DESIGN OF A HARDWARE TWO-LAYER PERCEPTRON NEURAL NETWORK USING FIELD PROGRAMMABLE GATE ARRAYS (FPGAS) FOR SPEECH RECOGNITION [+]

تصميم كيان مادي لشبكة عصبية ثنائية الطبقة نوعPerceptron باستخدام البوابات المنطقية القابلة للبرمجة الحقلية لتمييز المقاطع الكلامية

Ahmad Kussay Kayali[*] Maan Mohamad Shikr [**] Mazin Rejab Khalil[**]

Abstract:

Multi-layer perceptron neural networks (MLPNN) have been used in many applications in science and engineering. Real time applications necessitate the hardware implementation of MLPNN. This paper introduces a method to design a two-layer perceptron neural network using field programmable gate arrays (FPGAs). The different modules composing the system are designed using very high speed description language (VHDL) and assembled hierarchically. Fixed point numbers format is used to avoid data width inflation due to successive multiplications and additions. Piece wise second-order approximation of the sigmoid function is adopted. Xilinx ISE 9.2i software was used to develop and simulate the VHDL modules. The designed neural network is configured on Xilinx Spartan 3e starter kit, its functionality is tested by applying it as a discriminating unit in a certain speech recognition system.

المستخلص:

إن للشبكات العصبية متعددة الطبقات نوع Perceptron تطبيقات واسعة في المجالات الهندسية والعلمية. إن التطبيقات العملية في الزمن الحقيقي لهذه الشبكات تتطلب وضع بُنية هذه الشبكات في مجال الكيان المادي. في هذا البحث تم تصميم شبكة عصبية ثنائية الطبقة نوع Perceptron ذات الانتشار الأمامي وتنفيذها على شريحة مصفوفة البوابات القابلة للبرمجة الحقلية (FPGAs). إن المكونات المختلفة لهذه الشبكة صُممت باستخدام لغة وصف الكيان المادي (VHDL) ورُكبت بطريقة متسلسلة. تم استخدام نظام الأرقام الثابتة لمنع تضخم سعة الأرقام بسبب العمليات الرياضية المتعاقبة. وصف عمل وحدة دالة التفعيل نوع Sigmoid باستخدام معادلات من الدرجة الثانية. إن تصميم النظام ومكوناته ونتائج المحاكاة أُجريت في البيئة البرمجية Xilinix ISE 9.2i ونفذت الشبكة على شريحة نوع Spatran 3e.ولاختبار كفاءة أداء الشبكة العصبية المصممة تم استخدامها تجريبيا كجزء مميز في أحد أنظمة تمييز المقاطع الصوتية.

Introduction:

Artificial neural networks (ANNs) have been mostly implemented in software. This has benefits since the user need not know the inner operations of neural network elements, but

[+] **Received on 17/2/2010 , Accepted on 23/2/2011 .**
[*] **Prof/Faculty of Electrical and Electronics Engineering. University of Aleppo.**
[**] **Assist Prof./Technical College of Mosul .**

concentrates only on its application. In real time application of software-based ANNs, the execution is slower compared with hardware-based ANNs. Also the increasing demands for real time operation in the wide range fields of ANNs applications necessitated the search for high performance implementation that meets the computational requirements of the ANNs. Field programmable gate arrays (FPGAs) provide a suitable implementation platform for this type of performance due to its high flexibility and adaptability with the system design requirements[1].

Very high speed hardware description language (VHDL) is widely used in designing systems modules as a tool suite to facilitate FPGAs systems design.

Table.1 exhibits a previously performed works to design two types of the most outstanding neural networks. The type of the neural network, the number of the layers, the number of neurons and the type of the application which depends mainly on the resources of the available FPGAs slice. The discrimination capability of the neural networks that are used in speech recognition depends on the number of the features that are used to characterize each spoken syllable, therefore the more is the number of layers and neurons, the better is the recognition efficiency.

The target of the research is to design a two-layer perceptron neural network with the following conditions

- It consists of 32 neurons in the input layer, 15 neurons in the hidden layer and 7 neurons in the output layer to cope with high discrimination capability.
- The designed network is to be configured on FPGAs slice of Spartan 3e kit, a special attention should be paid to the resources available in the slice.
- To accommodate with the resources available in the FPGAs slice the architecture must be performed in serial mode in stead of parallel mode and internal block Random Access Memory(BRAM) will be used in stead of external Static Random Access Memory(SRAM).
- The performance of the configured neural network is to be tested by applying it as a discriminating part in a certain speech recognition system.

The design procedure implies partitioning the two layers MLP into several modules that are connected hierarchically to construct the network.

Table(1). Different types of designed NN compared with the suggested work

Reference Number	*Type of Neural Network*	*Number of Layers*	*Number of Neurons*	*Activation Function*	*Storage Element*	*Application*
1	**two – layer perceptron (parallel MLP)**	**2**	**22-24-9**	**tan-sigmoid**	**look up tables**	**speech recognition**
2	**two – layer perceptron (serial)**	**2**	**2-3-1**	**tan-sigmoid**	**look up tables**	**-**
3	**kohenen**	**two dimensional**	**2 × 32**	**neighborhood function**	**SRAM**	**speech recognition**
4	**kohenen**	**two dimensional**	**25 × 64**	**neighborhood function**	**SRAM**	**speech recognition**
5	**kohenen**	**two dimensional**	**20 × 20**	**neighborhood function**	**SRAM**	**speech recognition**
suggested work	**two – layer perceptron**	**2**	**32 × 15 × 7**	**log – sigmoid**	**block ram**	**speech recognition**

Multi-layer perceptron neural network (MLPNN)

The multi-layer perceptron is a feed forward neural network with several layers; the input layer which is simply an input vector, some hidden layers and an output layer. The architecture of the suggested two-layer MLPNN is shown in figure (1).[1].

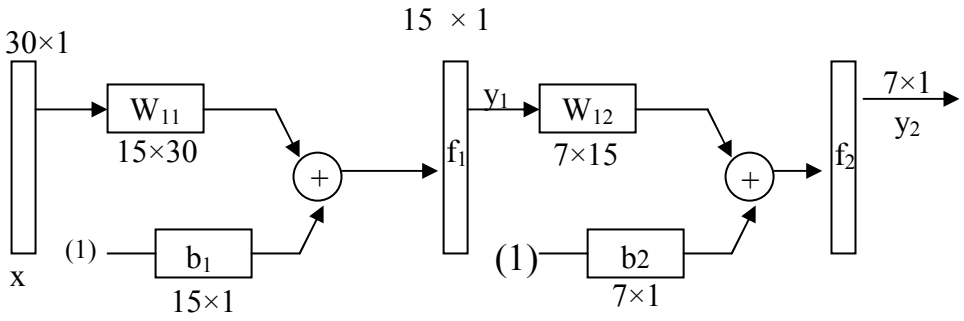

Figure (1): The architecture of two-layer perceptron neural network

The computation of the output of each layer is given by the following equations[6].

$$y_1(i) = f_1(b_1(i) + \sum_{j=1}^{30} W_{11}(i,j)x(j)).....(1)$$

$$y_2(k) = f_2(b_2(k) + \sum_{j=1}^{15} W_{12}(k,j)y_1(j)).....(2)$$

Where:
f_2 , f_1 are log-sigmoid functions
i = 1 , 2 … 15
k = 1, 2 … 7

The configuration shown in figure (1) can be used to discriminate a spoken syllable in speech recognition systems where the input vector (X) represents the features of the spoken syllables, each syllable is characterized by (30) features. The neural network is designed to discriminate seven syllables, each syllable is to be uttered five times, therefore the vector (X) can be repeated 35 times or described in the form of (30*35), 35 columns, each column contains 30 features of a syllable. The features representing a spoken syllable can be obtained using wavelet packet decomposition (WPD) or discrete wavelet transform of the spoken syllable signal [7,8].

According to [9] a software speech recognition system is constructed where each spoken syllable is analyzed using WPD to obtain it's features, software two-layer MLP is built and trained on the extracted features. The resulting weights of the two layers are applied on equations (1) and (2) for the purpose of discrimination.

Hardware Implementation of MLPNN:

The hardware implementation of the two-layer perceptron neural network whose architecture is shown in figure (1) depends mainly on realizing equations (1) and (2) to be mapped on the field programmable gate arrays (FPGAs). Figure (2) shows the block diagram of the hardware implementation procedure and the modules composing the system. Each layer consists of RAM, Rom, multipliers, accumulator, log-sigmoid activation function and controller modules.

RAM1 is a random access memory that is used to write the features of the spoken syllable under test, it stores 32 items; (30) items represent the features and the thirty-first item is given a value of one to be multiplied with the bias (b) and the thirty second item is given a value of zero to coincide with the time of the accumulator reset pulse in order not to lose any information when resetting the accumulator.

Rom1 is a read only memory that is used to store the weights of the first layer W_{11} whose size is (15 × 32). The number (15) represents the number of the first layer output nodes. The number 32 represents 30 weights value, bias value for each node and zero value to accommodate with accumulator resetting, the bias is considered here as a weight with input node having a value of (1).

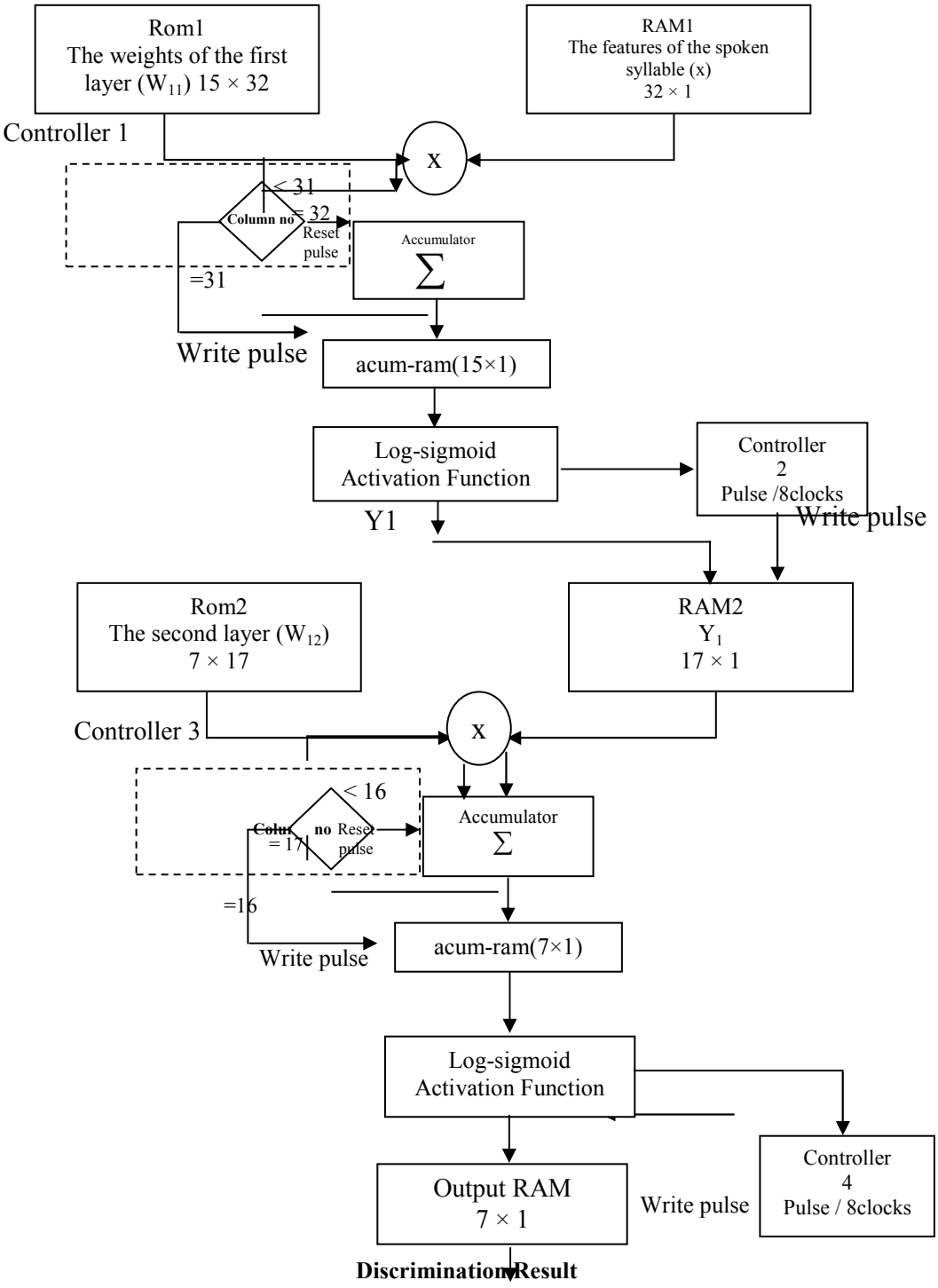

Figure (2): flowchart of a two-layer perceptron neural network hardware implementation on FPGAs.

The multiplier and accumulator modules perform the multiplication and accumulation operations of equation (1) row by row. At the end of each row(30^{th} column) the accumulator output must be written on the accumulator RAM (acum-ram) module, this must occur at pulse

number 31, then the accumulator is reset at pulse number 32 in order to start the operation on the 2nd row of W_{11}, and so on for the fifteen rows.
Controller module provides the write pulse at clock (31) and reset pulse at clock (32) to enable the writing process of the acum-ram module and resetting the accumulator.
The log-sigmoid activation function module is used to perform the log-sigmoid computation, it takes 8 clocks for each accumulator output, therefore; controller 2 provides a write pulse each 8 clocks to the RAM2 module to enable it to write the activation function output. Controller 2 is activated once the log-sigmoid module start computation. The same discussion applies for the 2nd layer.
An important consideration for implementing an ANN on FPGAs is the arithmetic representation format, the successive multiplication and addition operations causes an inflation in the bit width of each number i.e. multiplying two numbers each with 32 bits will result in 64 bits number; therefore, fixed point numbers with 32 bits is used. The format of the 32 bits fixed point numbers is shown in figure (3). Truncation to 32 bits is performed after each multiplication process.

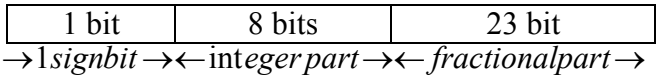

Figure (3) The format of 32 bits fixed point numbers

Each module shown in figure (2) is designed using VHDL language[10]. The RTL schematic diagram for the main modules will be demonstrated step by step throughout the research.

1-Random Access Memory (RAM) Module

The input samples (x), the results of the accumulations and the results of log-sigmoid calculations need to be stored in storage elements such as random access memories. As the processing speed is an essential factor, a dual port RAM is designed, such that memory read operation can be arranged to be a number of clocks (it can be one clock) behind the write operation. Five types of dual port RAM modules were designed and used in the system as shown in table (2).

Table (2) RAM modules

RAM title	Dimension row × column	Description
RAM1	32 × 1	store the input samples
acum-ram1	15 × 1	store the results of the 1st layer accumulations
RAM2	17 × 1	store the results of the 1st layer log-sigmoid computations
acum-ram2	7 × 1	store the results of the 2nd layer accumulations
output RAM	7 × 1	2nd layer log-sigmoid computations

Figure (4) shows the register transfer logic (RTL) schematic diagram of a dual port RAM (acum-ram) as a sample of the designed RAMs, with a capacity of (16 sample × 64bits).

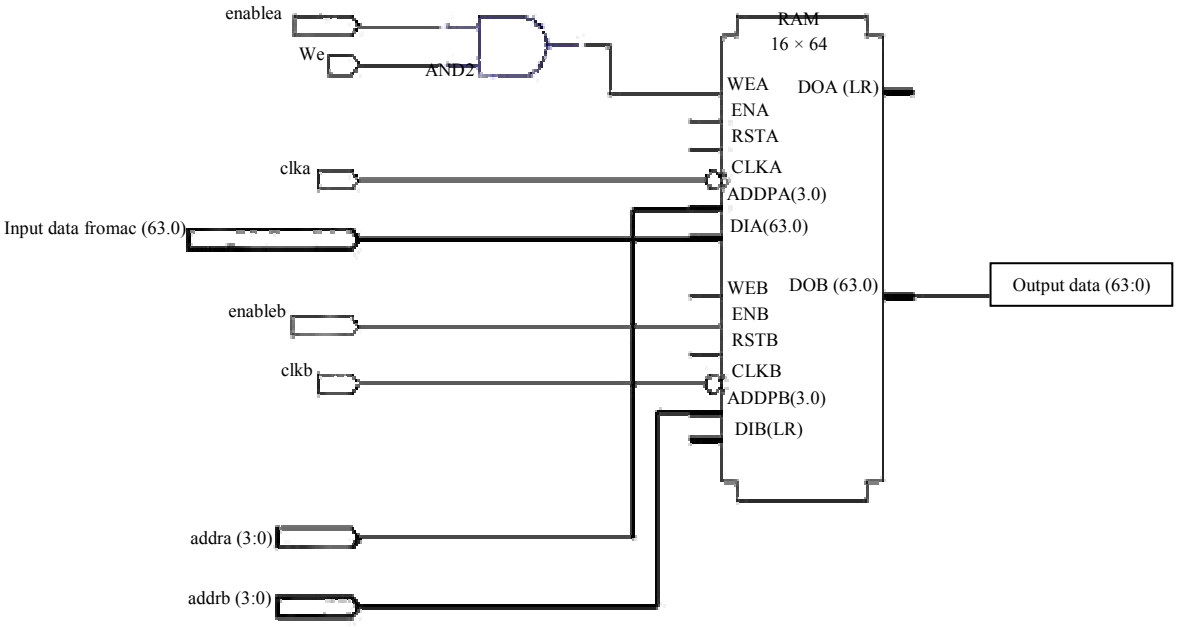

Figure (4): RTL schematic diagram of dual RAM (acum-ram)

Table (3) presents the pin description of the RAM module.

Table (3) Pin description of the dual port RAM module

I/0 Pin	Type	Description
WEA	Active high input	Write enable (port A)
ENA/ENB	Active high input	RAM enable (port A/port B)
ADDRA	4 bits input	Write address
ADDRB	4 bits input	Read address
CLKA	Falling Edge clock	Write clock
CLKB	Falling Edge clock	Read clock
DIA	64 bits input	Data input
DOB	64 bits output	Data output

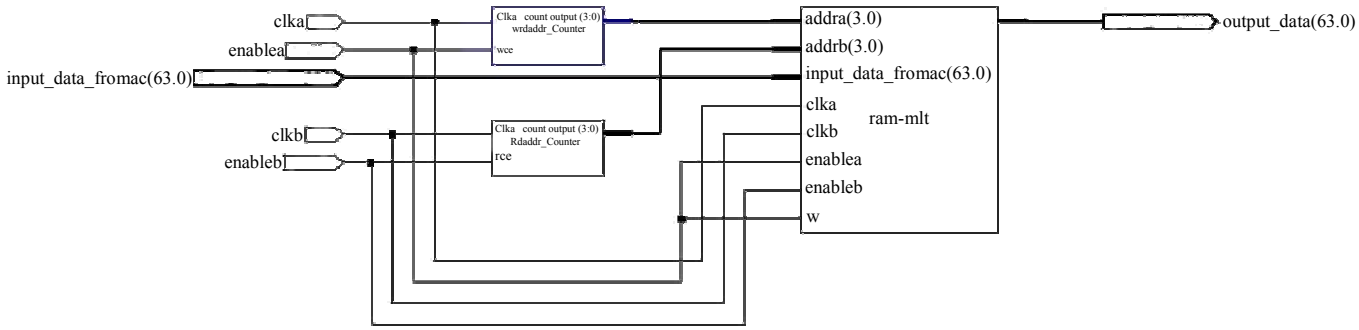

Figure (5): RTL schematic diagram of the write and read counters connected to the memory addra and addrb respectively

Figure (5) demonstrates the connection of write address counter (wadd-counter) and the read

address counter (rdadd-counter) to the (addra) and (addrb) ports of the memory. Table(4) shows the pin description of the address counters.

Table (4) Pin description of address counters

I/0 Pin	Type	Description
Wce	Enable	Active high write address counter enable signal
rce	Enable	Active high read address counter enable signal
CLKC	Clka, Clkb	Write / read address counters clock

2 Read only Memory (Rom) Module:

The designed system employed two Rom modules. The 1st module (Rom1) was used to store the weights and biases of the 1st layer (15 rows × 32colums). The 2nd layer Rom module (Rom2) was used to store the weights and biases of the 2nd layer (7 rows × 17 columns). The design of the Rom module is similar to the design of the RAM module except that the Rom module need only read address counter, as the module is used to read data only.

3 Log-Sigmoid Module

The common activation function is the sigmoid function described by equation [7].

$$f_1 = \frac{1}{1+e^{-x}} \quad \text{.....(3)}$$

A straight forward sigmoid implementation requires a lot of area; therefore, approximation is the only practical solution in digital ANNs. A second order approximation scheme that requires two adders, two multipliers and comparators is characterized by the following equations [11].

$$f_1 = 2^{-1} * (1 - \left|2^{-2} * x\right|^2) \; for \quad -4 \le x \langle o \text{.....(4)}$$

$$f_1 = 1 - 2^{-1} * (1 - \left|2^{-2} * x\right|^2) for \;\; 0 \le x \le 4 \text{.....(5)}$$

$$f_1 = o \quad for \quad x \langle -4 \text{.....(6)}$$

$$f_1 = 1 \quad for \quad x \rangle 4 \text{.....(7)}$$

The log-sigmoid unit takes 8 system clock pulses to perform sigmoid calculations. Figure (6) displays the RTL schematic diagram of the log-sigmoid module.

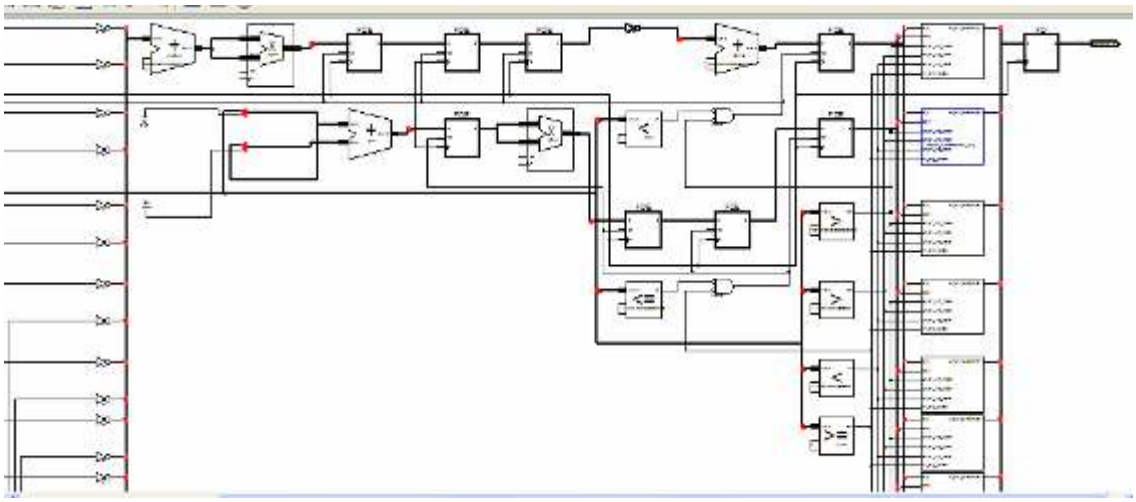

Figure (6): Log-sigmoid module.

System Hierarchy:

The two-layer perceptron neural network is constructed hierarchically by designing each module individually using VHDL language, testing its performance using Xilinx ise simulator and connecting it with other modules.

The following figures represent the RTL schematic diagram that results from synthesizing and hierarchically building the successive modules. Figure (7) displays the RTL schematic diagram of RAM1, Rom1 and the multiplier modules with their connection to perform the multiplication of the input vector with the 1st –layer weight vector.

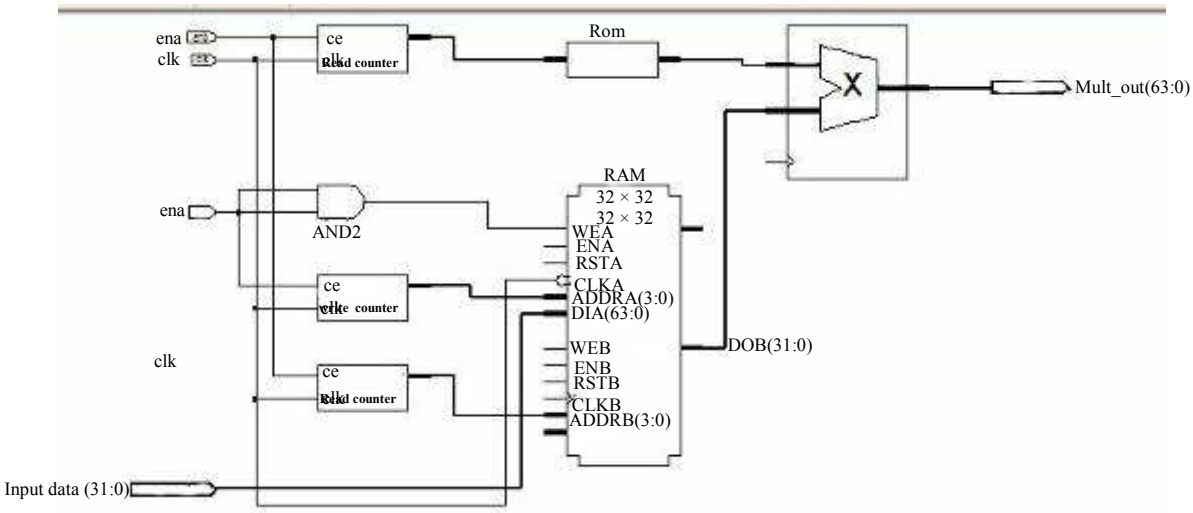

Figure (7): RTL diagram of the RAM1, Rom1 and their vectors multiplication.

Figure (8) demonstrates the accumulator module whose input (acum-in) is the output signal of the multiplier, therefore; its bit width is 64 bits.

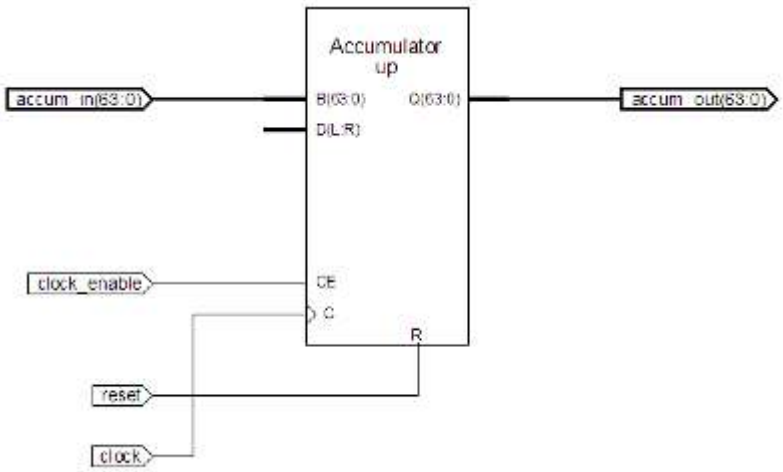

Figure (8): RTL schematic diagram of the accumulator module.

The output of the accumulator is stored in a RAM module (acum-ram) as shown in figure (5).

Figure (9) presents the structure of the 1st layer where the Mc-unit implies the Rom1, RAM1, multiplier and the accumulator of the 1st layer. The output of the accumulator is introduced to the acum-ram unit to be stored there and to be fed to the log-sigmoid unit

comp2. the counter-nor unit is used to generate two pulses, count-30 and count-31. the count-30 pulse is used to drive the clock input (clka) of the address write counter, it appears once each 31 system clock pulses and used to

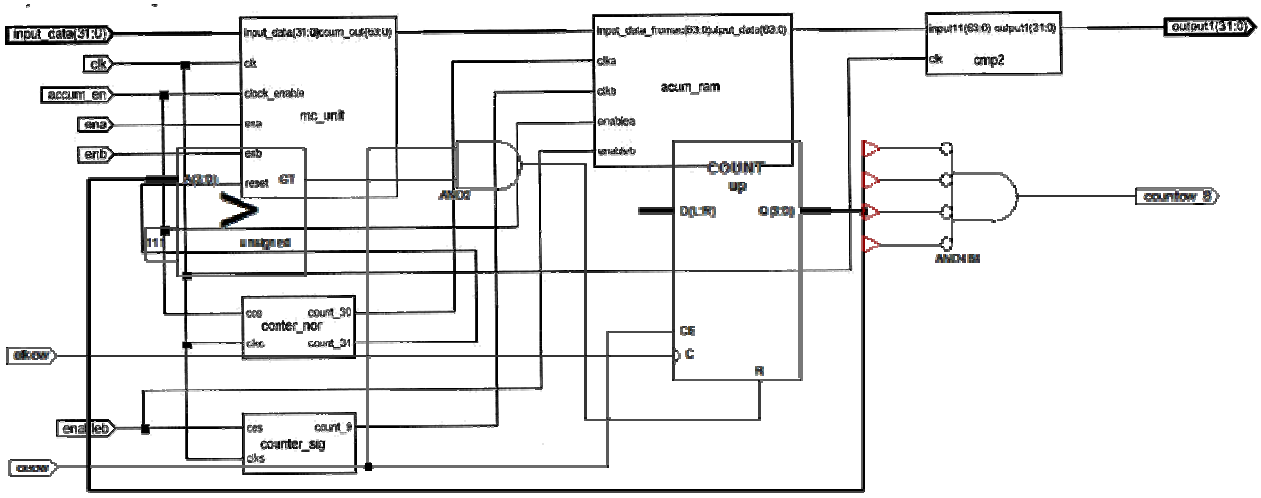

Figure (9): RTL schematic diagram of the 1st layer.

write the output of the accumulator to the memory after completing the multiplication and accumulation process of each weight matrix row.

Count-31 pulse is a pulse that appears once each 32 system clock pulses and used to reset the accumulator to start new row.

Counter-sig unit is used to generate a pulse per eight system clock pulses (count-9), it is used to enable reading the data from the memory (enable b port) once each 8 system clocks.

Both counter-nor and counter-sig units are composed of a counter, comparator and gates (NOR, AND gates) as shown in figure (10).

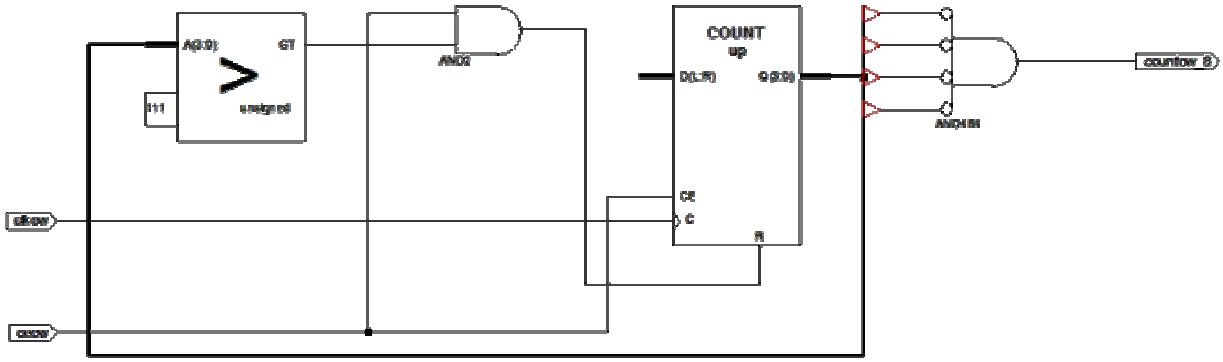

Figure (10): RTL schematic diagram of counter-sig unit.

Figure (11) displays the RTL schematic structure of the two-layer perceptron neural network where lyr1 unit implies all the modules of the 1st layer, ly2 unit implies all the modules of the 2nd layer, output ram unit is a RAM module to store the final discrimination result. Counter-sigm12 module is used to act as comp2 unit and finally the write counter unit is to supply the output ram unit a write clock pulse (clka). The modules in the 2nd layer are

similar to those in the 1st layer except their capacities which are shown in figure (2).

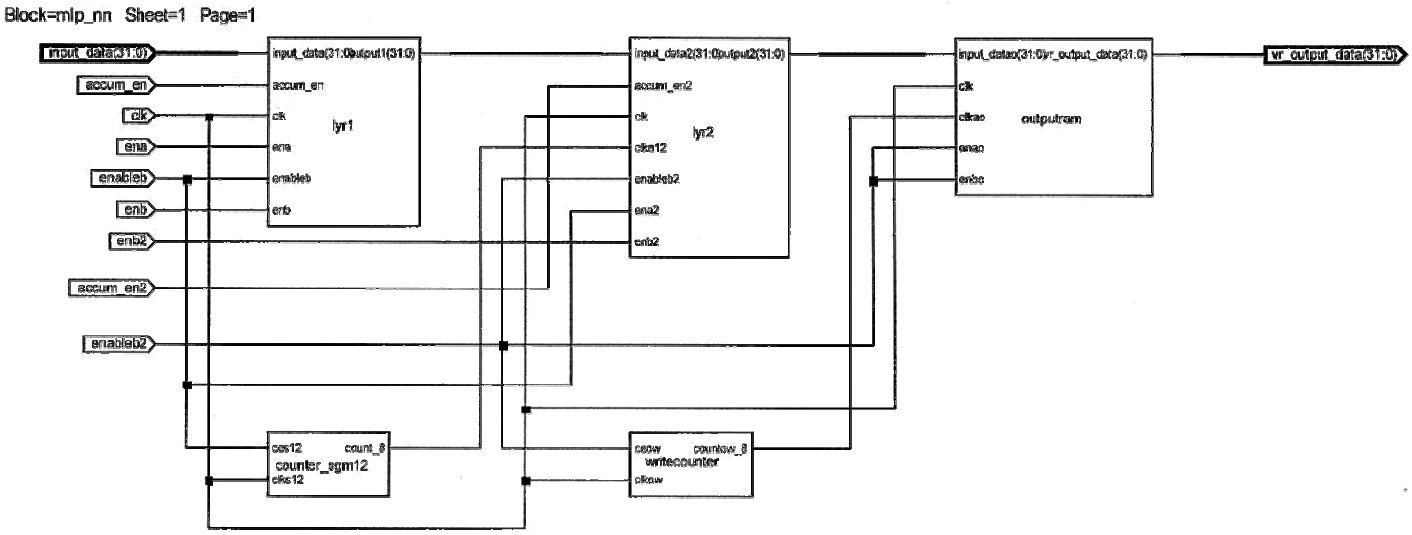

Figure (11): The RTL schematic diagram of the two-layer perceptron neural network.

Practical Results:

In order to test the functionality of the designed network, the features of (7) spoken syllables with the weights of the 1st and 2nd layers (W_{11},W_{12}) are obtained from running a specific programs prepared for this purpose using matlab software[9]. The obtained results are as following. Figure (12): shows the weights W11,W21 stored in Rom1 and Rom2 respectively.

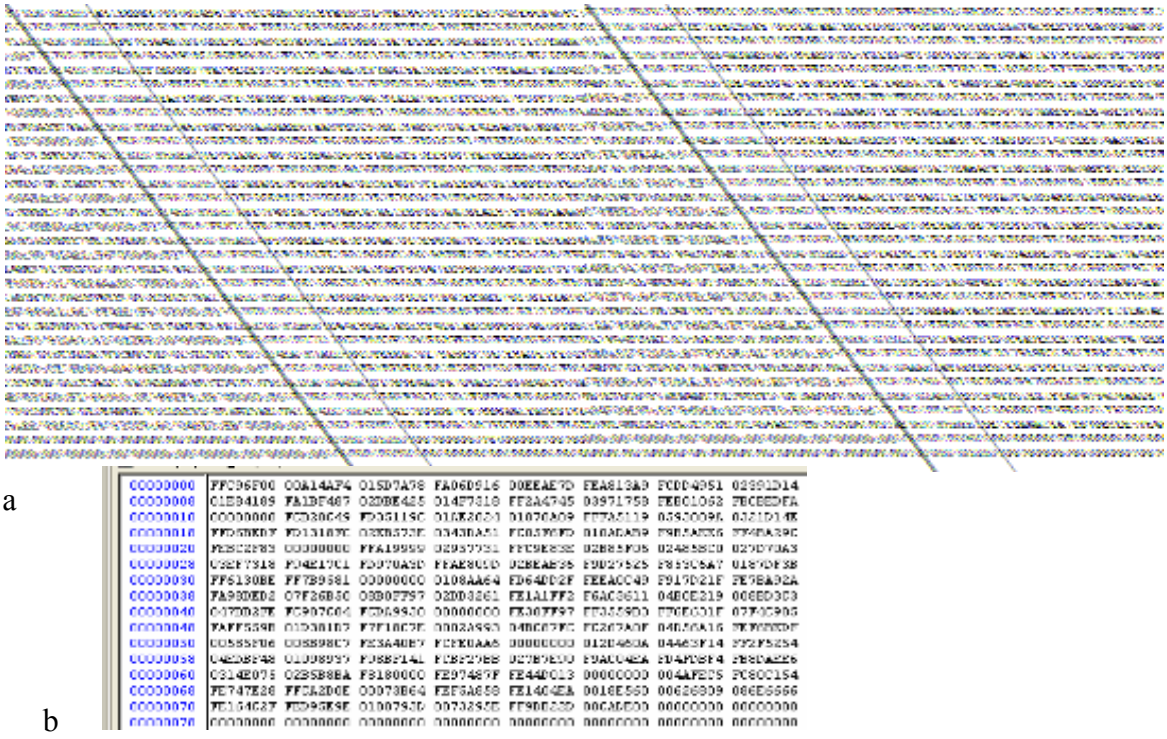

Figure (12): The reads only memories contents.
a. weights of the 1st layer stored in Rom1.
b. Weight of the 2nd layer stored in Rom2.

2. Figure (13): demonstrates the control signals of the system.

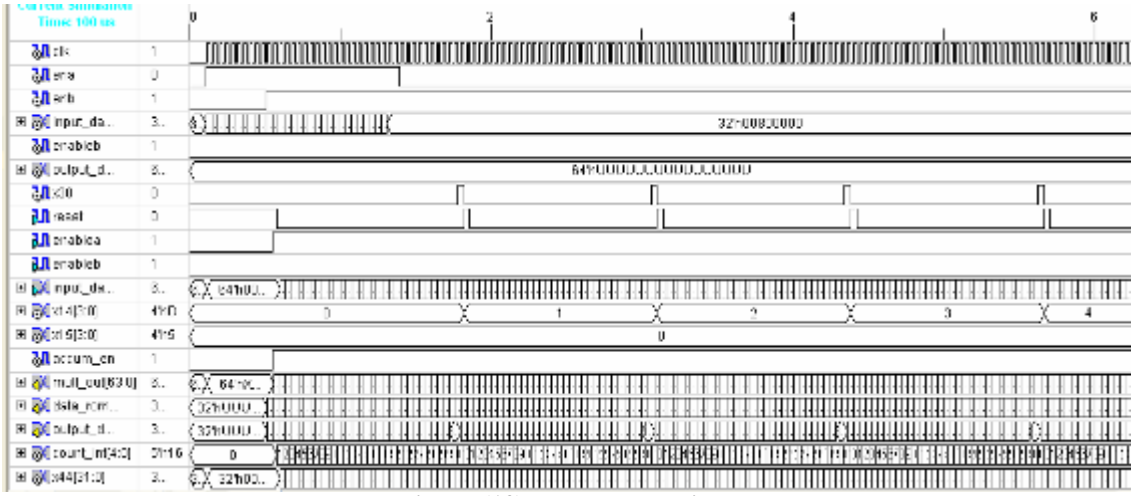

Figure (13): The control signals.

The memories (RAM1, acum-ram) write / read enables (ena, enb, enablea, enableb respectively), count-30 (x30), count-31 (Reset), accumulator enable (acum-en) and system clock (clk) signals besides the input-da signal at the top which represents the input vector (features) are shown.

3. Figure (14): displays the output of Rom1 (data-rom), RAM1 (output-d), the result of their multiplications (mult-out) and the result of accumulation that is stored in acum-ram (input-da) at the middle of the figure.

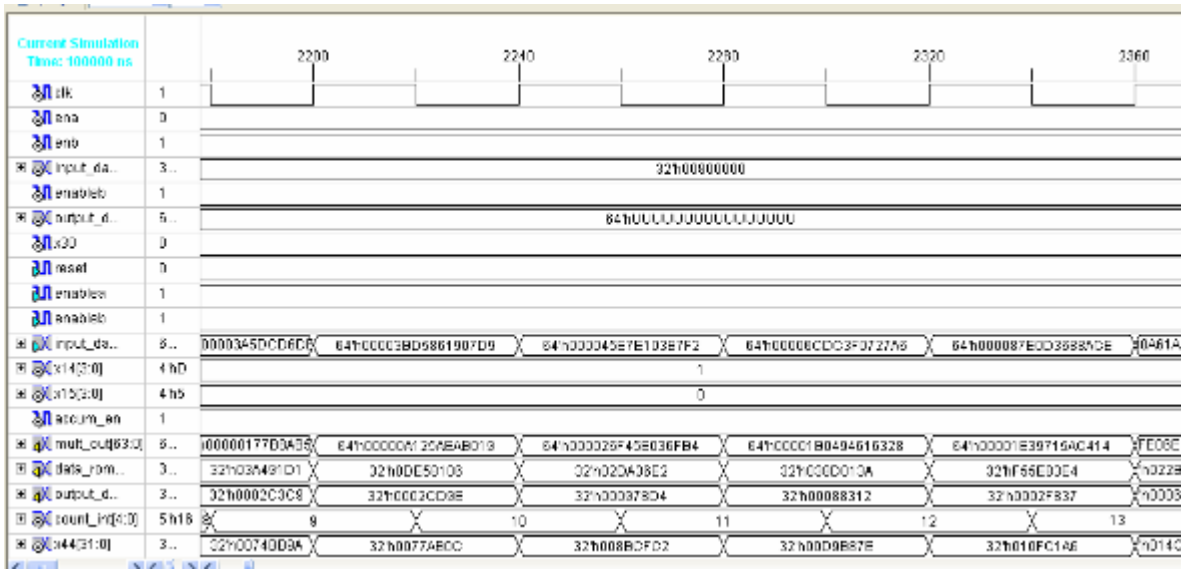

Figure (14): Multiplications and accumulations of Rom1 and RAM1 contents.

4. Figure (15): presents the timing of the control signals, X30 and resets, where X30 is used to hunt the last term of the 1st row accumulation (FFFFE3 F52 AB21EF1) and store it in the acum-ram module. The reset signal is next to the X30 signal and resets the accumulator. The

count-int signal represents the number of items in each row of the weight matrix that are stored in Rom1 (0-31), the last item in each row is zero in order not to lose data during the accumulator reset process.

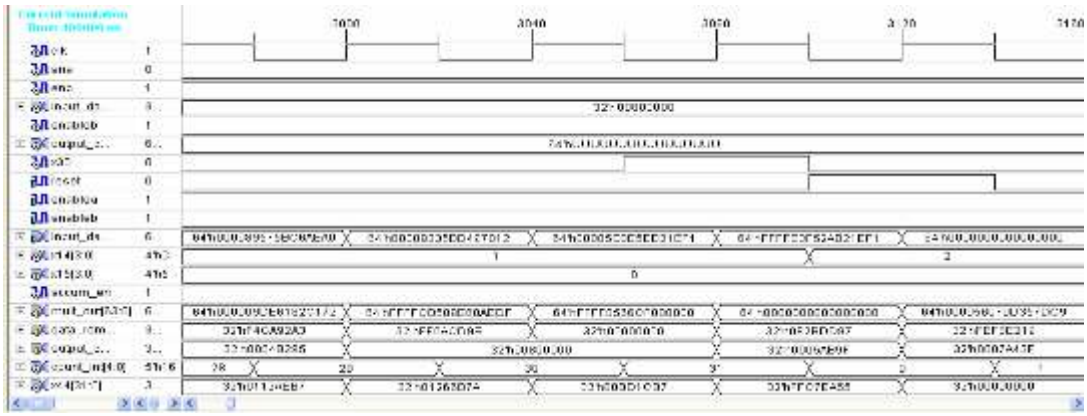

Figure (15): Timing diagram of reset and hunting signal ×30.

5. Figure (16): shows the contents of acum-ram module which represents the hunted outputs of the accumulator at the end of each accumulation process. This is shown by the signal (output-d) in the upper part of the figure.

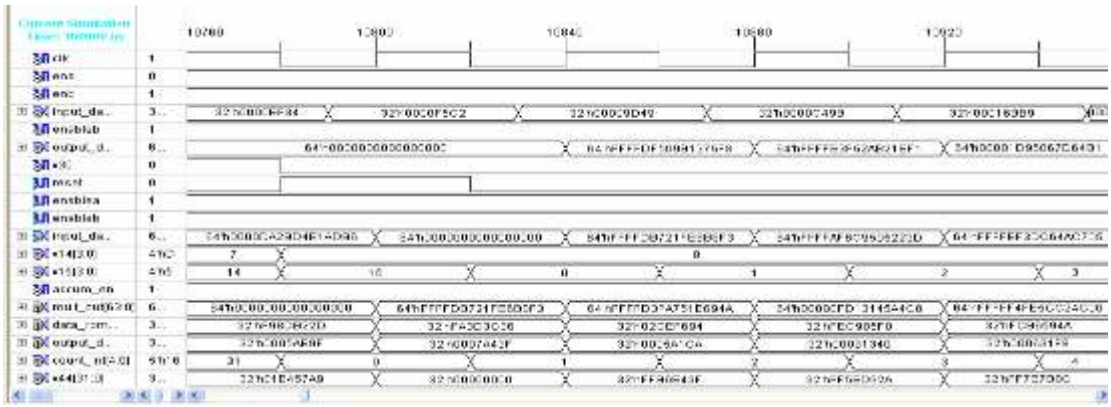

Figure (16): The contents of the memory module named acum-ram.

6. Figures (17)&(18): demonstrate the output of the log-sigmoid units. The signal XS9 is the output of the control unit counter-sig, it is used as clock signal to the acum-ram address read counter to read the data (signal X55) from the memory and feed it to the log-sigmoid unit. After 8 clocks as shown in the signal count-int[3:0]; the output of the log sigmoid unit appears as shown in the signal output1.

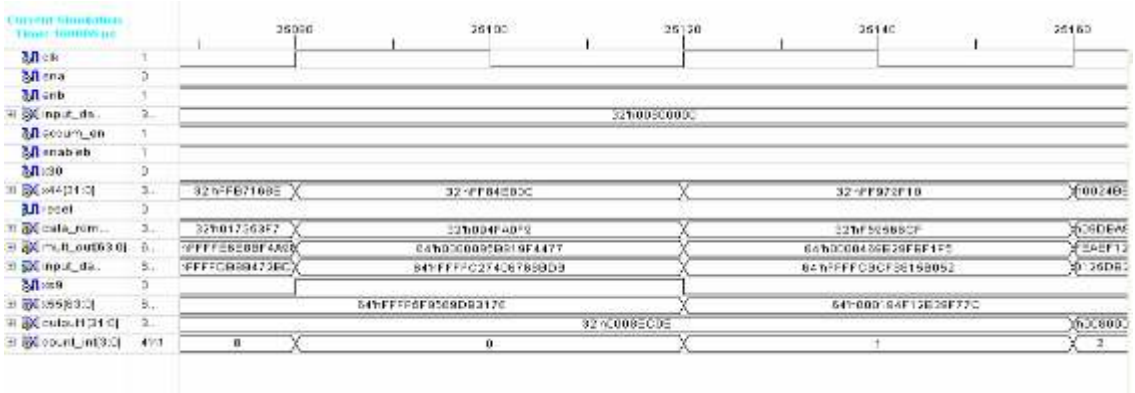

Figure (17): The output of the 1st layer log-sigmoid unit.

7. Figure (19): presents the result of the discrimination process performed by the designed network, addrbo signal is the address of the output memory module (output ram). The signal Vr-output represents the output value stored in each memory location, it is (0100000) which represents the recognition of the 2nd syllable out of the 7 syllables.

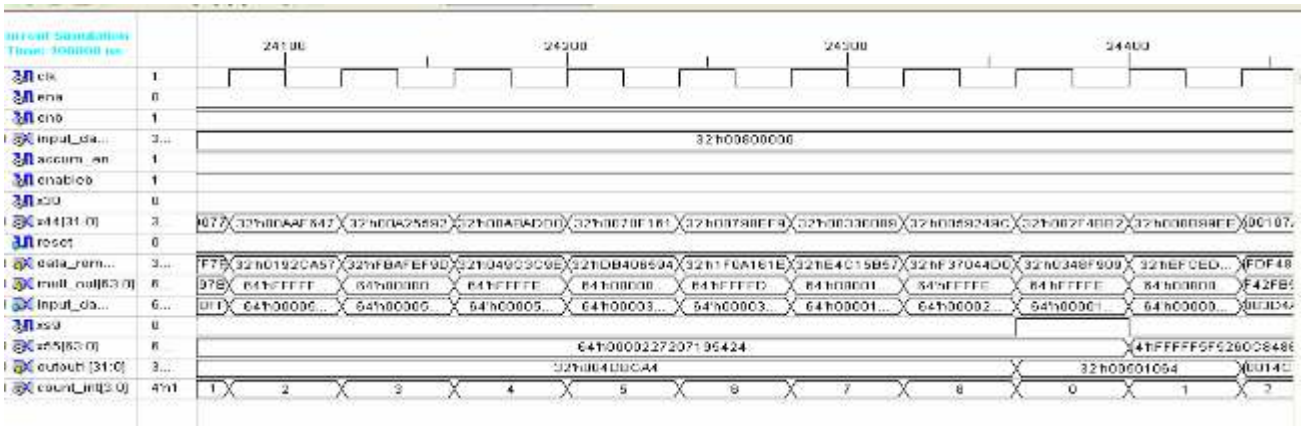

Figure (18): The output of the 2nd layer log-sigmoid unit.

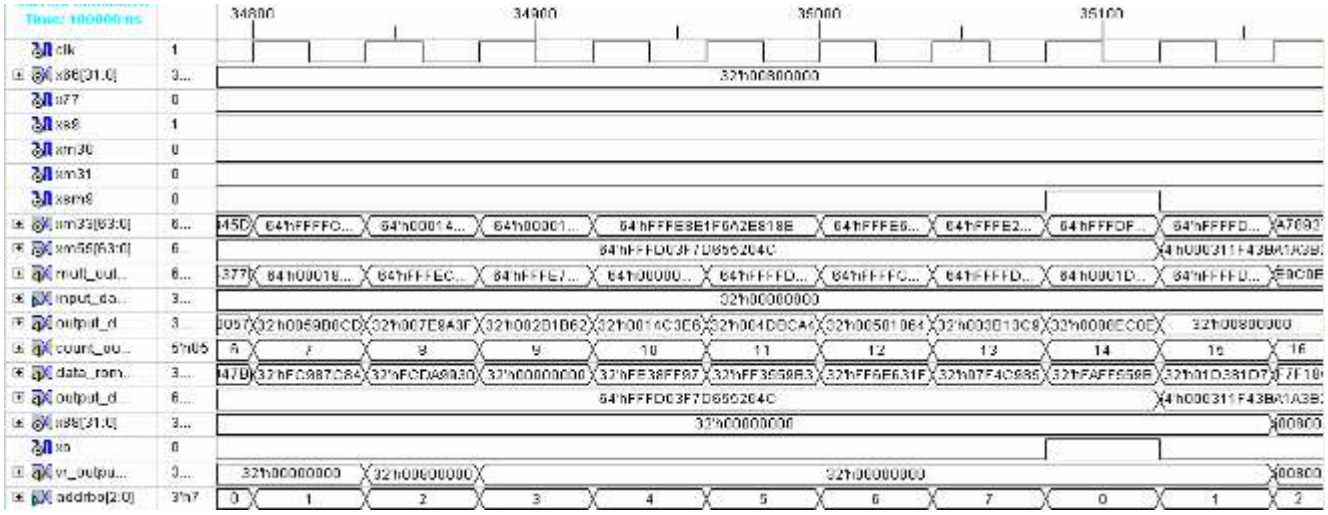

Figure (19): The recognition result.

8.The above mentioned results showed that the designed neural network when trained for the features of seven syllables , it can recognize any of them, if the input to the network is the feature of that syllable. The test input that was fed to the network was the features of the 2nd syllable, therefore the discrimination result was(0100000) .Table (5) shows the real time discrimination results for each spoken syllable.

Table (5) The utterance of each spoken syllable, targets and real time system discrimination output

Spoken Syllable	Target	Real time discrimination output
1. Move	0000001	0000001
2. Forward	0000010	0000010
3. Backward	0000100	0000100

4. Pause	0001000	0001000
5. Right	0010000	0010000
6. Left	0100000	0100000
7. Stop	1000000	1000000

Conclusions:

The field programmable gate arrays provided a suitable environment for implementing MLPNN, since the number of neurons, layers and interconnections can be varied dynamically. A hierarchical design of a feed forward two-layer peceptron neural network is presented to be configured on a field programmable arrays (FPGAs) slice type spartran 3e. On the basis of experimental results the following conclusions are inferred:

1.The designed network can be applied in real time recognition systems.

2.Reasonable consumption of the available resources can be taken into consideration throughout the design steps by selecting the best algorithm for approximating the activation sigmoid function in addition to using a dual port random access memories.

3. Fixed point as well as floating point numbers are the best solution for preventing data width inflation throughout successive mathematical operations.

4.Using hierarchical design structure in implementing FPGAs based systems facilitates the debugging and verification of the system performance.

References:

1. Amos R., Jagath C., *FPGA implementation of neural networks*, springer. The Netherlands, 2006.
2. Pavlitov K., Mancler O., *FPGA Implementation of Artificial Neurons*, Electronics, Vol.6, PP.22-24,2004.
3.Hidetoshi O., Kiyoshi T., Keilcichj T., *Hardware Architecture for Kohonen Network*, IEEE Int Symp. on Circuits and Systems, vol. II, pages 1073-1077, 1990.
4. Wai-Chi F., Bing S., Oscal C., Joongho C., *A VLSI Neural Processor for Image Data Compression Using Self- Organization Networks*, IEEE Transaction on Neural Networks, Vol. 3, No. 3, May 1992.
5. Jarkko V., Teuvo K., *Fast DSP Implementation of High-Dimensional Vector Classifier*, In Proc. ICNN'95 . IEEE Int. Conf. on Neural Networks, volume IV , Pages 2019-2022, 1995.
6. Pricipe J., Euliano N, Lefrebvre W., *Neural and adaptive systems*, John Wiley, New York, 2000.
7. Haddad A., *Multiresolution signal Decomposition*, Academic press, San Diego, 1992.
8. Jiang H., Jooer M., Gooy. ,*Feature Extraction Using Wavelet Packet Strategy*, Proceeding of 42nd IEEE conference on Decision control, Hawaii USA. PP. 4517-4520, 2003.
9. Salem M., *Speech identification based on neural network* Msc dissertation, comp Eng. Dept. Technical college of Mosul, F.T.E. Iraq. 2008.
10. Volnei, A., *Circuit design with VHDL*, MIT press, London, 2004.
11. Tommiska M. ,*Effective Digital Implementation of the sigmoid function for Reprogrammable logic*,. IEE computers and digital techniques, Vol. 6. PP. 403-411, 2003.