. Morabito, Recent Advances of Neural Network Models and Applications: Springer International Publishing, 2014.

[4] J. G. M. Oliveira, R. L. Moreno, O. d. O. Dutra, and T. C. Pimenta, "- Implementation of a reconfigurable neural network in FPGA," vol. -, pp. - 44, 2017.

[5] F. Ortega-Zamorano, J. M. Jerez, G. Juárez, J. O. Pérez, and L. Franco, "High precision FPGA implementation of neural network activation functions," in 2014 IEEE Symposium on Intelligent Embedded Systems (IES), 2014, pp. 55-60.

[6] H. Hu, J. Huang, J. Xing, and W. Wang, "Key Issues of FPGA Implementation of Neural Networks," in 2008 Second International Symposium on Intelligent Information Technology Application, 2008, pp. 259-263.

[7] S. Sahin, Y. Becerikli, and S. Yazici, "- Neural Network Implementation in Hardware Using FPGAs," pp. - 1112, 2006.

[8] A. R. Omondi and J. C. Rajapakse, FPGA Implementations of Neural Networks: Springer US, 2006.

[9] C. Thomos and G. Kalivas, "- FPGA-based architecture and implementation techniques of a low-complexity hybrid RAKE receiver for a DS-UWB communication system," vol. - 52, pp. - 2099, 2013.

[10] A. A. Moldovyan, N. A. Moldovyan, and N. Sklavos, "- Controlled elements for designing ciphers suitable to efficient VLSI implementation," vol. - 32, pp. - 163, 2006.

[11] A. Banerjee and A. S. Dhar, "Novel architecture for QAM modulator–demodulator and its generalization to multicarrier modulation," vol. - 29, pp. - 357, 2005.

[12] J. Bag, S. Roy, P. K. Dutta, and S. K. Sarkar, "Design of a DPSK Modem Using CORDIC Algorithm and Its FPGA Implementation," vol. - 60, pp. - 363, 2014.

[13] F. Angarita, M. J. Canet, T. Sansaloni, A. Perez-Pascual, and J. Valls, "- Efficient Mapping of CORDIC Algorithm for OFDM-Based WLAN," vol. - 52, 2007.

[14] S. W. Mondwurf, "- Versatile COFDM demodulator design based on the CORDIC-algorithm," vol. - 48, pp. - 723, 2002.

[15] M. Amini, A. Samimi, and A. Mirzavandi, "Joint optimisation of transmission and waiting times in cognitive

radio," International Journal of Electronics, vol. 103, pp. 747-764, 2016/04/02 2016.

[16] M. Amini and A. Mirzavandi, "Phase-Type Model Spectrum Sensing for Cognitive Radios," IETE Journal of Research, vol. 0, pp. 1-7, 2015 2015.

[17] M. R. Amini and E. Balarastaghi, "Universal Neural Network Demodulator for Software Defined Radio," International Journal of Machine Learning and Computing, vol. 1, pp. 305-310, 2011.

[18] M. R. Amini and E. Balarastaghi, "Improving ann bfsk demodulator performance with training data sequence sent by transmitter," presented at the Machine Learning and Computing (ICMLC), 2010 Second International Conference on, Bangalore, India, 2010.

[19] P. A. Kowalski and M. Kusy, "Sensitivity Analysis for Probabilistic Neural Network Structure Reduction," IEEE Transactions on Neural Networks and Learning Systems, vol. PP, pp. 1-14, 2017.

[20] N. T. Dukov, "Comparative evaluation of probability density estimators for the probabilistic neural network," in 2016 XXV International Scientific Conference Electronics (ET), 2016, pp. 1-4.

[21] C. Phaudphut, C. So-In, and W. Phusomsai, "A parallel probabilistic neural network ECG recognition architecture over GPU platforms," in 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2016, pp. 1-7.

[22] S. Ramakrishnan and I. M. M. El Emary, "- Classification brain MR images through a fuzzy multiwavelets based GMM and probabilistic neural networks," vol. - 46, pp. - 252, 2011.

[23] O. Vysochyna and A. M. Salman, "Using probabilistic neural networks for diagnostics and prediction model of the router state," in 2010 International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), 2010, pp. 255-255.

[24] S. Kala, "- Design of a software defined radio receiver using FM channelizer — A case study," vol. -, pp. - 4, 2016.

[25] H. Arslan and H. Celebi, "- Software Defined Radio Architectures for Cognitive Radios," pp. - 144, 2007.

[26] M. R. Amini, M. Mahdavi, and M. J. Omidi, "Analysis of a multi-user cognitive radio network considering primary users return," Computers & Electrical Engineering, vol. 53, pp. 73-88, 7// 2016.

[27] M. R. Amini, F. Hemati, and A. Mirzavandi, "Trilateral Tradeoff of Sensing, Transmission, and Contention Times in a Multiuser Split-Phase CR Networks," IEEE Sensors Journal, vol. 15, pp. 6044-6055, 2015.

ISSN: 2456-1851

Volume:03, Issue:01 "January-February 2018"

[28] B. Dacorogna, Direct Methods in the Calculus of Variations: Springer-Verlag New York, 2008.

[29] M. Razzaghi and H.-R. Marzban, "Direct method for variational problems via hybrid of block-pulse and chebyshev functions," Mathematical Problems in Engineering, vol. 6, pp. 85-97, 2000.

