Design Neural Wireless Sensor Network Using FPG*A*

Ban M. Khamma*s*
Information Engineering College , University of Al Nahrain / Baghdad
Email: Banmoh-79@yahoo.co***m***

Received on: 22/6/2011 & Accepted on: 1/3/2012

ABSTRACT

Wireless sensor networks(WSN) are an exiting emerging technology that scientists believe to become a part of every day life in the next few years. However, at this time many issues in wireless sensor networks remain unresolved. This paper studies the architecture of a neural wireless sensor network designed to identify technical condition of the base station of wireless sensor networks ,and this work presents an on-chip artificial neural networks(ANN) in a Field Programmable Gate Arrays (FPGA) system. In order to take maximum advantage of the distributed architecture of multiple NN systems is to providing a high degree of parallelism between NNs and, hence, a higher speed-up in relation to a sequential implementation. The goal of this work is to realize the hardware implementation of the base station of neural wireless sensor network using FPGAs to measure hamidi ty, temperature and light for the security system of an office.

Keywords:- Wireless sensor networks, artificial neural networks, FPGA.

تصميم شبكة المتحسس اللاسلكي العصبي بواسطة أستخدامFPGA

الخلاصة

شبكات المتحسس اللاسلكي هي تكنلوجيا حديثة منبثقة يعتقد العلماء انها ستصبح جزء مـــن الحياة اليومية خلال السنوات القليلة القادمة. و مع ذلك ، فأن هنالك عدة مشاكل فـــي شـــبكات المتحسس اللاسلكي بقية غير محلولة. هذا البحث يقوم بدراسة تصميم معمارية شبكة المتحسس اللاسلكي العصبي لمعرفة الشروط التقنية للقاعدة الاساسية لشبكات المتحسس اللاسلكي . وان هذه العملية ستتم بواسطة تمثيل الشبكة العصبية على الرقائق الالكترونية الممثلة بال FPGA . ان الاستفادة من استخدام عملية التجزئة لمعمارية الشبكة العصبية هي للحصول على درجة عالية من التوازي بين الشبكة العصبية وبالتالي ستؤدي الى زيادة السرعة الخاصة بالتنفيذ المتعاقب. الهدف من هذا البحث هو البناء المادي للقاعدة الاساسية لشبكة المتحسس اللاسلكي العصبي باســـتعمال FPGA لبناء نظام حماية لمقر عمل ما و ذلك من خلال التحسس بالرطوبـــة و الحـــرارة و الضوء .

INTRODUCTION

Wireless communication is one of the most vibrant research areas in the communication field today. While it has been a topic of study since the 60's, the past decade has seen a surge of research activities in the area. This is due to a confluence of several factors. First is the explosive increase in demand for connectivity, driven so far mainly by cellular telephony but is expected

https://doi.org/10.30684/etj.30.9.15
2412-0758/University of Technology-Iraq, Baghdad, Iraq

to be soon eclipsed by wireless data applications. Second, the dramatic progress in VLSI technology has enabled small-area and low-power implementation of sophisticated signal processing algorithms and coding techniques. [1]

With the requirements for advanced integration, intensive onboard processing, and low power consumption, FPGAs emerge as a technology of choice that strikes an optimal balance between processing power, energy requirements, and flexibility. Through the power of reconfigurability, wireless sensor network designs containing reprogrammable logic can be upgraded, errors can be fixed, and limited-resource applications can be dynamically reprogrammed in the field. [1]

The programmability of reconfigurable FPGAs yields the availability of fast special purpose hardware for wide applications. Its programmability could set the conditions to explore new neural network algorithms and problems of a scale that would not be feasible with conventional processor.

The concept of ANNs is emerged from the principles of brain that are adapted to digital computers. The first works of ANNs were the models of neurons in brain using mathematics rule [2]. These works show that each neuron in ANNs take some information as an input from another neuron or from an external input. This information is propagated as an output that are computed as weighted sum of inputs and applied as non-linear function.

Architectural ANNs parameters such as number of inputs per neuron and each neuron's conductivity change remarkably from one application to another. Thus, for special purpose network architectures parameters must be carefully balanced for efficient implementation. [3]

Neural Network (NN) Implementation in hardware using FPGAs are chosen for implementation ANNs with the following reasons:

1. They can be applied a wide range of logic gates starting with tens of thousands up to few millions gates.
2. They can be reconfigured to change logic function while resident in the system.
3. FPGAs have short design cycle that leads to fairly inexpensive logic design.
4. FPGAs have parallelism in their nature. Thus, they have parallel computing environment and allows logic cycle design to work parallel.
5. They have powerful design, programming and syntheses tools.

At the first step of FPGAs system design, the architecture of ANNs must be specified with either schematic or algorithmic[3].

The idea of using FPGA in WSN have been used by Jorge P. , and other [4], design a modular architecture for nodes in WSN, the architecture is made up of four layers, these layers are (communication, processing, power supply and sensing/actuating). A set of such layers used to create the node for a specific application. Changing of application implies changing the layers or redesigning one or some of them from scratch all the system, which drives to high cost and time inefficiency. Other trend have been used by Stephen J. , and other [5], there work partitions the wireless sensor module into a series of layers with area (25 mm X 25 mm), The modulehas resulted in the specification of a series of layers including a FPGA for signal processing. Forming the initial elements of the 25 mm sensor node toolkit that can be programmed for use with different sensors and application.Freek Z. [6], shows that WSN and NNs can be combined into a

working system, in this case, the system tries to detect the shadows of clouds as they drift over a field.
From the above works, a new trend can be used that combine NN with WSN for asecurity system and implement the system using FPGA.

WSN and NN

WSNs are an emerging technology which holds the potential to revolutionize everyday life [7, 8]. Continuous advances in semiconductor technology have enabled miniaturization of radios and mechanical structures and deployment of very cost-efficient wireless sensor nodes.

The number of applications which can take advantage of wireless sensor networks is huge. Among these multiple applications we have:

1. Home or office automation and security.
2. Health systems.
3. Structural monitoring.
4. Traffic control.
5. Military applications.
6. Agriculture.
7. Elderly people surveillance.[4]

A sensor node basically consists of a microcontroller (processor and memory), sensors, analog-to-digital converter (ADC), transceiver (sender and receiver) and power supply. The abstract architecture of a sensor node and the dependencies of its units are depicted in fig.(1).

The structure of neural networks and wireless sensor networks can both be described as a collection of interconnected simple components. Both kinds of networks are also similar in that they implement a function that maps input values to output values. These similarities between wireless sensor networks and neural networks led to the construction and investigation of a system that combines both technologies. Physical limitations however complicate straightforward combination of both technologies.[6]

For example, neural-network theory usually assumes data transport to be free, but in wireless sensor networks input and output are usually geographically separated. The cost of the connection is not the only issue, neural networks are very sensitive to faulty connections; wireless sensor networks usually have unreliable radio links. Another difference is in the structure of the network. The random distribution of components, and tree-like topology in wireless sensor networks, is not necessarily a suitable neural network topology for the application at hand. With an unsuitable topology, the redundancy in wireless sensor networks could lead to overtraining in the neural network, and make it more sensitive to noise[1].Physical limitations however complicate straightforward combination of both technologies.

In general Artificial Neural Networks (ANN) show characteristics such as distributed representation and processing, massive parallelism, learning and generalization ability, adaptively, inherent contextual information processing, fault tolerance and low computation. Many of those characteristics are either inherent or desirable for WSNs as well. As shown in fig.2, we transfer the perception of a biological neuron via an artificial neuron to the sensor node. The functionality h(x) is basically a weighted sum over all inputs. The sensor converts the physical world to an electrical signal which is filtered or preprocessed corresponding to weighting/synapse. The subsequent processing within the processor corresponds to the chemical processing accomplished by the soma or applying the particular functionality h(x), respectively.

Finally, the sensor node sends out the modified sensor reading via the radio link. This strong analogy shows that the sensor node itself can be seen as a biological and artificial neuron, respectively.

Therefore, efficient neural network implementations using simple computations can replace traditional signal processing algorithms to enable sensor nodes to process data by using less resources[9].

In general, the NN approach can potentially achieve increased reliability, dimension reduction, error correction and sensor validation[10]. Scientists believe that this process is similar to what happens in the nervous system. The decentralized structure and emergent behavior of neural networks closely resembles how we would want wireless sensor networks to work. This inspired us to see if neural network techniques can indeed solve some of the problems in wireless sensor networks.

The most basic element of the neural network, the neuron, transforms its inputs in the feed forward stage of operation according to the steps in (1) as shown in Fig. (3).

1) Multiply each input by its corresponding weight.
2) Sum the multiplication results.
3) “Squash” the sum using a transfer function. [11]

This concept of neuron structure is used in the NN that design in this paper.

SYSTEM DESIGN

Any system consisting of computer nodes that sense, reason about and act upon their environment may be called an ambient intelligence system. One major distinction that can be made is between single sensor systems and sensor networks, where data from multiple, and possibly different sorts of sensors is fused. The tasks our system has to perform clearly demand that multiple sensors are spread throughout the environment, and so we must use a sensor network. Note that since we do not focus on the acting capabilities of the system, we shall from now on speak of our system as a sensor network.

The aim of this paper is design security system using neural wireless sensor network**.** This security system offer suitable environments for an office.

The system consists of several neural network modules operate in parallel[12]. Each module is a three-layer neural network and has its own training data set. A high speed back propagation learning algorithm has been used to train the networks. Before learning starts, tolerances are defined for the output units. During learning, the weights are updated only when the output errors exceed the tolerances. The learning data for which the output errors do not exceed the tolerances are eliminated from the training data sets [10].Parameter selected and used for :-

1. Determine the learning algorithm (back propagation algorithm).
2. Determine the structure of input, hidden and output nodes of the network.
3. Determine the activation function (log-sigmoid function).
4. Determine the learning rate. The learning rate is set to 0.1.

5. Determine the number of iterations. Itdepends on the error generated from BPNN.

The base station system is design to received signals from multiple sensors (temperature sensors, humidity sensors ,and light sensors), which measure the total solar radiation (TSR) and the photo synthetically active radiation (PAR).

The number of input neurons equals the number of bits that use from all sensors. To be more precise, this paper use the following number of bits related to the particular sensors (7 bits for temperature (the measure range 0-100^{C}), 7 bits for humidity (the measure range 0-100%) , and 7 bits for each light sensor, with the TSR and PAR measure range 0-100% .

Also this paper use a offline learning method, the software that chosen for offline training is MATLAB.

The MLP neural network with one hidden layer is used for every sub- neural networks system . Fig. 4 shows the default three layers MLP sub- neural network which is designed by MATLAB in this research. Each layer includes some sub-blocks .

The security system that design in this research working by controlling four neural networks modules in base system of wireless sensor network as depicted in fig. 5.

The details of neural WSN design:-

1- Select the number of input neuron for each sub neural network :(7 input neurons for temperature , 7 input neurons for humidity , 7 input neurons for TSR , 7 input neurons for PAR).

2- Select the number of neuron in hidden layer for each sub neural network :(3 input neurons for temperature , 3input neurons for humidity , 3 input neurons for TSR , 3 input neurons for PAR).

3- Select the number of output neuron for each sub neural network :(1 input neurons for temperature , 1 input neurons for humidity , 1 input neurons for TSR , 1 input neurons for PAR).

Then detail of the complete neural network is shown in figure (6).

The system take extremely short training times. The network runs entirely autonomously ; it does not need any outside control, it can operate at the same time, provides fast access to match results, and is designed to work with infinite stream of data. All these features make it an excellent choice for applications in wireless sensor networks.The initial parameters of the NWSN used in this research are given in table (1) .These parameters are used in MATLAB program to obtain the final correct weights of NN, which will be used in the hardware design of NWSN. Then these weights be converted to binary form, and save these binary weights in a file for transmitting it to the Xilinx program. By using the method of conversion fractional decimal to fractional binary number as in the following example which convert 0.95 to its binary equivalent[13]:

0.95 * 2=1.90	integer part : 1
0.90 * 2=1.80	integer part : 1
0.80 * 2=1.60	integer part : 1
0.60 * 2=1.20	integer part : 1
0.20 * 2=0.40	integer part : 0
0.40 * 2=0.80	integer part : 0

0.80 * 2=1.60 fraction repeats last four digits

$(0.95)_{10} = (0.111100)_2$

The procedure design of NWSN as seen in figure(7) consist of two parts: The first part is learning and training of NN which implemented by using Matlab software. The second part is hardware design of NWSN which its implemented by using Xilinx program. The second part receive file of binary weights from the first part.

Back-propagation algorithm is the most popular approach to implement learning in neural networks (also called generalized gradient descent and generalized delta rule) [14,15]. This algorithm is a multi-layer network using a weight adjustment based on the sigmoid function; The sigmoidal function is very popular for neural networks, because it performs very similar to a step function, but it is everywhere differentiable. The standard form of the sigmoidal is:

$$f\,(sumt) = \frac{1}{1 + e^{-sumt}} \qquad \text{.... (1)}$$

Where fis the transfer function.

$$sumt=\sum X_j W_j+b \qquad \text{...(2)}$$

sumt: is the summation of multiplying the input and the weight.

Xj : is the input .

W_j: is the weight associated to each input .

b: is the biased.[16]

This equation was selected because it provides the necessary limiting of the outputs while having some properties, which are useful in the learning phase of the algorithm.

Unfortunately, this equation contains the transcendental function (exponential term, exp), which is somewhat difficult to calculate. Nordstrom and Svensson[16], list several functions, which may be used as an approximation to the function used by McClelland and Rumelhart. These functions all have the same general characteristics. They are continuously increasing, approach 0 at $-\infty$ and 1 at $+\infty$, and have a continuous first derivativeThe approximation that used in this neural network is given by the function:

$$f(net) = \frac{1}{2}\left[\frac{net}{1 + |net|} + 1\right] \text{... (3)}$$

This equation is convert to hardware and name it a transfer function block in Figure(8).The activation Block in Figure(8), which performs the multiplications $X_jW_j$and the summation of these multiplied terms as in Equation (2), is always on the neuro-chip (or the processing element of the neurocomputer). The data flow between these blocks is controlled by the Control Unit that is always on the chip.The data flow is such that the weights from the Weights Block and the inputs from outside or from the outputs are multiplied and the products are summed in the Activation Block, then the outputs are obtained in the Neuron State Block from the transferred sum of the products . Neuron states and weights can be stored in digital form. Weights can be stored in registers if it is little numbers of weights or stored in RAM .

Figure(9) represent overall hardware design of NWSN which explained as a block diagram in figure(5)that represents ((temperature NWSN, humidity NWSN ,and light(TSR-NWSN , PAR-NWSN)), for the base station of WSN.

The figure(10) explain the details of hardware design for each module (sub-neural network)which contain multiple HIDDENNEURON block (neuron which explained as a block diagram in figure(8) for each NN module.

The details of hardware design for each neuron (containing the design of the activation function according to equation 3) is shown in figure(11).

RESULT AND DISCUSSION

The process of training is shown in the following figures (12, 13, 14, 15) in which training curves is approaching its goal through readjustment of weights and biases using MATLAB program. Figure(16) shows the comparison between Matlab results and Xilinx results for the outputs of NWSN, they are very close.

The result sample of the system shown in table 2 is the correct results as in fact. The result of FPGA , for an example the humidity sub-neural network that built above, and after specified input values of the weights, and the input values to the neuron, shown in fig. 17, and the result of complete operation of the total NN is presented in fig. 18 .

Where:-h39.out7: is the output result for temp. , h41.out7: is the output result for PAR , h42.out7 : is the output result for TSR .

CONCLUSIONS

A modular architecture for the network shown that the analogy between WSNs and ANNs is not only theoretical, but can be also used to actually implement this paradigm efficiently into real WSNs. It is important to stress that ANNs exhibit exactly the same architecture as WSNs, and it is shown that implementation of neural networks using FPGAs. The resultant neural networks are modular, compact, and efficient and the number of neurons, number of hidden layers and number of inputs are easily changed.

The choice of precision and arithmetic hardware architecture used in neural networks application has a direct impact on the processing density achieved. Using suitable precision arithmetic design, one can achieve adequately high speed and small size for real time ANNs implementations.

In order to validate this work, the results are compared with the reported by Frank O. and Petri M¨ah[10] which use Hopfield network (HN) with a lot of neurons (480 neurons)to get stable HN with single layer that will increase the hardware that use to the hardware design and decrease the speed , and the result recognition rate is 83% .

The agreement is found to be acceptable in this work, which validates the present recognition rate is 98 % with the facility of using sub- NN that use BP of multiple layers and decreasing the number of neurons and increase the speed of the system that design in FPGA.

However this study shows that FPGAs are versatile devices for implementing many different applications. The FPGA shown to be a very powerful embedded system design tool, with low cost, reliability, and multi-faceted applications. As FPGAs allow the hardware design via configuration software control, the improvement of circuitry design is just a matter of modifying, debugging and downloading the new configuration code in a short time.

REFERENCES

[1] Joel Wilder, Vladimir Uzelac, **"Wireless Sensor Networks for Updating Reprogrammable Logic Designs in Real-time"**, The University of Alabama in Huntsville, 301 Sparkman Drive, Huntsville, AL 35899 , 2007.

[2] HAYKIN, S. ,**"Neural Networks A Comprehensive Foundation"**, 2nd edition, Prentice Hall Publishing, New Jersey 07458, USA, Vol.1, pp 6-7 , . 1999 .

[3]SuhapSahin, YasarBecerikli*, and SuleymanYazici , **"Neural Network Implementation in Hardware Using FPGAs" ,** Department of Computer Eng., Kocaeli University, Izmit ,Turkey , ICONIP 2006, Part III, LNCS 4234, pp. 1105 – 1112, 2006.

© Springer-Verlag Berlin Heidelberg 2006

[4]Jorge Portilla , Angel de Castro , Eduardo de la Torre , Teresa Riesgo, **"A Modular Architecture for Nodes in Wireless Sensor Networks " ,**Journal of Universal Computer Science, vol. 12, no. 3 (2006), 328-339 , © J.UCS , (Universidad Politécnica de Madrid, Spain.

[5]Stephen J. Bellis*, Kieran Delaney, Brendan O'Flynn, John Barton, Kafil M. Razeeb, CianO'Mathuna, **"Development of field programmable modular wireless sensor network nodes for ambient systems",**ELSEVIER , computer communications, May 2005

[6]FreekZindel, " **A Neural Infrastructure for Wireless Sensor Networks"**Master's Thesis in Computer Science , Parallel and Distributed Systems group, Faculty of Electrical Engineering, Mathematics, and Computer Science ,Delft University of Technology,15th January 2006.

[7] omer, K. R¨ and F. Mattern, **"The design space of wireless sensor networks" ,** *IEEE Wireless Comm.*, vol. 11, no. 6, pp. 54–61, Dec. 2004.

[8] Culler,D. D. Estrin, and M. Srivastava**, "Overview of sensor networks,"**IEEE Computer, vol. 37, no. 8, pp. 41–49, Aug. 2004.

[9] Savich, A. M. Moussa, and S. Areibi, **"The impact of arithmetic representation on implementing MLP-BP on FPGAs: a study",** IEEE Transactions on Neural Networks, vol. 18, no 1, pp. 240 - 252, Jan 2007.

[10]Frank Oldewurtel and Petri M¨ah¨onen**"Neural Wireless Sensor Networks"** ,RWTH Aachen University, Department of Wireless Networks ,Kackertstrasse 9, D-52072 Aachen, Germany

[11]Antony W. Savich, MedhatMoussa**,**ShawkiAreibi, **"The Impact of Arithmetic Representation on Implementing MLP-BP on FPGAs: A Study ",**IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 18, NO. 1, JANUARY 2007.

[12]Duc T. Pham and Liu Xing **"Neural Networks for Identification, Prediction and Control ",**printed in Great Britain , London , 3rd printing 1997.

[13]Robert K. Dueck,“**Digital Design with CPLD Applications and VHDL”,** Delmar Thomson learning , ISBN0-7668-1160-3,© C 2001.

[14]YihuaLiao , **“Neural Networks in Hardware:ASurvey”,** Department of Computer Science, University of California, Davis One Shields Avenue, Davis , 1999

[15]Haitham Kareem Ali and EsraaZeki Mohammed **“Design Artificial Neural Network Using FPGA”**IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.8, August, 2010

[16] Nordstrom, T. and Svensson, B. " Using and Designing Massively Parallel Computers for Artificial Neural Networks" Journal of Parallel and Distributed Processing, no.3, pp. 260-285, March 1998.

[17]Anil K. Jain, Robert P.W. Duin and Jianchang, **"Statistical Pattern Recognition:AReview,",**IEEETRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 1, JANUARY 2000.

[18] Maischberger,O. v. Salapura , **" A Fast FPGA Implementation of a General Purpose Neuron ",** Technical University , Institute of informatik , Austria , 2006.

[19] Ryuji TANAKA, Kunihiro UEDA, Koichiro TANAKA , “**DESIGN OF A ROUTER FOR WIRELESS SENSOR NETWORKS “ ,** Faculty of Information Science, Kyushu Sangyo University , 2-3-1 Matsukadai, Higashi-ku, Fukuoka-shi, Fukuoka 813-8503, Japan , 2010 .

[20] Raeisi, R. and A. Kabir, **“Implementation of Artificial Neural Network on FPGA”,** American Society for Engineering Education, Indiana and North Central Joint Section Conference, 2006.

[21] Assist. Prof. Dr. Hanan A. R. Akkar, 2M. Sc. Student Firas R. Mahdi , “ **Implementation of Digital Circuits Using Neuro - Swarm Based on FPGA “,** International Journal of Advancements in Computing Technology Volume 2, Number 2, June, 2010

[22]A.Muthuramalingam,S. Himavathi and E. Srinivasan, **“Neural Network Implementation Using FPGA: Issues and Application”,** International Journal of Information Technology, Vol. 4, No. 2, PP 86-92.

[23] Turner, R. H. R. F. Woods**, “Highly efficient limited range multipliers for LUT- based FPGA architecture”,** IEEE transactions on vary large scale integration system Vol. 15, No.10, 2004, PP 1113-1117.

[24] Noory, B. and V. Grozo**, “A Reconfigurable Approach to Hardware Implementation of Neural Network”**, Canadian conference on Electrical and Computer Engineering IEEE, CCGE1, 2003, PP1861-1863

[25]Freek van Polen, **” Towards a Sentient Environment Using a Neural Sensor Network”**,Master’s Thesis , Cognitive Artificial Intelligence ,Utrecht University , November 2, 2008.

[26] Assist. Prof. Dr. Hanan A. R. Akkar, 2M. Sc. Student Firas R. Mahdi ,“**Implementation of Digital Circuits Using Neuro - Swarm Based on FPGA “,** International Journal of Advancements in Computing Technology Volume 2, Number 2, June, 2010 .

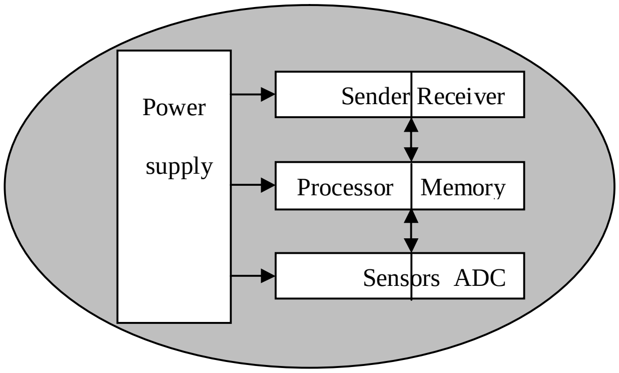


Figure (1) A simplified architecture of a sensor node.

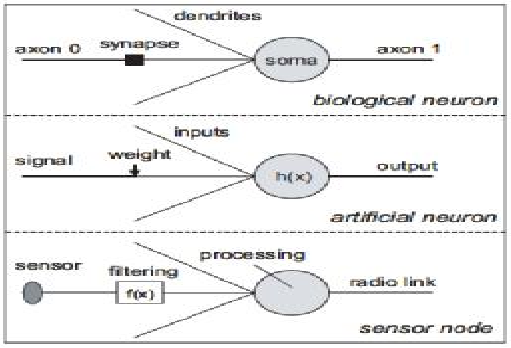


Figure (2) the analogy of the biological neuron (Simplified), Artificial neuron and the sensor node

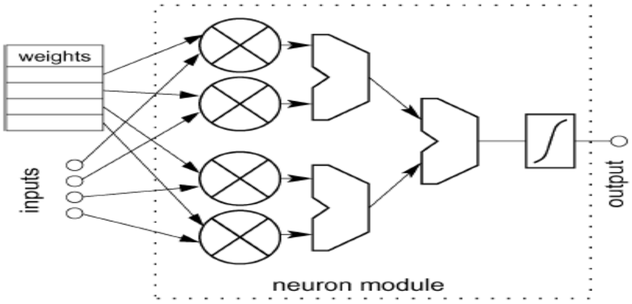


Figure (3) A neuron structure.

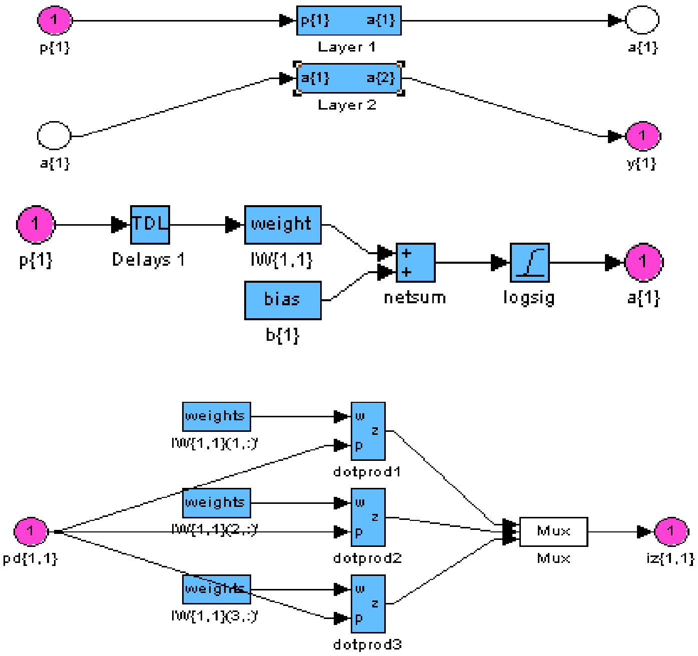


Figure (4) Two layer MLP neural network and sub-block of layer.

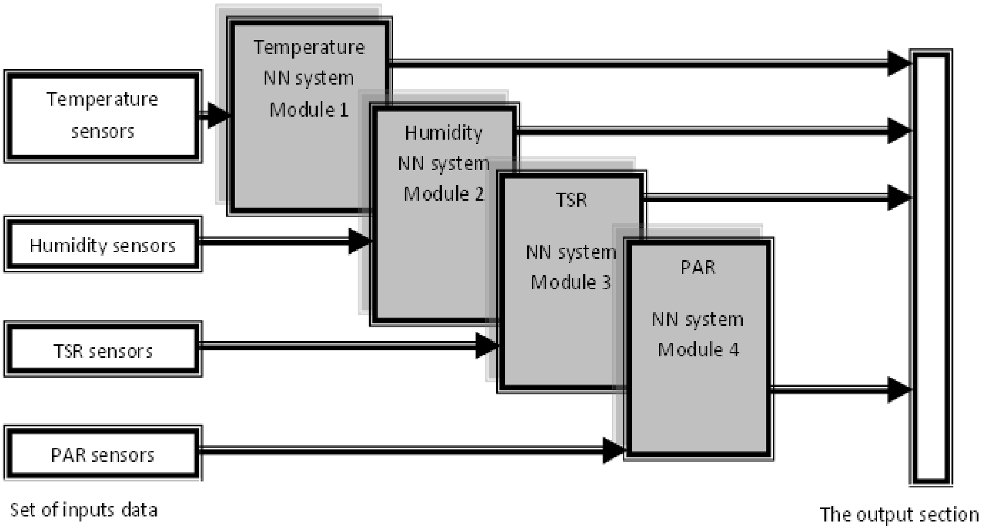


Figure (5) Block diagram of NWSN.

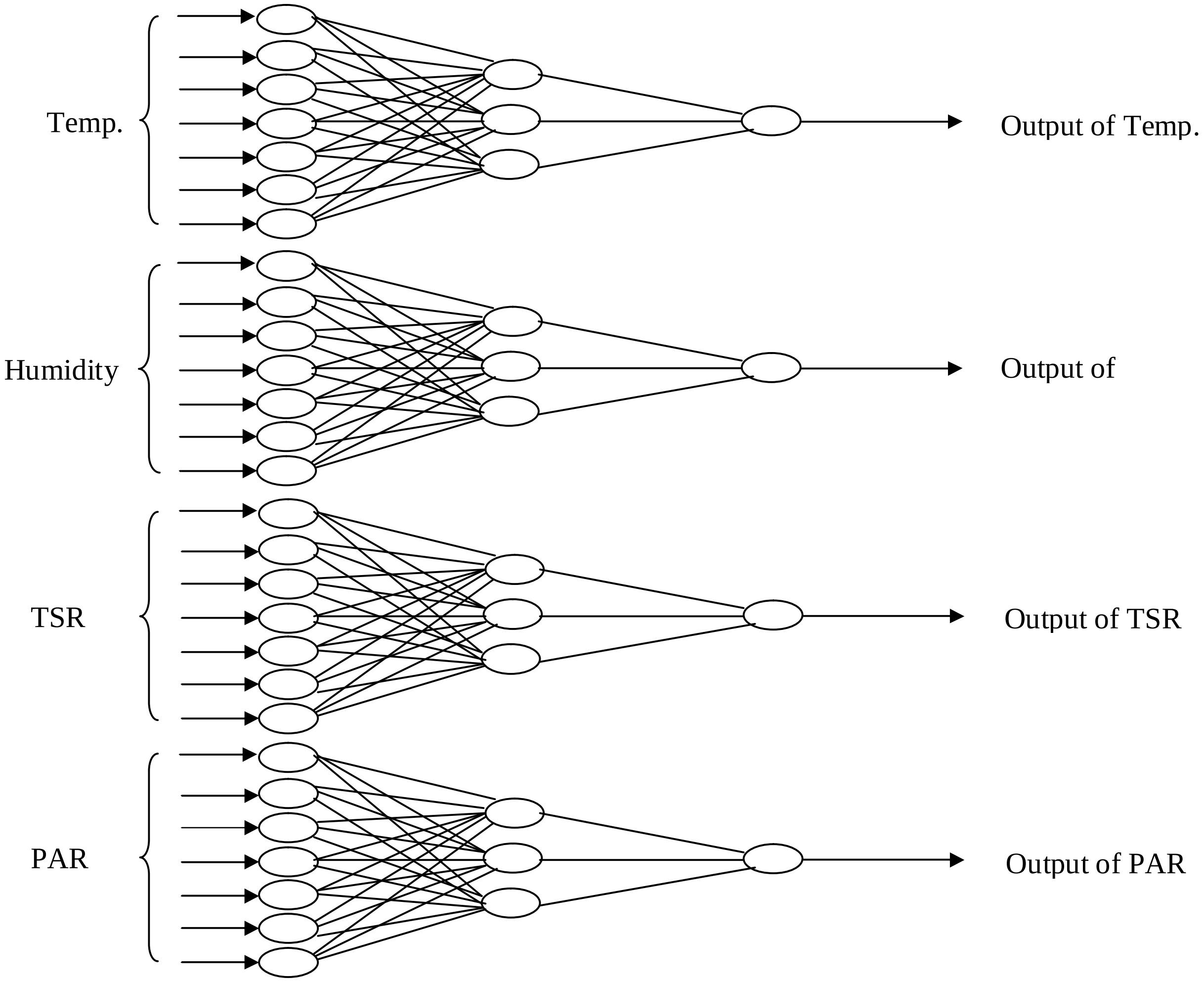


Figure (6) the complete NN of NWSN.

Table (1) : Experimental setup for each sub-NN systems.

Parameter	Value
Number of input neuron	7
Number of hidden layers	1
Number of total output neuron	1
Number of total neurons in hidden layers	3
Input range	[0,1]
Output range	[0,1]
Activation function of neurons in hidden layer	logsig
Activation function of neurons in output layer	logsig
Type of neural network	Feed forward back propagation network
Training Function	Trainlm
Number of epochs used in training	10000
Back propagation weight/bias learning function	learngdm
Error tolerance	0.000000001
Performance function	mse

Table (2) : NWSN results.

Case	*Output of temp. NN (1 bit)*	*Output of humidity NN (1 bit)*	*Output of TSR NN (1 bit)*	*Output of PAR NN (1 bit)*
Temp. < 30^C	0	--	--	--
Temp. > 30^C	1	--	--	--
humidity < 50%	--	0	--	--
humidity > 50%	--	1	--	--
TSR < 50%	--	--	0	--
TSR > 50%	--	--	1	--
PAR < 50%	--	--	--	0
PAR > 50%	--	--	--	1

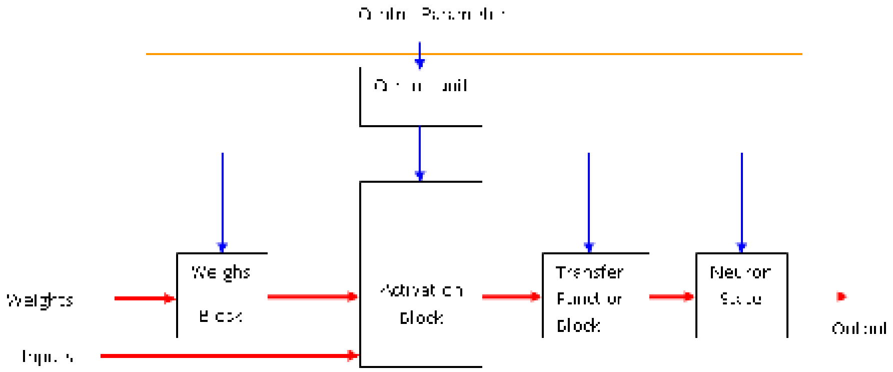


Figure (7): Design environment.

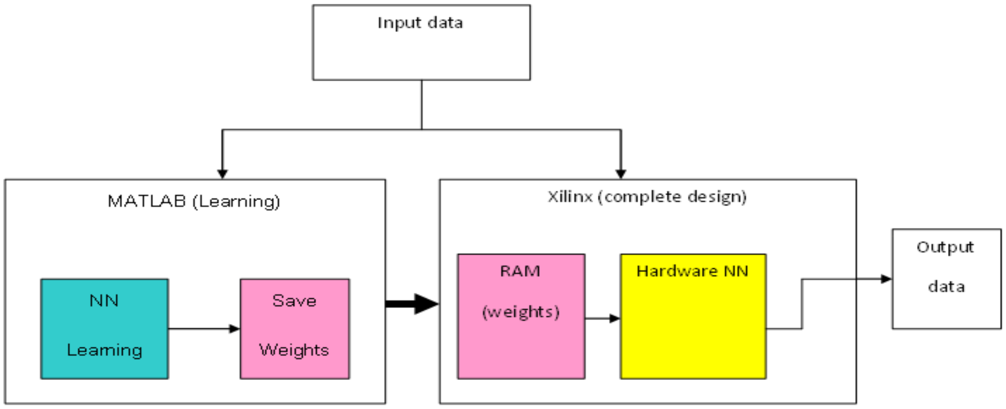


Figure (8): The activation Block of the hardware neuron.

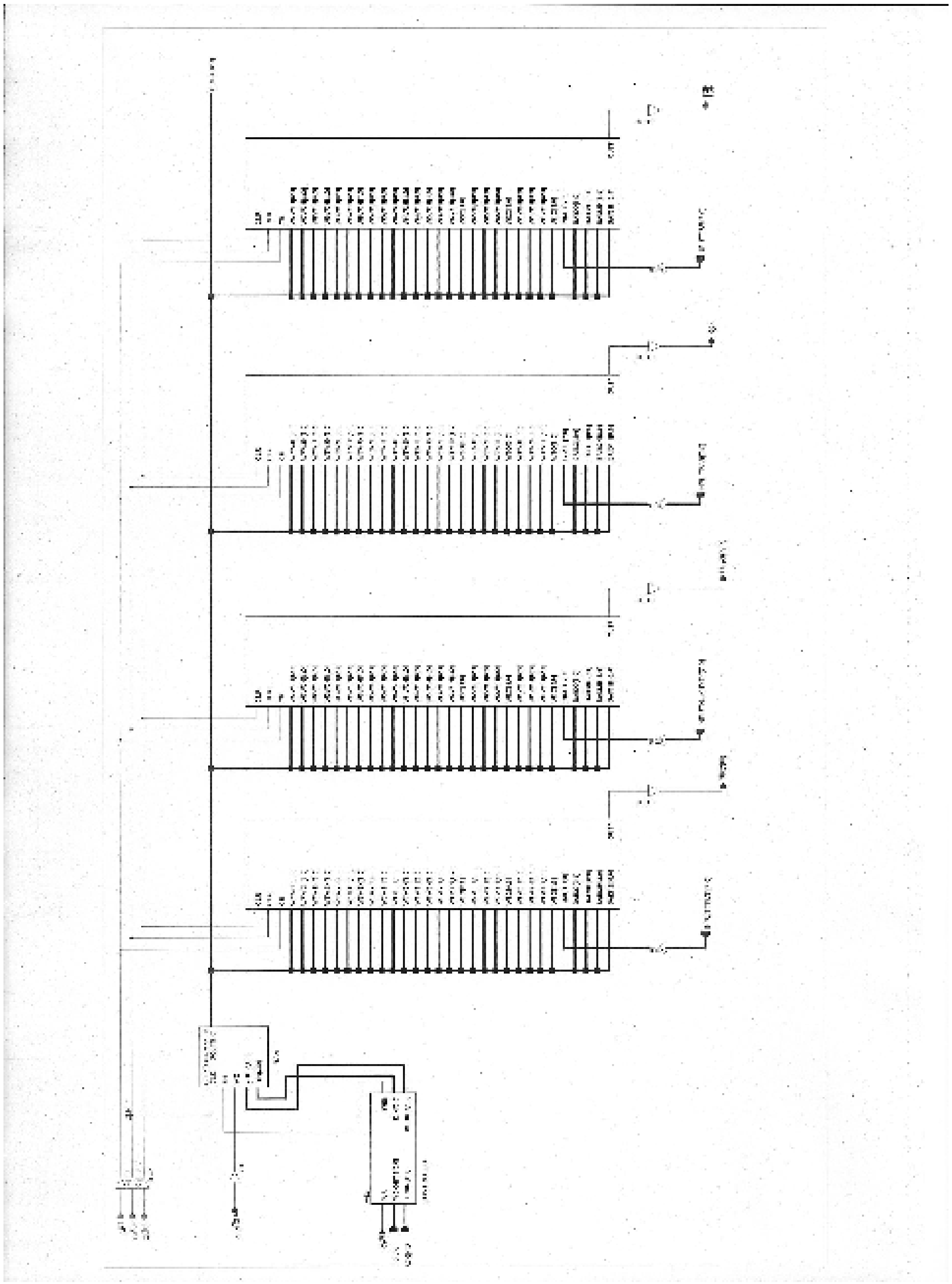


Figure (9): The complete system of NWSN

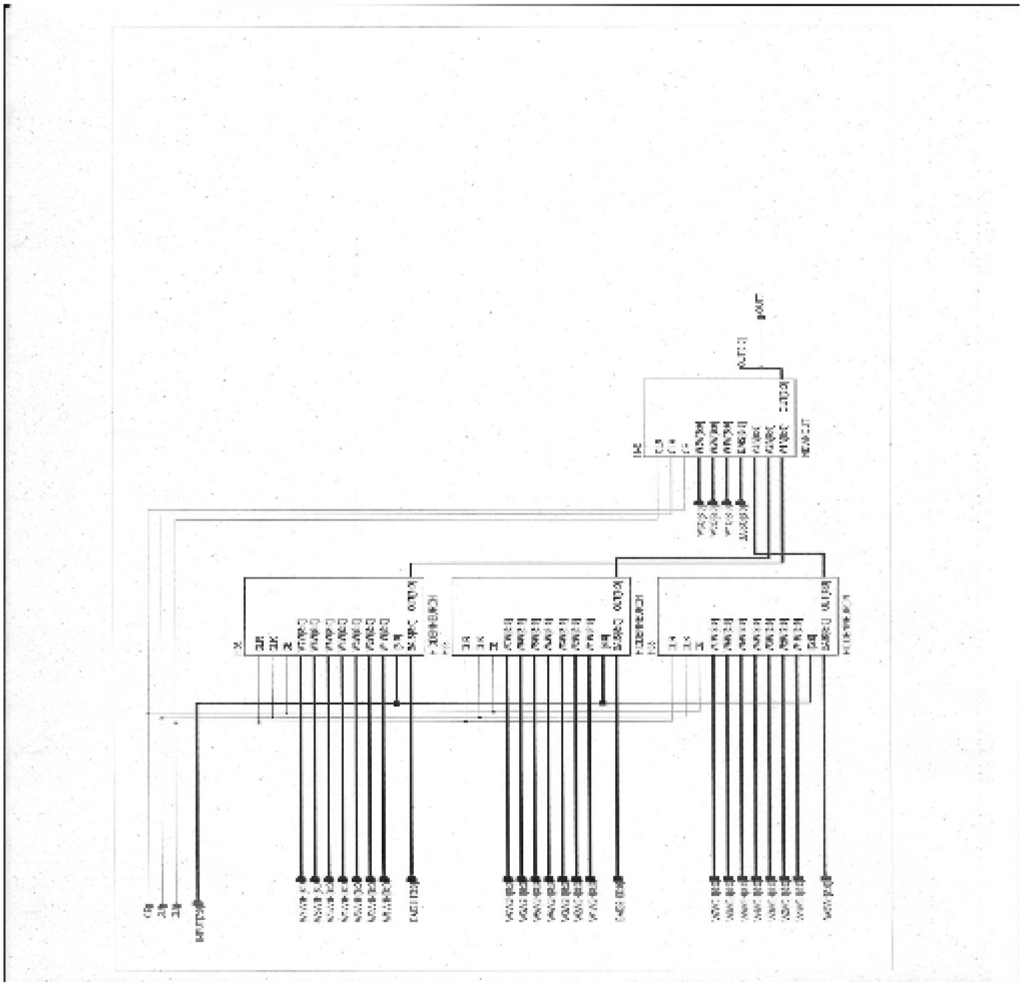


Figure (10): Sub-neural network circuit

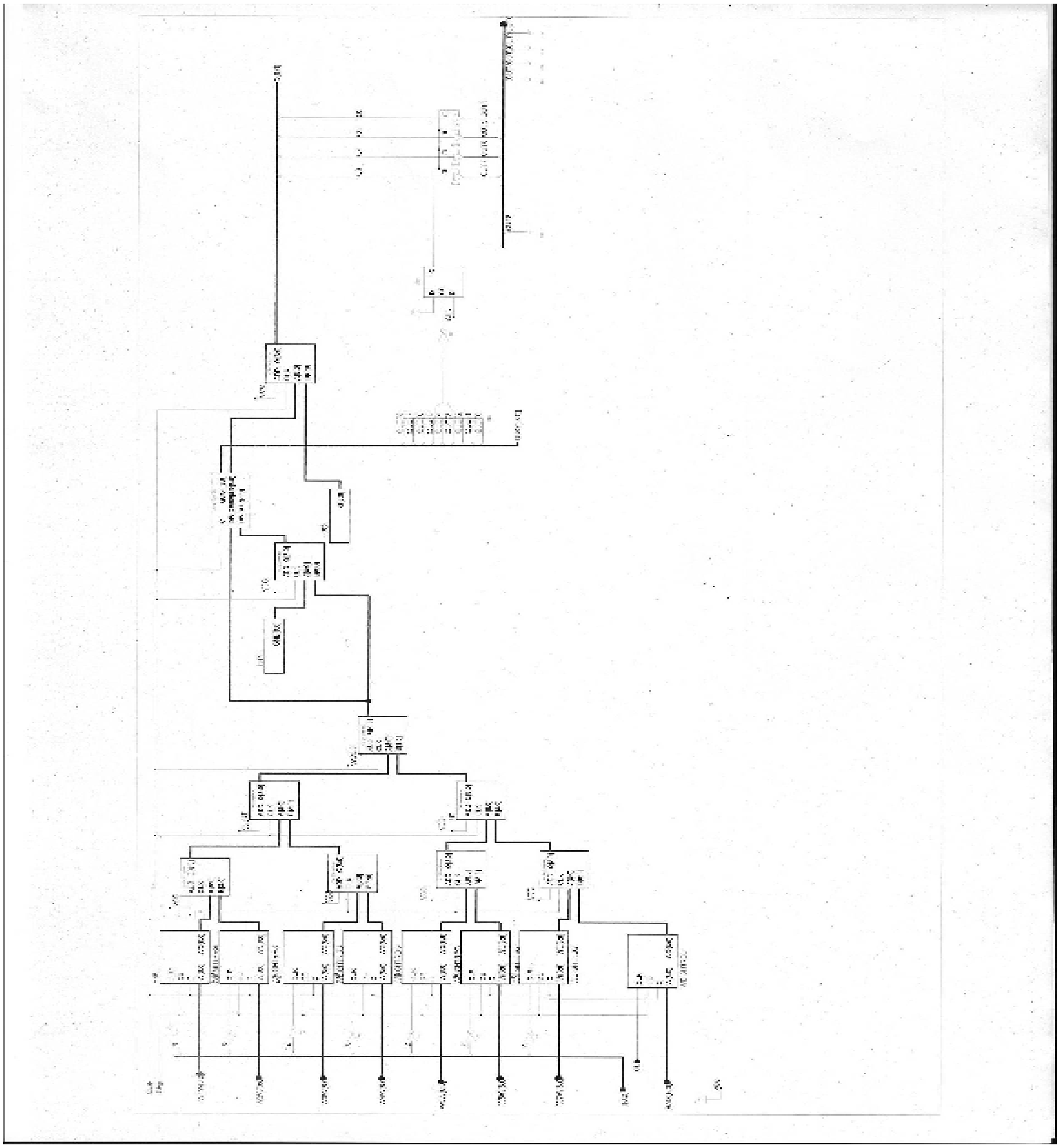


Figure (11): The design of single neuron in each layer

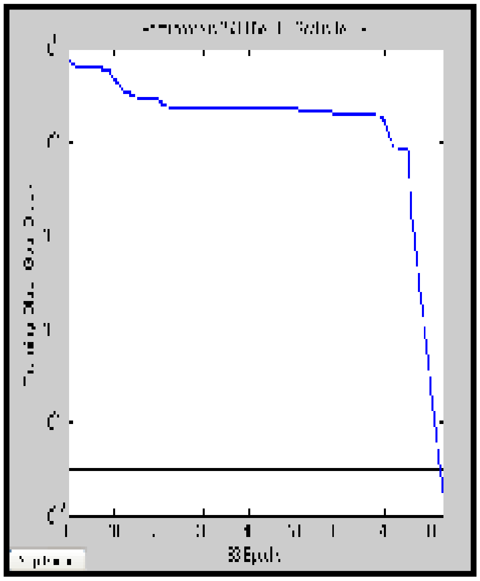


Figure (12): Process of training for Humiditv NN

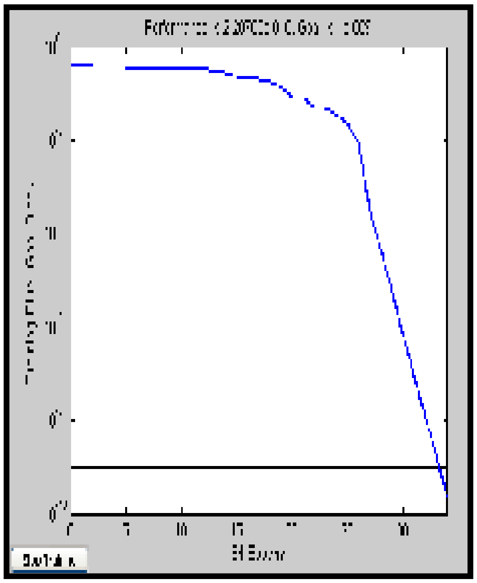


Figure (13): Process of training for PAR NN

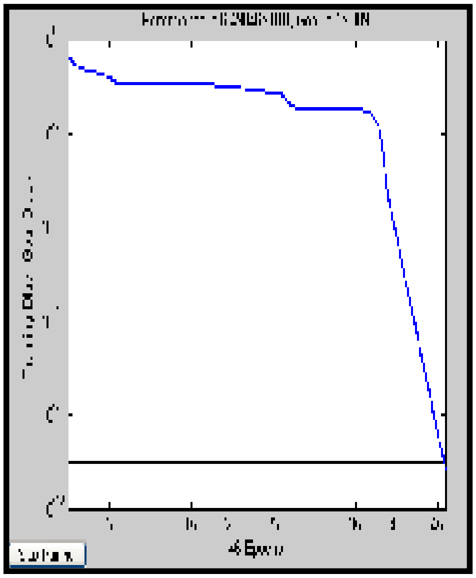


Figure (14): Process of training for temper. NN

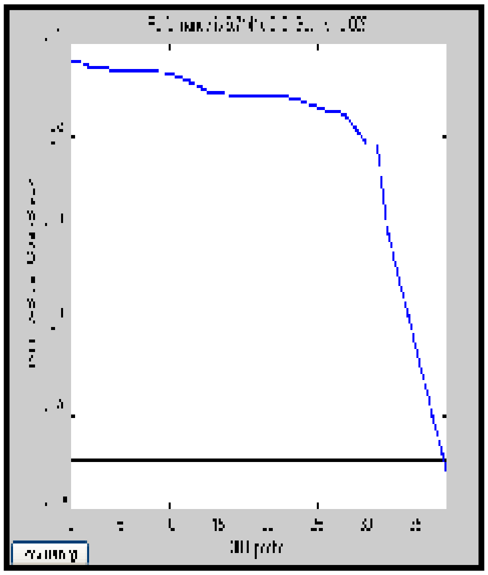


Figure (15): Process of training for TSR NN

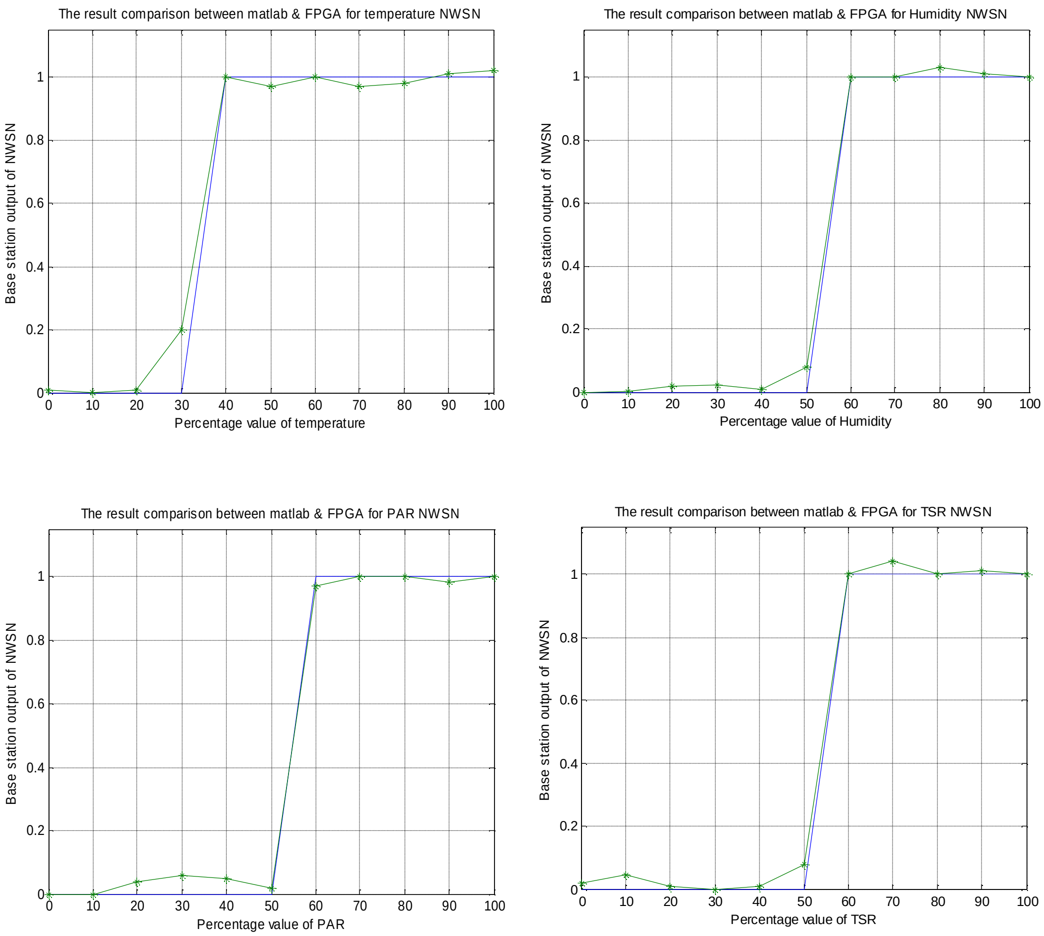
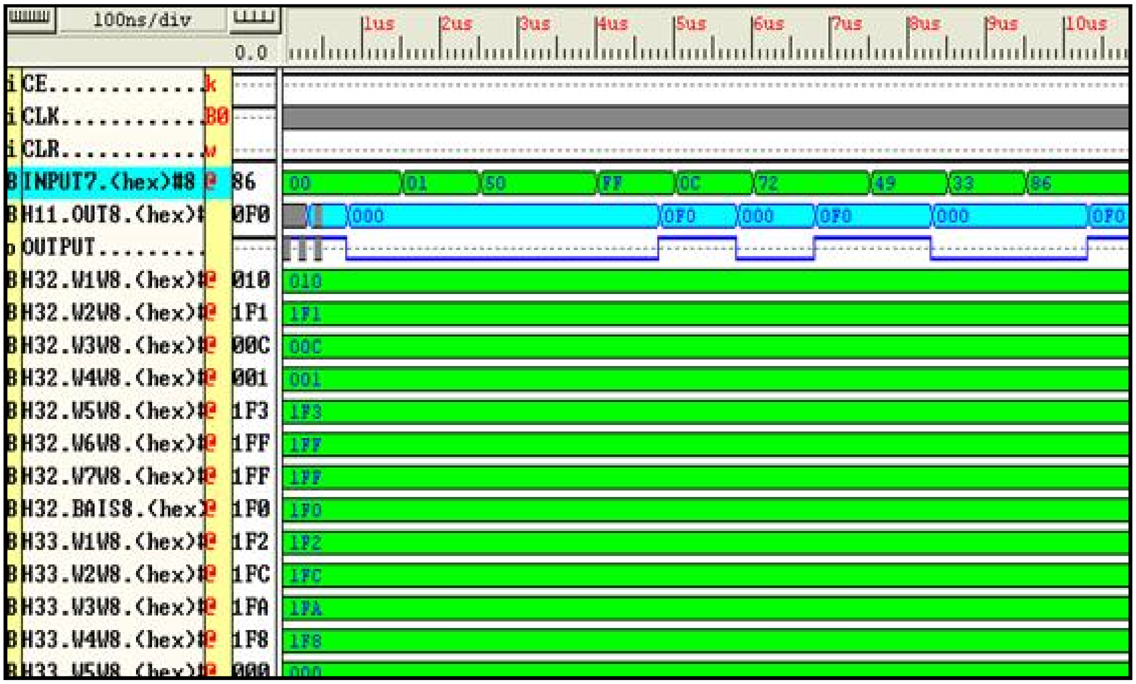
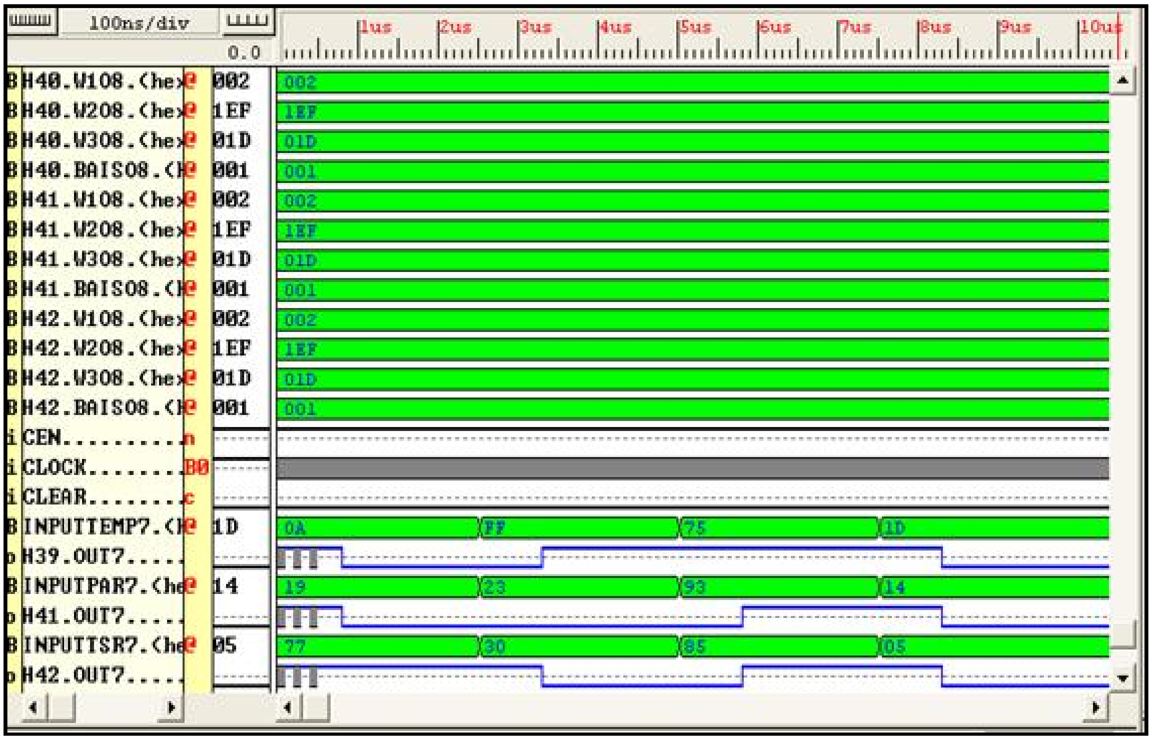


Figure (16) :The result comparison between MATLAB & FPGA



```
i CE............k
i CLK...........B0
i CLR...........w
B INPUT7.(hex)#8 @ 86     00 01 50 FF 0C 72 49 33 86
B H11.OUT8.(hex)#  0F0    000 0F0 000 0F0 000 0F0
o OUTPUT.........
B H32.W1W8.(hex)#@ 010    010
B H32.W2W8.(hex)#@ 1F1    1F1
B H32.W3W8.(hex)#@ 00C    00C
B H32.W4W8.(hex)#@ 001    001
B H32.W5W8.(hex)#@ 1F3    1F3
B H32.W6W8.(hex)#@ 1FF    1FF
B H32.W7W8.(hex)#@ 1FF    1FF
B H32.BAIS8.(hex)@ 1F0    1F0
B H33.W1W8.(hex)#@ 1F2    1F2
B H33.W2W8.(hex)#@ 1FC    1FC
B H33.W3W8.(hex)#@ 1FA    1FA
B H33.W4W8.(hex)#@ 1F8    1F8
B H33.W5W8.(hex)#@ 000    000
```

Figure (17) : Output of humidity sub-neural network



```
100ns/div
                        0.0
B H40.W108.(hex e  002
B H40.W208.(hex e  1EF
B H40.W308.(hex e  01D
B H40.BAIS08.(H e  001
B H41.W108.(hex e  002
B H41.W208.(hex e  1EF
B H41.W308.(hex e  01D
B H41.BAIS08.(H e  001
B H42.W108.(hex e  002
B H42.W208.(hex e  1EF
B H42.W308.(hex e  01D
B H42.BAIS08.(H e  001
i CEN..........n
i CLOCK.......B0
i CLEAR.......c
B INPUTTEMP7.(H e  1D
o H39.OUT7.....
B INPUTPAR7.(he e  14
o H41.OUT7.....
B INPUTTSR7.(he e  05
o H42.OUT7.....
```

Figure (18): The result of complete design of the base station of NWSN operation-timing diagram.