Indonesian Journal of Electrical Engineering and Computer Science
Vol. 19, No. 3, September 2020, pp. 1469~1479
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v19.i3.pp1469-1479

Design and implementation hamming neural network with VHDL

Liqaa Saadi Mezher
Department of Computer Engineering Al-Mustansiriyah University, Baghdad, Iraq

Article Info

Article history:

Received Oct 21, 2019
Revised Mar 15, 2020
Accepted Apr 1, 2020

Keywords:

Artificial neural network
hamming neural network
Feed forward layer
FPGA
Recurrent layer
VHDL

ABSTRACT

Hamming Neural Network is type of artificial neural network consist of two types of layers (Feed Forward Layers and Recurrent Layer). In this paper, two inputs of patterns in binary number were used. In the first layer, two neurons and pure line function were used. In the second layer, three neurons and positive line function were used. Also applied Hamming Neural networks algorithm in three simulation methods (Logical gate method, software program coding method and instant block diagram method). In this work in VHDL software program was used and FPGA hardware used.

Corresponding Author:

Liqaa Saadi Mezher,
Department of Computer Engineering,
AL Mustansiriyha University, Baghdad, Iraq.
Email: iqa35@uomustansiriyah.edu.iq

1. INTRODUCTION

Artificial neural networks (ANN) is a technique of artificial intelligence inspired by the human neurons system and is commonly used for modeling and improving the complex phenomena that involve a large number of process variables [1-3]. Artificial neural networks are computational methodologies that result in multiple analyzes factors [4, 5]. Inspired by biological nervous networks, artificial neural network containing layers of simple computing nodes that operate as organs summary nonlinear models of networks [6]. Field Programmable Gate Arrays (FPGA) device is a hardware implementation large number of artificial neural network [7-9]. Very High-Speed Integrated Circuits Hardware Description Language (VHDL) the software program [10] applied to humming neural network using many methods [8, 11-13].

The paper is arranged as in the following. In the Section 2 and displays a description briefly Hamming Neural Network about the parameters of patterns, weights matrix and the transfer function using (pure line in Feed Forward Layer and positive line in Recurrent Layers). In Section 3, displays the algorithm of Hamming Neural Network. In Section 4, Artificial Neural Network design using Hamming Neural Network with many simulation methods displays applied by VHDL software program and FPGA hardware. In Section 5, displays the result of Hamming Neural Network, last Section 6, are given some important conclusions and future work.

2. HAMMING NEURAL NETWORK

In this paper, the initial parameters of the humming neural network used, the architecture of neural network contains of many layers or many neural networks, as shown in Figure 1.

2.1. Feed forward layer

In the Feed Forward Layer FFL (Feed Forward neural networks) consist of two layers [14-16] also this layer two inputs were used with one output. Where the input of binary number, the first layer (input layer of feed forward): its takes the inputs and passes them to the input layer without doing anything.

The second layer (the output layer of feed forward): an a output layer, composed by n neurons, which processing n inputs, P1,…., Pn, and multiply with the weights matrix and composed by two neuron and combined with the bias and then entered into a linear activation function (Purelin (P)) and extracted the result (a1), in the FFL doesn't have the hidden layers, as shown in Figure 1(a).

2.2. Recurrent layer

In the Recurrent Layer RCL (recurrent layer neural networks) consist of three layers [17-20], the first layer (input layer of Recurrent Layer): In this layer the output from the previous layer (feed forward) was used and used as a primitive input with one output (initial a2=0). the second layer (the hidden layer of recurrent layer): the layer of feed forward became input to the hidden layer in recurrent layer, composed by n neurons [21], which processing n inputs, P1,….,Pn, and multiply with the weights matrix and composed by three neurons and combined with the bias (b2) and then entered into a positive linear activation function (Poslin (P)). And the third layer (the output layer of recurrent layer): the result (F) was extracted, as shown in Figure 1(b).

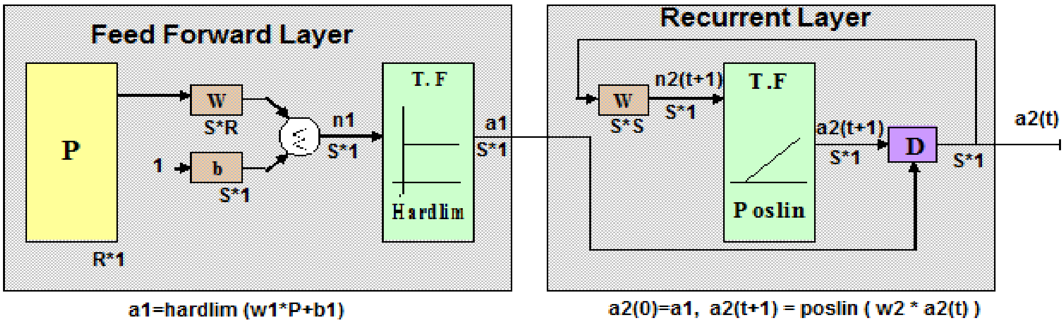

Figure 1. Block diagram of humming neural network (a) feed forward layer (b) recurrent layer

3. ALGORITHM OF HAMMING NEURAL NETWORK

Step 1: Set activation and initialize weights and bias values.
I: Number of input nodes (input vector of Patterns).
J: Number of length of Patterns.
Step 2: For each input vector P, N=1,2, …, I, M=1, 2…, J, do step (3 to 4).
Step 3: For each neuron, multiply each input by its corresponding weight and sum with bias values. It receives for the next step.
Step 4: Update the activation for N=1,2, …, I, M=1, 2…, J.

$$X_m(0) = F_{hardlim}(X_{(ni,m)})$$

Step 5: Apply activation function and save current activations in Xm (old) to be used in the next iteration for M=1, 2, …, j.

$$X_m(old) = X_m(0))$$

Step 6: choose an abs for the weight matrix (set 0 <absolute < 1/J).

$$\text{Weight:}\quad W_{n,m} = \begin{matrix} 1 & for\ n = m \\ -\epsilon & otherwise \end{matrix}$$

Step 7: For t=1,2,3, … repeat steps (8 to 11), while stopping condition is false.
Step 8: For each neuron, N=1,2,3, …, I, M=1, 2…, J, compute the net signal, it receives for the next step.

$$X_{(ni,m)}(new)) = W_{ni,m} * P_n + B_n$$

Step 9: Update the activations for N=1,2, …, I, M=1, 2…, J.

$$X_m(new) = F_{Poslin}(X_{(ni,m)}(new)))$$

Step 10: Save the activation for use in the next iteration.

$$X_m(old) = X_m(new))$$

Step 11: Test stopping condition. If more than one node has a nonzero output then go to step 8, otherwise, as shown in Figure 2.

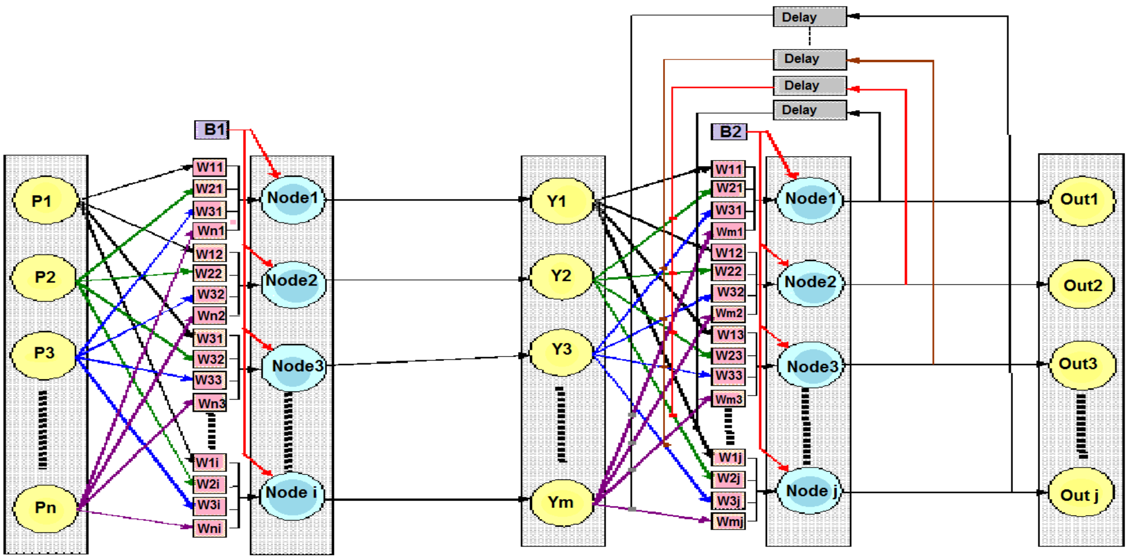

Figure 2. Architecture of humming neural network

4. SIMULATION METHODS FOR HUMMING NEURAL NETWORK

In this paper, 3 methods for programming were used using the VHDL program and using the FPGA device as shown below:

4.1. Logical gates method

The network is synthesized by connecting it using the NAND gate and inputs (P1, P2, W11, W12, W21, W22, W23, b1, b2) and outputs (F) are used, as shown in Figure 3.

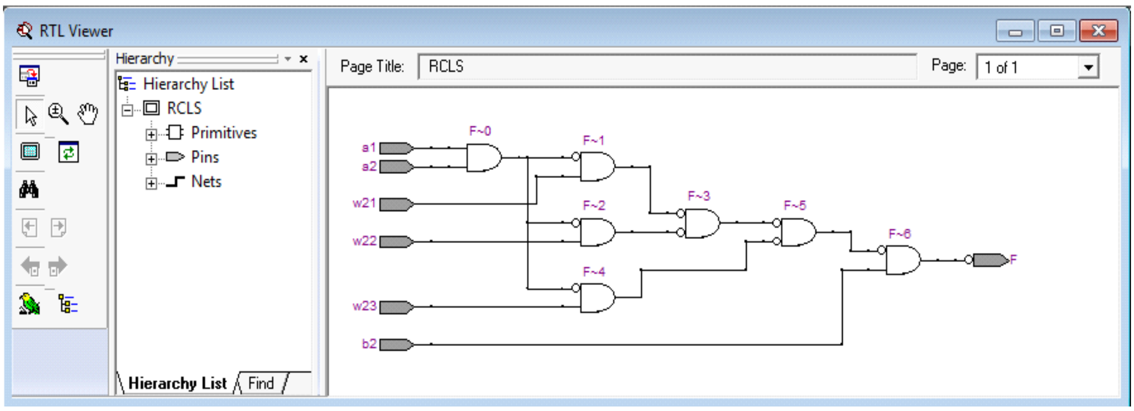

Figure 3. Hamming neural network using logic gate

4.2. Software program coding method

In this way the programming was used by code and divided the hamming neural network into layers:

4.2.1. Feed forward layer (FFL)

In this layer, programming was done using the entity to define the inputs (P1, P2, W11, W12, b1) and output (a1), and the Logic Function was used in the architecture to extract the output as in Figure 4.

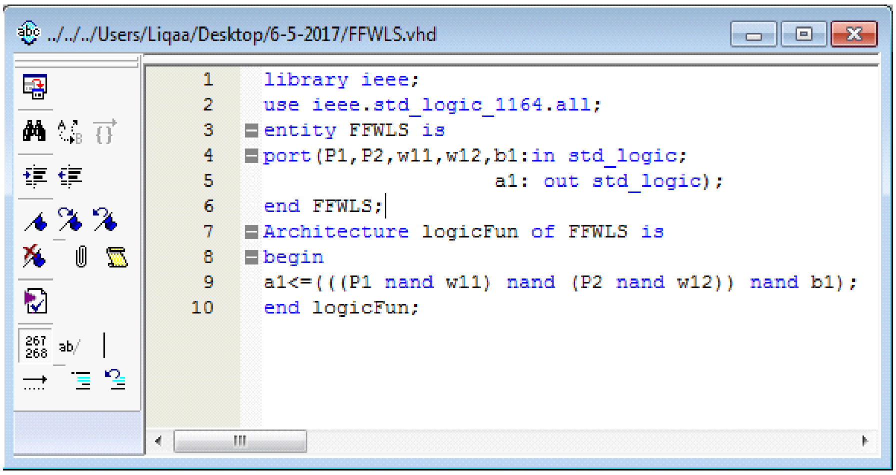

Figure 4. Feed forward neural network using code method

Summary: A summary review of program implementation and the special ratios of the network (FFL), as shown in Figure 5.

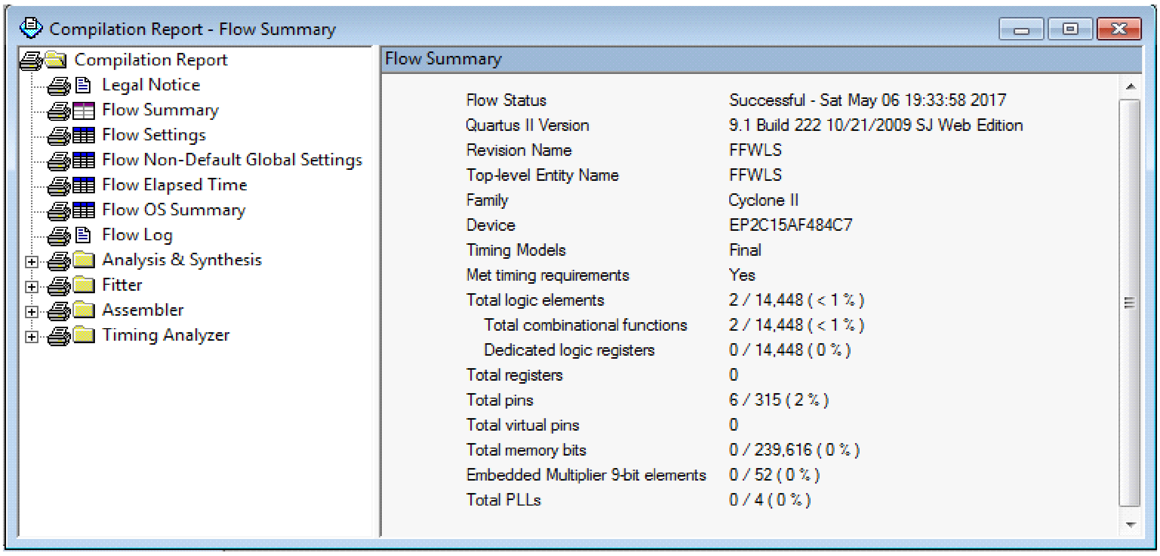

Figure 5. Summary for feed forward layer

RTU Viewer: To review the FFL network binding after execution by (select program – tools – Netlist viewers – RTL viewer), as in Figure 6.

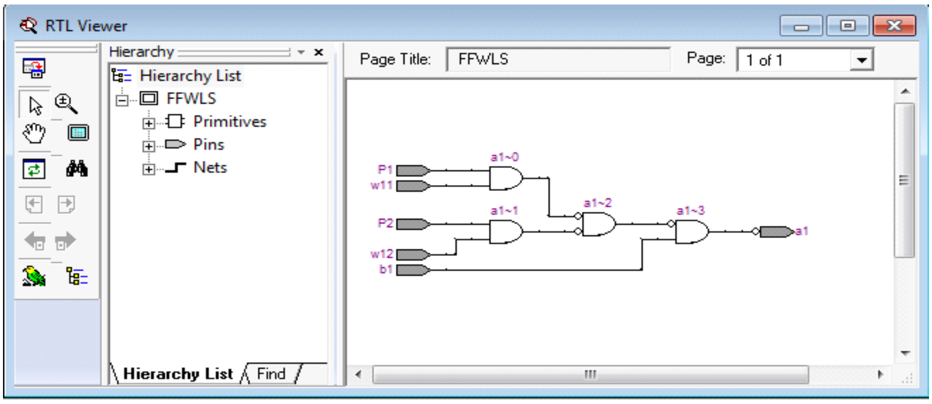

Figure 6. RTL viewer for feed forward layer

Instant Block Diagram: To view the network binding after execution the form Block Diagram is done by (select program – tools – Netlist viewers – Technology Map Viewer – Post Mapping), as in Figure 7.

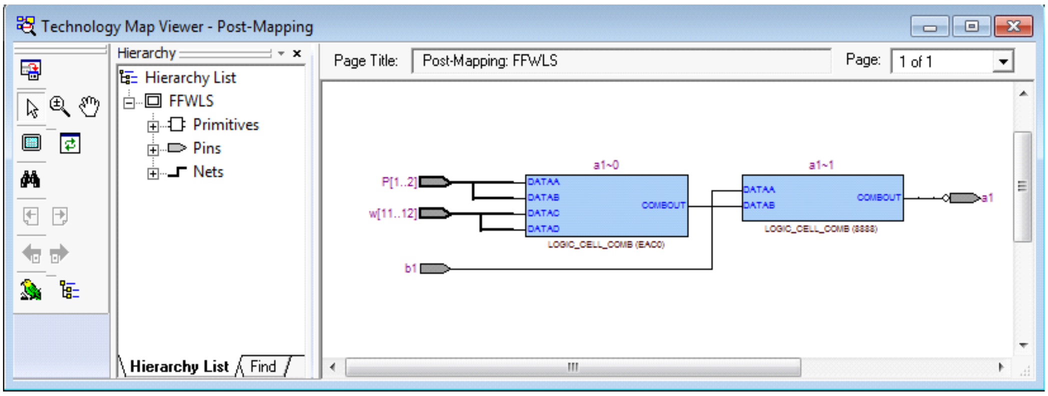

Figure 7. Instant for feed forward layer

4.2.2. Recurrent Neural Network (RCL)

In this layer, programming was done using the entity to define the inputs (a1, a2, W21, W22, W23, b2) and output (F), and the Logic Function was used in the architecture to extract the output as in Figure 8.

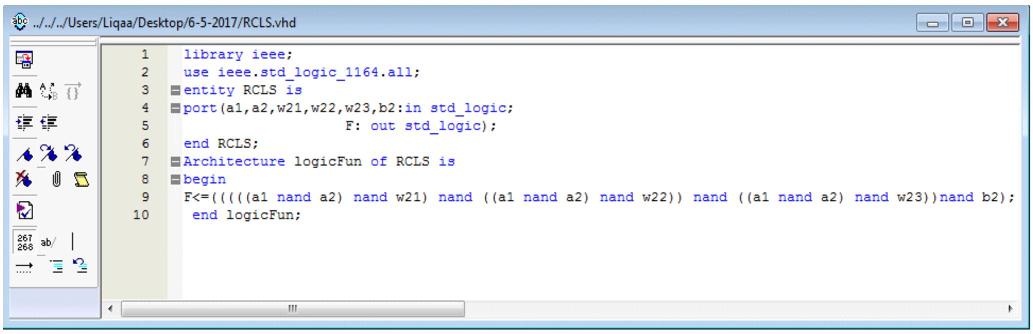

Figure 8. Feed forward neural network using code method

Summary: A summary review of program implementation and the special ratios of the network (RCL), as shown in Figure 9.

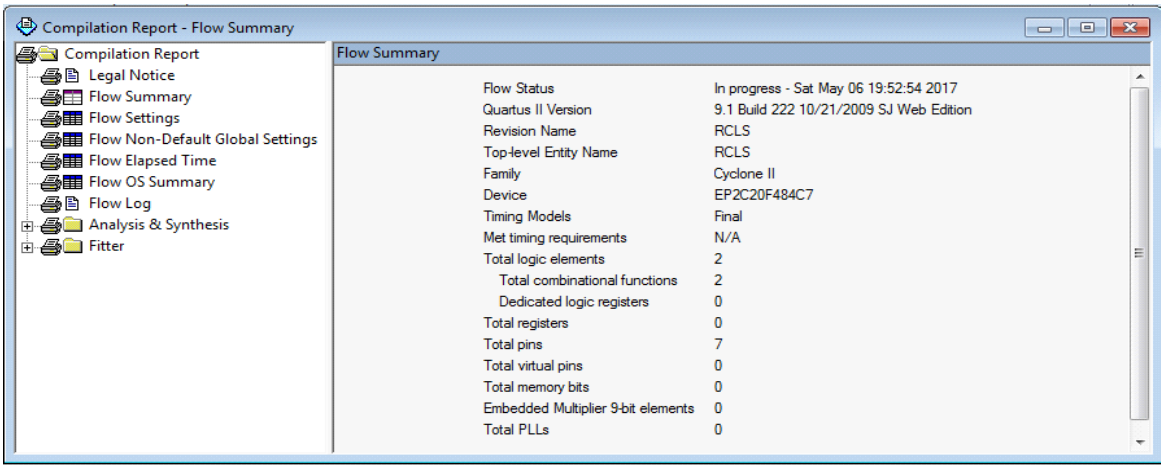

Figure 9. Summary for recurrent layer

RTU Viewer: To review the RCL network binding after execution by (select program – tools – Netlist viewers – RTL viewer), as in Figure 10.

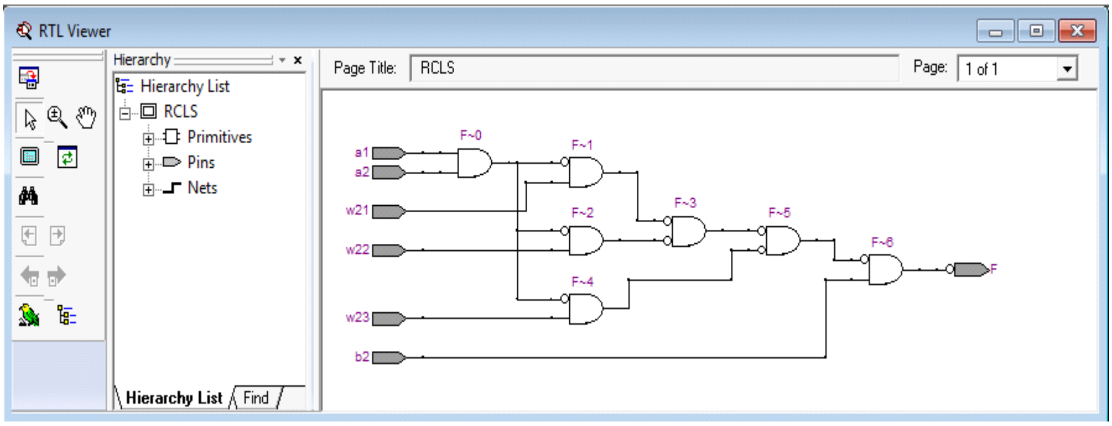

Figure 10. RTL viewer for recurrent layer

Instant Block Diagram: To view the RCL network binding after execution the form Block Diagram is done by (select program – tools – Netlist viewers – Technology Map Viewer – Post Mapping), as in Figure 11.

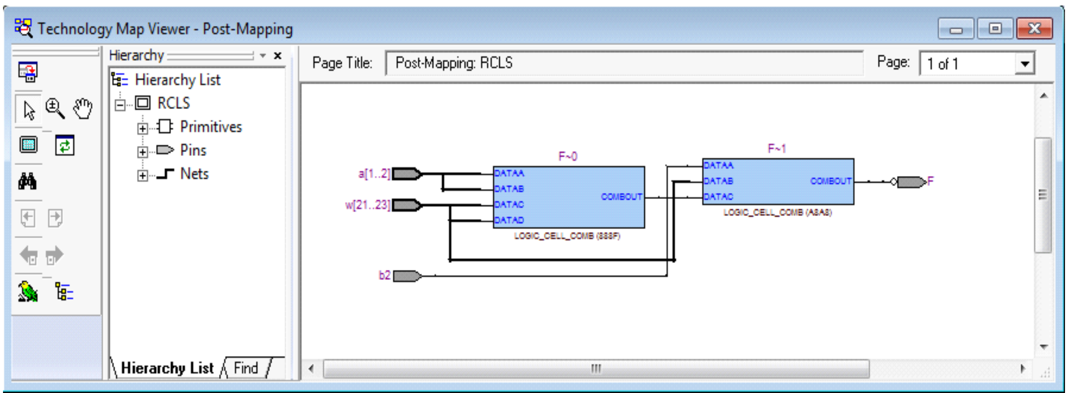

Figure 11. Instant for recurrent layer

4.2.3. Hamming Neural Network (HNN)

It is the main program of the hamming neural network where the programs [22] are called for the two layers (FFL & RCL) by the component and then merge them and define the input and output and the use of the structure in architecture and extraction of output [23-25], as in Figure 12. And then convert from a software programmable to Block Diagram and implement it on the FPGA machine as a as a practical and extract the results.

HNNSS.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY work;

ENTITY hnns IS
        PORT
        (
                P1  :  IN   STD_LOGIC;
                P2  :  IN   STD_LOGIC;
                W11  :  IN   STD_LOGIC;
                W12  :  IN   STD_LOGIC;
                B1  :  IN   STD_LOGIC;
                W21  :  IN   STD_LOGIC;
                W22  :  IN   STD_LOGIC;
                W23  :  IN   STD_LOGIC;
                b2  :  IN   STD_LOGIC;
                a1  :  OUT  STD_LOGIC;
                F  :  OUT  STD_LOGIC
        );
END hnns;

ARCHITECTURE bdf_type OF hnns IS

COMPONENT ffwls
        PORT(P1 : IN STD_LOGIC;
                 P2 : IN STD_LOGIC;
                 w11 : IN STD_LOGIC;
                 w12 : IN STD_LOGIC;
                 b1 : IN STD_LOGIC;
                 a1 : OUT STD_LOGIC
        );
END COMPONENT;

COMPONENT rcls
        PORT(a1 : IN STD_LOGIC;
                 a2 : IN STD_LOGIC;
                 w21 : IN STD_LOGIC;
                 w22 : IN STD_LOGIC;
                 w23 : IN STD_LOGIC;
                 b2 : IN STD_LOGIC;
                 F : OUT STD_LOGIC
        );
END COMPONENT;

SIGNAL  SYNTHESIZED_WIRE_0 :  STD_LOGIC;
SIGNAL  SYNTHESIZED_WIRE_1 :  STD_LOGIC;


BEGIN
a1 <= SYNTHESIZED_WIRE_0;
F <= SYNTHESIZED_WIRE_1;



b2v_inst : ffwls
PORT MAP(P1 => P1,
                 P2 => P2,
                 w11 => W11,
                 w12 => W12,
                 b1 => B1,
                 a1 => SYNTHESIZED_WIRE_0);


b2v_inst1 : rcls
PORT MAP(a1 => SYNTHESIZED_WIRE_0,
                 a2 => SYNTHESIZED_WIRE_1,
                 w21 => W21,
                 w22 => W22,
                 w23 => W23,
                 b2 => b2,
                 F => SYNTHESIZED_WIRE_1);


END bdf_type;
```

Figure 12. Represent coding for hamming neural network

4.3. Instant block diagram method

After completing the programming process for each layer separately, take the following steps to convert the software program to block Diagram.

4.3.1. The first layer FFL

Select all software program of FFL - File - Creat/Up date - Create Symbol File for Current File, and open new file - block Diagram/Schematic File – Ok – then choose instant block from project, as shown in Figure 13.

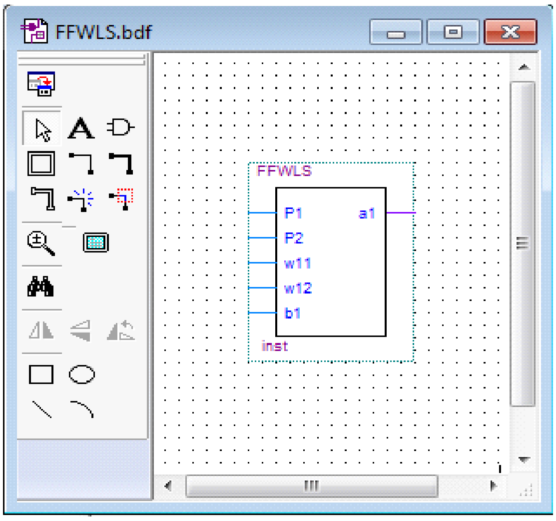

Figure 13. Represent instant block diagram for FFL

4.3.2. The second layer RCL

Select all software program of RCL - File - Creat/Up date - Creat Symbol File for Current File, and open new file - block Diagram/Schematic File – Ok – then choose instant block from project, as shown in Figure 14.

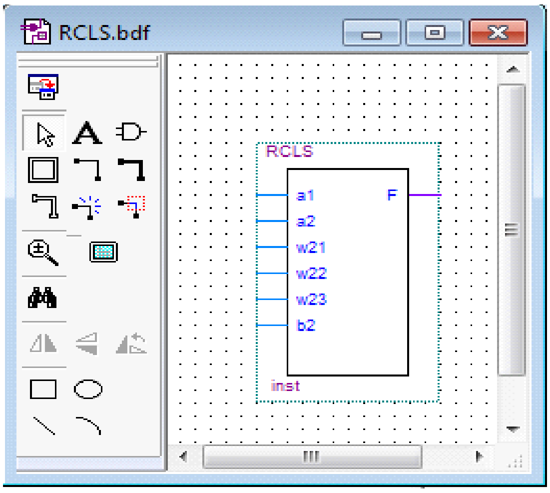

Figure 14. Represent instant block diagram for RCL

In Figure 15, the previous two instant block diagrams are integrated, and integrated network, the main input representation, the weight matrix and its main output.

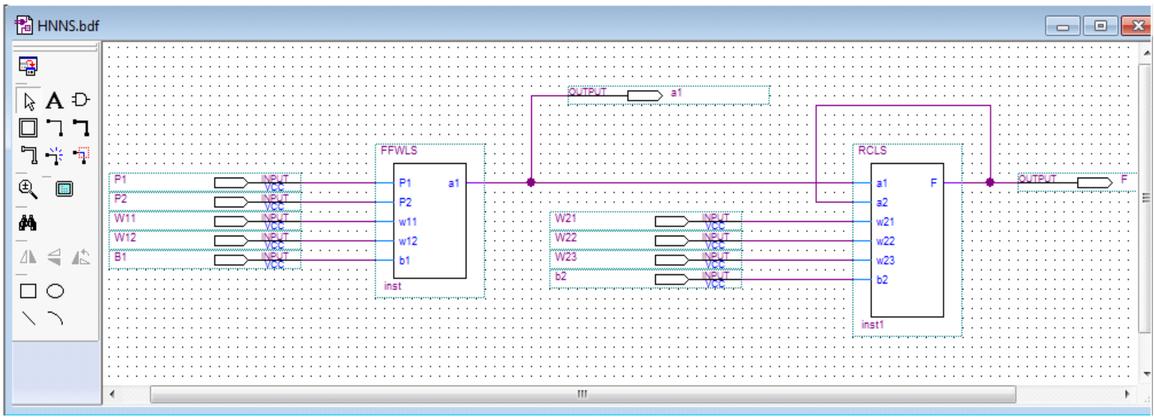

Figure 15. Represent instant block diagram for HNN

In the FPGA device, 9 pins were used (9) inputs representing (2 patterns +5 weights + 2 biases) as shown Table 1 represents the used pins. And two outputs were used. One of which represents is the extraction of the interlayer from the first layer and the other represents the final output of the main network as shown Table 2 represents the used pins.

Table 1. Represent the input pins in hamming neural network

Switch	Pins	Represented
SW0	PINS_L22	B1
SW1	PINS_L21	B2
SW2	PINS_M22	P1
SW3	PINS_V12	W11
SW4	PINS_W12	P2
SW5	PINS_U12	W12
SW6	PINS_U11	W21
SW7	PINS_M2	W22
SW8	PINS_M1	W23

Table 2. Represent the output pins in hamming neural network

LED	Pins	Represented
LEDR1	PINS_R19	A1
LEDR0	PINS_R20	F

5. THE RESULT
5.1. The input
The Input values for patterns and weights matrix in layers FFL and RCL, as shown Table 3.

Table 3. Represent the input values in hamming neural network

Patterns		Weights Matrix in FFL		Weights Matrix in RCL		
P1	P2	W11	W12	W21	W22	W23
0	0	0	0	0	0	0
0	1	0	1	1	0	0
1	0	1	0	0	1	0
1	1	1	1	1	1	0
0	0	0	0	0	0	1
0	1	0	1	1	0	1
1	0	1	0	0	1	1
1	1	1	1	1	1	1

5.2. The output
- when b1 and b2=0

  a1= 11111111

  F = 11111111
- when b1=1 and b2=0

  a1= 00010001

  F= 11111111
- when b1=0 and b2=1

a1= 11111111
F= 00111111

– when b1=1 and b2=1
a1= 00010001
F= 00000000

6. CONCLUSIONS

In this paper hamming neural network applied. This network content of two types of network FFL & RCL. In FFL layer two patterns (P1, P2) used with one output (a1). Where the input of patterns binary number was multiplied with the weight's matrix (w11, w12) and combined with the baise (b1) and then entered into the pure line functions. In the RCL layer, the output from the previous layer (feed forward) was used and used as a primitive input (a1 and initial feedback a2=0) with one output (F). Where the input was multiplied with the weight matrix (W12, W22 and W23) and combined with the baise (b2) and then entered into the pos line function. And three simulation methods (Logical gate method, software program coding method and instant block diagram method) applied using VHDL program and connected to FPGA hardware to extract the final result.

REFERENCES

[1] Ibrahim M. N., Mohammed O. A. and Samy S. A., "Artificial Neural Network for Diagnose Autism Spectrum Disorder", *International Journal of Academic Information Systems Research (IJAISR)*, ISSN: 2000-002X, vol. 3 no. 2, pp. 27-32, 2019.
[2] Binghui W. and Tingting D., "A Performance Comparison of Neural Networks in Forecasting Stock Price Trend", *International Journal of Computational Intelligence Systems*, vol. 10, pp. 336 – 346, 2017.
[3] Osama M. A., Bassem S. A. and Samy S. A., "Predicting Overall Car Performance Using Artificial Neural Network", *International Journal of Academic and Applied Research (IJAAR)*, ISSN: 2000-005X, vol. 3, no. 1, pp. 1-5, 2019.
[4] Mohd N. E., Mohd I. and Kim S. C., "Artificial neural network and partial least square in predicting blood hemoglobin using near-infrared spectrum", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 701-708 ISSN: 2502-4752, 2019.
[5] Oludare I. A., Aman J., Abiodun E. O., Kemi V. D., Abubakar M., Okafor U. L., Humaira A., Abdullahi A. K., Usman G. And Muhammad U. K., "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition, IEEE, vol. 7, ISSN: 2169-3536, pp. 158820 – 158846, 2019.
[6] Joko S., Anton S. P., Azizi A. and Bahari I., "A linear Model Based on Kalman Filter for Improving Neural Network Classification Performance", *Journal of Expert Systems with Applications*, vol. 49, pp. 112–122, 2016.
[7] Dinda P., Takashi M. and Hakaru T.," Implementation of Multi-FPGA Communication using Pulse-Coupled Phase Oscillators", International Conference on Artificial Life and Robotics (ICAROB), Jan 19-22, PP 128-131, Seagaia Convention Center, Miyazaki, Japan, 2017.
[8] Nasri S., Zeyad A. O., Mohammad H. M., and Mohd N. H. "Design and Implementation of FPGA-Based Systems - A Review", *Journal of Australian Journal of Basic and Applied Sciences*, vol. 3, no. 4, 2009.
[9] A. Muthuramalingam, S. Himavathi and E. Srinivasan," Neural Network Implementation Using FPGA: Issues and Application", *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 2, no.12, 2008.
[10] Km M. and A. K. Gautam,"Artificial Neural Network architecture and hardware Chip Implementation using VHDL", *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 5, no. 3, pp. 1245, 2019.
[11] J. Steckert and A. Skoczen, "Design of FPGA-based radiation tolerant quench detectors for LHC", *Journal of Instrumentation*, vol. 12, 2017.
[12] J M Vilardy, Leiner Barba J., C O Torres, "Design and implementation in VHDL code of the discrete cosine transform over finite fields for symmetric convolution operation", *Journal of Physics*, vol. 12, no. 19, 2019.
[13] Mohammed Y. H., and Ahmed A. M., "A Spiking Neural Network Controller Design with VHDL Based on SSO Algorithm for Inverted Pendulum", *International Journal of Engineering Science and Computing*, vol. 9, no.1, 2019.
[14] Pierre B. and Roman V., "The capacity of feed forward neural networks", *Journal of Neural Networks*, vol. 116, pp. 288-311, 2019.
[15] Grzegorz D., "Data-Driven Randomized Learning of Feed forward Neural Networks", *Journal of Computer Science*, August, 2019.
[16] Leandro D. M., Taras I., Jose V. F. V., Manuel B. M. And Alfredo R. M., "A Novel Systolic Parallel Hardware Architecture for the FPGA Acceleration of Feed forward Neural Networks", *IEEE*, vol. 7, 2019.
[17] Stefan H., Guillaume L. and Guy W.," Internal representation dynamics and geometry in recurrent neural networks", *Journal of Computer Science*, 2020.
[18] Sangeun K. and Juhan N.," Joint Detection and Classification of Singing Voice Melody Using Convolutional Recurrent Neural Networks", *Journal of Applied Sciences*, vol. 9, pp. 1324, 2019.
[19] Jie W., Jun W., Wen F. and Hongli N., "Financial Time Series Prediction Using Elman Recurrent Random Neural Networks", Computational Intelligence and Neuroscience, pp. 14, 2016.

[20] Qi Z., Xiaorui Z., Yuanyuan M., Zhe Y. and Ke X., "Recurrent neural networks made of magnetic tunnel junctions", *Journal of AIP Advances*, vol. 10, pp. 025116, 2020.
[21] Siti A. M., Muhaini O. and Mohd H. A.," A review on data clustering using spiking neural network (SNN) models", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 3, pp. 1392-1400, ISSN: 2502-4752, 2019.
[22] Engin P. and Selin S. K., "A Comprehensive Review for Artificial Neural Network Application to Public Transportation," vol. 35, no. 1, pp. 157-179, 2017.
[23] V S Klimov, A S Klimov and S V Mkrtychev, "Computer diagnostics of resistance spot welding based on Hamming neural network", *Journal of Physics*, vol. 13, no. 33, 2019.
[24] O.I. Khristodulo, A. A. Makhmutova and T.V. Sazonova, "Use algorithm based at Hamming neural network method for natural objects classification", *Journal of Computer Science*, vol. 103, pp. 388 – 395, 2017.
[25] S. Archana, B. K. Madhavi and Murlikrishna, "Design Hamming Neural Network for Real Time 0-9 Digit Recognition on HSPICE Synopsys Tool ", *International Journal of Electronics Communication and Computer Engineering*, vol. 8, no. 6, ISSN 2249–071X, 2017.

BIOGRAPHY OF AUTHOR

Liqaa Saadi Mezher, Master Sc. in Computer Engineering, B. Sc. in Computer & Software Engineering, She has 14 years of teaching experience. She published 6 research papers at International level. Worked in Al-Mustansiriyah University, Baghdad, Iraq.