World Academics Journal of Engineering Sciences
Vol.7, Issue.4, pp.63-69, December (2020)

Research Paper
E-ISSN: 2348-635X

Neural Network Architecture for Signal Processing Analog VLSI Implementation

Priyanjali Jain[1*], Priyanshu Jain[2]

[1]Department of Electronics and Communication, Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar (Ahmedabad) Gujrat, India
[2]Department of AVIONICS, Indian Institute of Space Science and Technology, IIST, Department of Space, Govt. of India, Valiamala P.O., Thiruvananthapuram – 695547, Kerala, India

[*]*Corresponding Author: 2710priyanjali@gmail.com*

Available online at: www.isroset.org

Received: 10/Oct/2020, Accepted: 05/Nov/2020, Online: 31/Dec/2020

Abstract- With the advent of new technologies and advancement in medical science we are trying to process the information artificially as our biological system performs inside our body. Artificial intelligence through a biological word is realized based on mathematical equations and artificial neurons. Our main focus is on the implementation of Neural Network Architecture (NNA) with on a chip learning in analog VLSI for generic signal processing applications. In the proposed paper analog components like Gilbert Cell Multiplier (GCM), Neuron activation Function (NAF) are used to implement artificial NNA. The analog components used are comprises of multipliers and adders' along with the tan-sigmoid function circuit using MOS transistor in subthreshold region. This neural architecture is trained using Back propagation (BP) algorithm in analog domain with new techniques of weight storage. Layout design and verification of the proposed design is carried out using Tanner EDA 14.1 tool and synopsys Tspice. The technology used in designing the layouts is MOSIS/HP 0.5u SCN3M, Tight Metal.

Keyword- Back Propagation, Algorith, Neural Network Architecture

I. INTRODUCTION

Artificial Intelligence

Intelligence is the computational part of the ability to achieve goals in the world. Actually intelligence is a biological word and is acquired from past experiences [1]. The science which defines intelligence mathematically is known as Artificial Intelligence (AI). Artificial Intelligence is implemented by using artificial neurons and these artificial neurons comprised of several analog components. The proposed paper is a step in the implementation of neural network architecture using back propagation algorithm for data compression[2]. The neuron selected is comprises of multiplier and adder along with the tan-sigmoid function. The training algorithm used is performed in analog domain thus the whole neural architecture is a analog structure[2,3]. Figure 1 can be expressed mathematically as

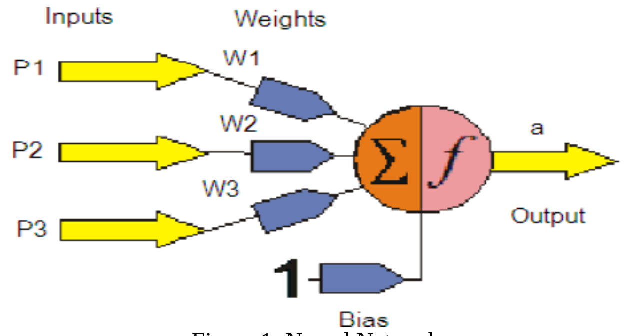

Figure 1: Neural Network

a = f (P1W1+P2W2+P3W3+Bias)

where „a" is the output of the neuron & „p" is input and „w" is neuron weight . The bias is optional and user defined. A neuron in a network is itself a simple processing unit which has an associated weight for each input to strengthening it and produces an output. The working of neuron is to add together all the inputs and calculating an output to be passed on. The neural architecture is trained using back propagation algorithm and also it is a feed forward network. The designed neuron is suitable for both analog and digital applications [4]. The proposed neural architecture is capable of performing operations like sine wave learning, amplification and frequency multiplication and can also be used for analog signal processing activities.

Multiple Layers of Neurons

When a set of single layer neurons are connected with each other it forms a multiple layer neurons, as shown in the figure 2.

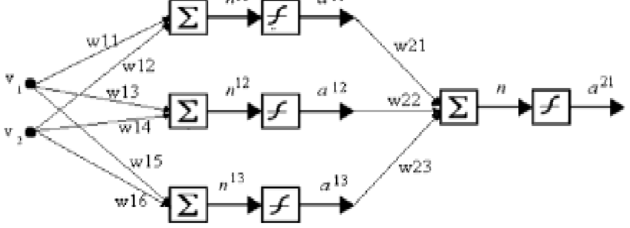

Figure 2: Layered structure of Neural Network

As it is clear from the above figure that weights w11 to w16 are used to connect the inputs v1 and v2 to the neuron in the hidden layer [5]. Then weights w21 to w23 transferred the output of hidden layer to the output layer. The final output is a21.

II. ARCHITECTURE

Analog Components for Neural Architecture

The inputs to the neuron v1 and v2 as shown in figure 2 are multiplied by the weight matrix, the resultant output is summed up and is passed through an NAF. The output of the activation function is then passes to the next layer for further processing. Blocks to be used are Multiplier block, Adders, NAF block with derivative[5,6,7].

Multiplier Block

The Gilbert cell is used as the multiplier block. The schematic of the Gilbert cell is as shown in the figure 3.

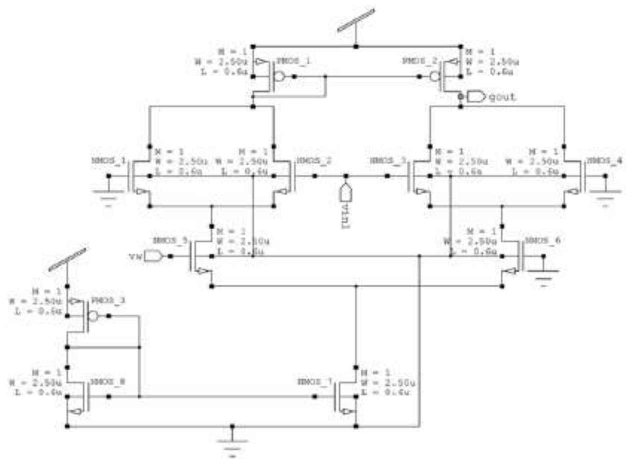

Figure 3: Gilbert cell schematic with current mirror circuit

As in this paper we are mainly focusing on layout part so the layout of the Gilbert cell multiplier is shown in figure 4. Layout design and verification of the proposed design is carried out using Tanner EDA 14.1 tool and Synopsys Tspice [8,9]. The technology used in designing the layouts is MOSIS/HP 0.5u SCN3M, Tight Metal.

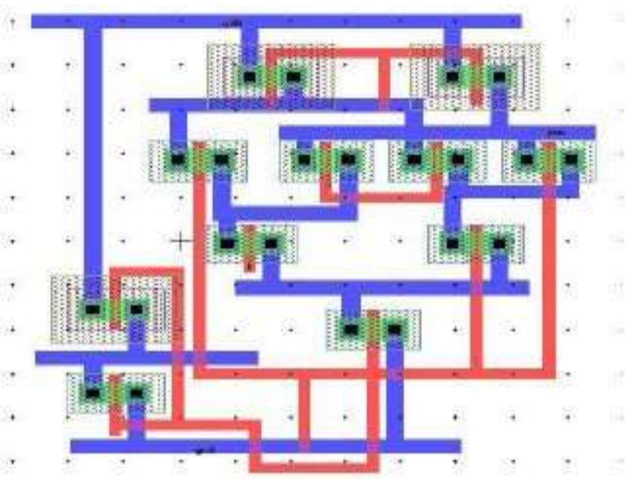

Figure 4: Layout of Gilbert cell multiplier

Adders

The output of the Gilbert cell is in the form of current (trans conductance).The node of the Gilbert cell connecting the respective outputs act as adder itself [5,7].

Neuron Activation Function (NAF)

The designed activation function is tan sigmoid. The proposed design is actually a differential amplifier modified for differential output. Same circuit is capable of producing output the activation function and the differentiation of the activation function. Here two designs are considered for NAF [11].

1. Differential amplifier as NAF
2. Modified differential amplifier as NAF with differentiation output.

Differential Amplifier Design As A Neuron Activation Function (Tan)

This block is named as tan in the final schematics for Neural Architecture. Differential amplifier when design to work in the sub-threshold region acts as a neuron activation function [3,8].

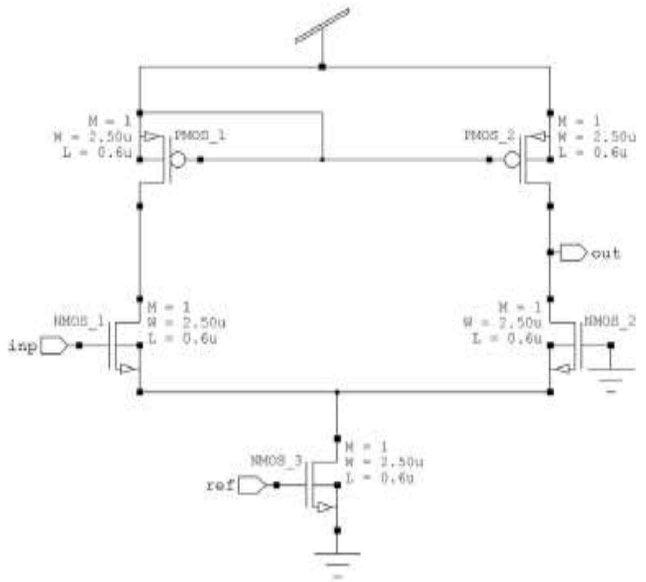

Figure 5: Schematic of NAF

Modified Differential Amplifier Design for Differentiation Output (fun)

Schematic of the design shown in the figure 5 is used for the tan sigmoid function generator with modification for the differential output. The NAF function can be derived from the same differential pair configuration [12]. The structure has to be modified for the differential output.

Layout issues

As we already discussed that we are using tan sigmoid function to realize NAF the reason behind using tan-sigmoid function is that after carried out a mathematical analysis we found that tan sigmoid function is best suitable for compression and achieving a tangential output [13]. The layout of NAF is shown in figure 6.

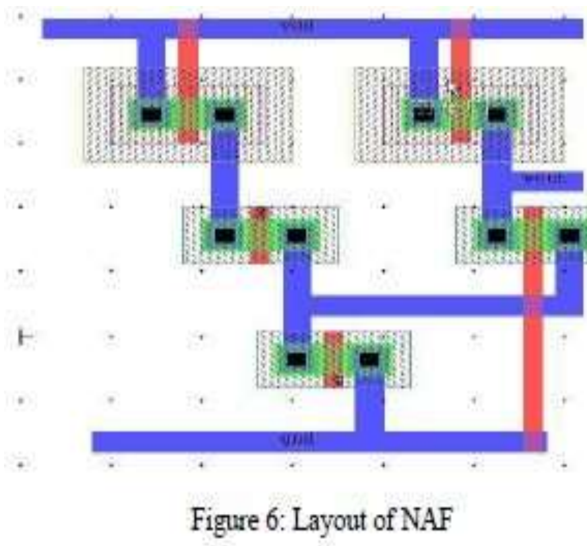

Figure 6: Layout of NAF

Realization of Neural Architecture using Analog Components

The components designed in the previous section are used to implement the neural architecture. The tan block is the differential amplifier block designed in section 2.1.3.1. This block is used as the neuron activation function as well as for the multiplication purpose [11,14]. The mult is the Gilbert cell multiplier designed in section 2.1.1 The fun is the Neuron activation function circuit with differential output designed in section 2.1.3.2

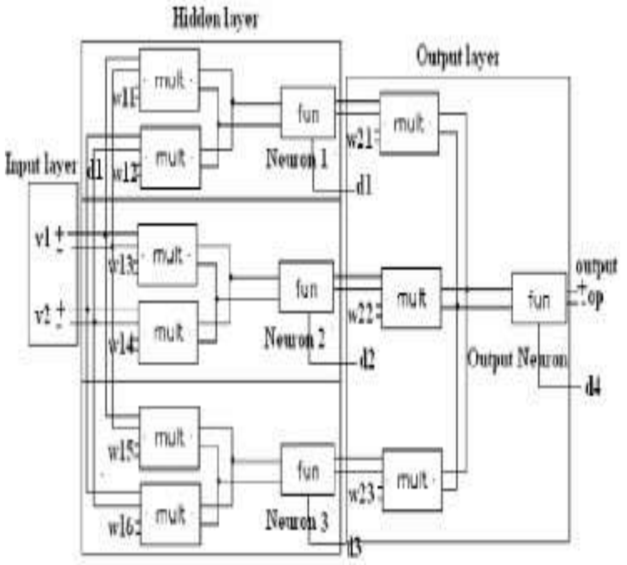

Figure 7: Implementation of the Neural Architecture using Analog Blocks

Figure 7 shows exactly how the neural architecture of figure 2 is implemented using analog components. The input layer is the input to the 2:3:1 neuron. The hidden layer is connected to the input layer by weights in the first layer named as w1i. The output layer is connected to input layer through weights w2j. The op is the output of 2:3:1 neuron [15]. The schematic of circuit for image compression is shown in figure 8.

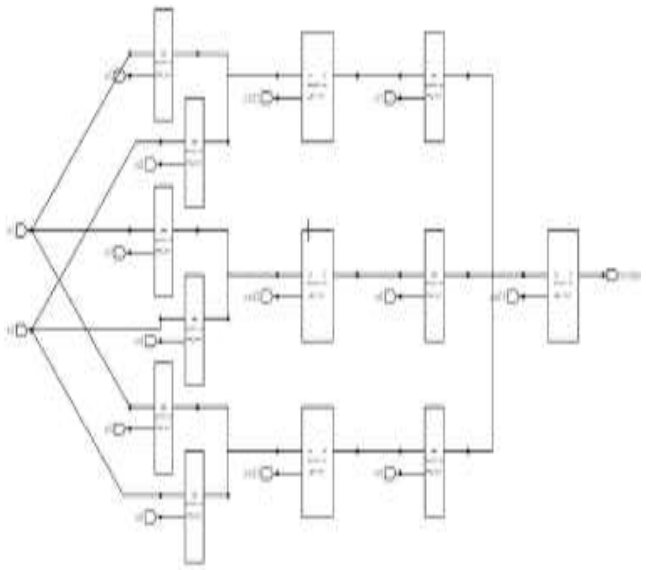

Figure 8: Schematic of Compression Block

The Layout of circuit for image compression is shown in figure 9.

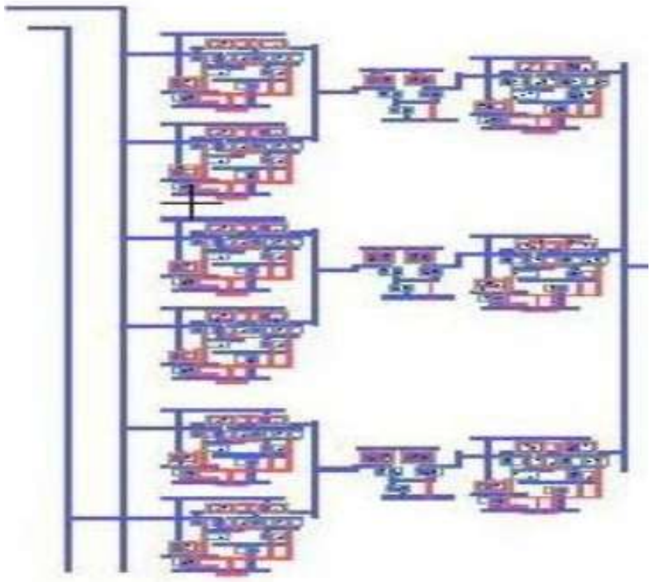

Figure 9: Layout of compression block

Neuron Application -Image Compression and Decompression

The above proposed NA is used for image compression. Image consisting of pixel intensities are fed to the network shown in Figure 8 for compression and then this compressed image act as input for the decompression block [16]. The layout circuit of decompression block is as shown in figure 11.

The 2:3:1 neuron proposed has an inherent capability of compressing the inputs, as there are two inputs and one output. The compression achieved is 97.30%. Since the inputs are fed in the analog form to the network there is no need for analog to digital converters [17].

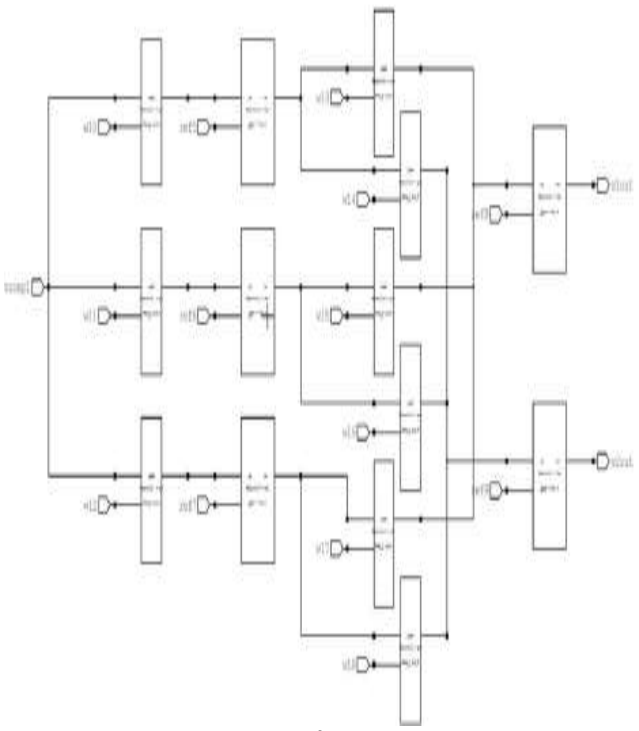
Figure 10: Schematic of Decompression Block\

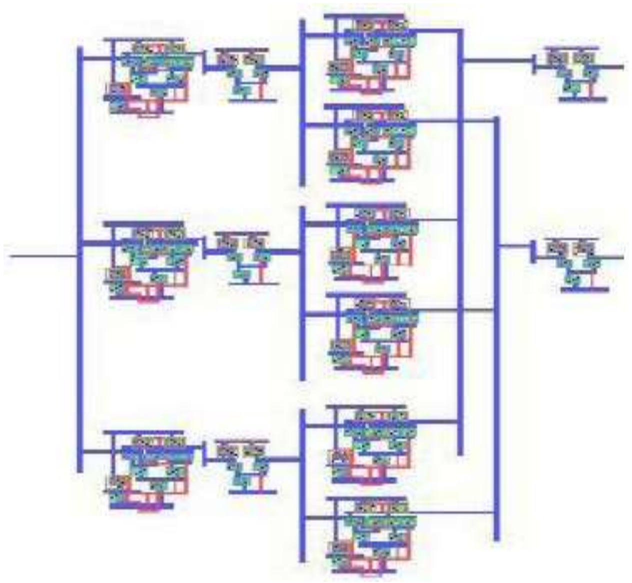
Figure 11: Layout of Decompression Block

This is one of the major advantages of this work. A 1:3:2 neural networks is designed for the decompression purpose. The neural network has 3 neurons in the hidden layer and two in the output layer [18,19].

Figure 12 shows the compression and decompression scheme. The training algorithm used in this network is Back Propagation algorithm. The error propagates from decompression block to the compression bock.

Once the network is trained for different inputs the two architectures are separated and can be used as compression block and decompression block independently [20].

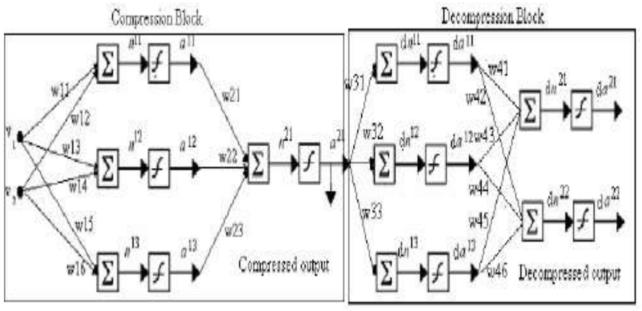

Figure 12: Image Compression and Decompression using proposed neural architecture

The layout implementation of block diagram of compression and decompression block is shown in figure 14 and figure 13 shows the combined schematic of compression and decompression block:

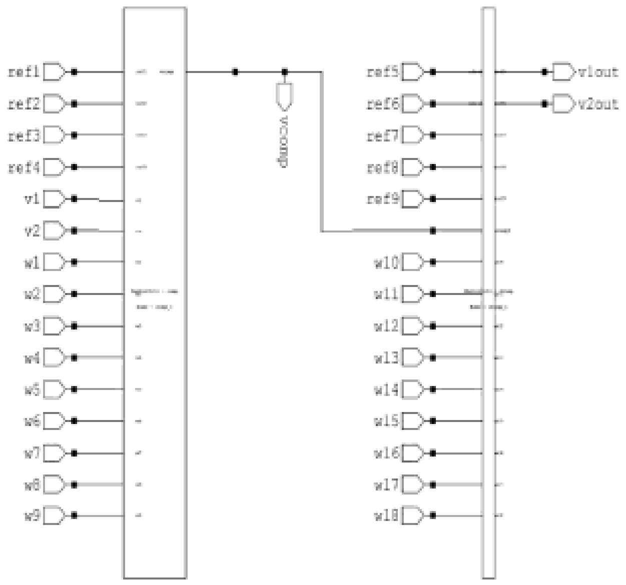

Figure 13: Combined Schematic of Compression and Decompression Block

The layout implementation of block diagram of compression and decompression block is shown in figure 14 and figure 13 shows the combined schematic of compression and decompression block:

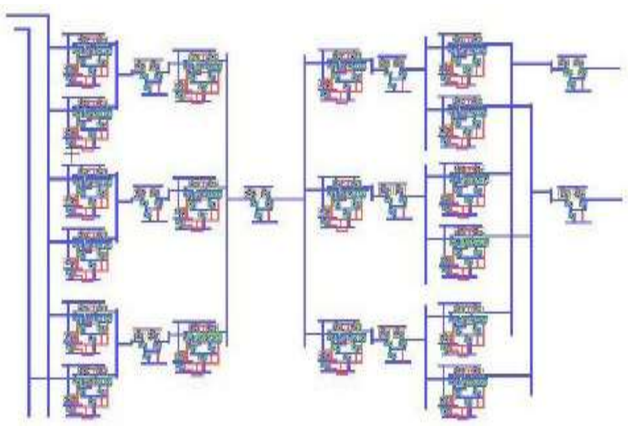
Figure 14: Combined Layout of Compression and Decompression block

III. BACK PROPAGATION ALGORITHM

Back propagation is the most common method of training of artificial neural network so as to minimize the objective function [5,8]. It is the generalization of delta rule and mainly used for feed forward networks. The back propagation algorithm is understand by dividing it into two phases. First phase is propagation and second is weight update [7].

1. Propagation
a) Forward propagation of training pattern's input through the neural network in order to generate the propagation's output activations [9].
b) Backward propagation of the of the propagation's output activations through the neural network using the training pattern's target in order to generate the deltas of all output and hidden in neurons [5].

2. Weight Update
a) For each weight synapse it multiply its output delta and input activation to get the gradient of the weight [7].
b) Bring the weight in opposite direction of the gradient by subtracting a ratio of it from the weight [3].

This ratio influences the speed and quality of learning and it is called learning rate. The sign of the gradient of a weight indicates where the error is increasing, that is why the weight must be updated in the opposite direction. These phases goes on repeating until the performance is of the network is satisfactory [12, 15, 19].

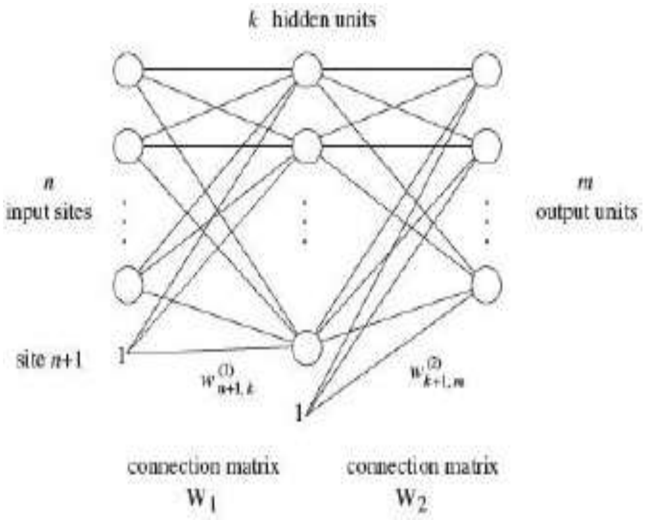

Figure 15: Notation for three-layered network

There are (n + 1) × k weights between input sites and hidden units and (k +1)×m between hidden and output units. Let W1 denote the (n+1)×k matrix with component $w^{(1)}$ at the i-th row and the j-th column. Similarly let W2 denote the (k + 1) × m matrix with components $w_{ij}^{(2)}$ ij . We use an overlined notation to emphasize that the last row of both matrices corresponds to the biases of the computing units [17]. The matrix of weights without this last row will be needed in the backpropagation step. The n-dimensional input vector ® = (o1, . . . , on) is extended, transforming it to ˆo = (o1, . . . , on, 1). The excitation netj of the j-th hidden unit is given by:

$$net_j = \sum_{i=1}^{n+1} w_{ij}^{(1)} \hat{o}_i .$$

The activation function is a sigmoid and the output O $_j^{(1)}$ of this unit is thus

$$o_j^{(1)} = s\left(\sum_{i=1}^{n+1} w_{ij}^{(1)} \hat{o}_i\right)$$

After choosing the weights of the network randomly, the backpropagation algorithm is used to compute the necessary corrections [20]. The algorithm can be decomposed in the following four steps:

(i) Feed-forward computation
(ii) Back propagation to the output layer
(iii) Back propagation to the hidden layer
(iv) Weight updates

The algorithm is stopped when the value of the error function has become sufficiently small.

IV. RESULTS AND DISCUSSIONS

Simulation Result for Gilbert cell Multiplier

The designed Gilbert cell is simulated using TSPICE. The simulation result shown in the figure 16 is for the multiplication of two voltages v1 and v2. v1 voltage is 4 Vpp and 100 MHz frequency. v2 is 2 Vpp 10 MHz frequency [11,15].

The output wave is the multiplication of v1 and v2 voltage done by the circuit. The output amplitude is 5.6 Vpp [20]. The output can be seen matching with the theoretical output.

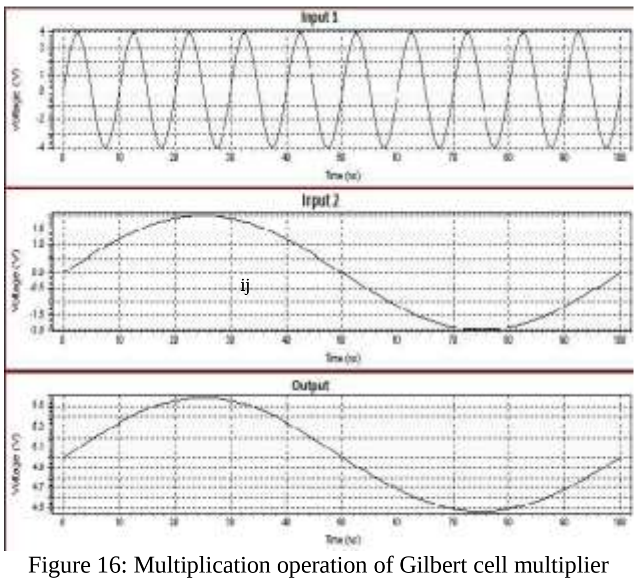

Figure 16: Multiplication operation of Gilbert cell multiplier

Simulation Result for Neuron activation function

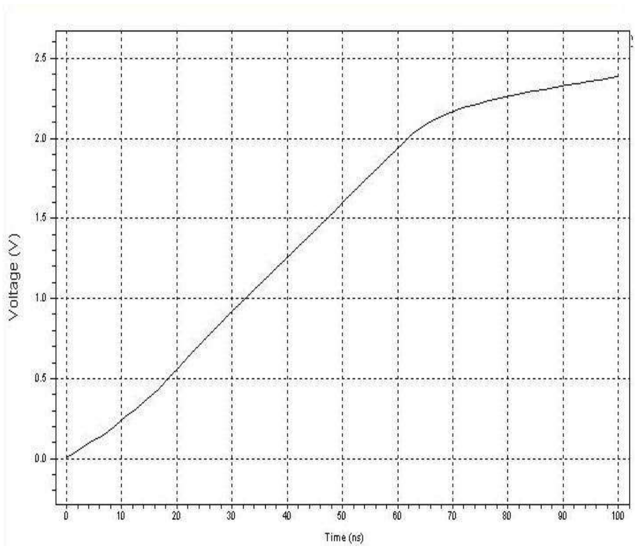

Figure 17: Circuit output for Neuron Activation function block (tan)

Image Compression using Neural Architecture
Image compression is separately performed using neural architecture [7]. The simulation result for image compression are shown in the figure 18 The input v1 was a sine wave with 2 Vpp voltage at 10 MHz frequency and input v2 was a sine wave with 4 Vpp at a frequency 100 MHz. The output of compression is a DC signal of 70 mV [13, 17].

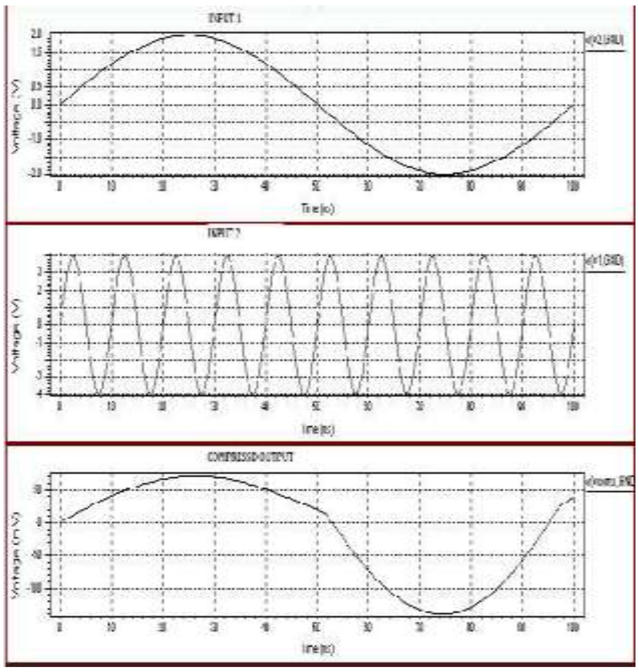

Figure 18: Simulation of Image compression

Image Decompression using Neural Architecture
The separate decompression of a 3 Vpp sine wave at 100 MHz gives the decompressed output as a signal of 7 Vpp [18]. The simulation result of image decompression is shown in figure 19.

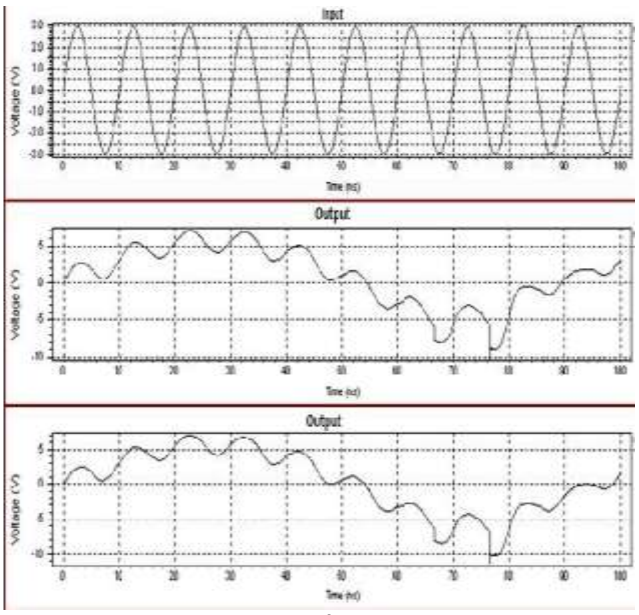

Figure 19: Simulation of Image decompression

Image Compression and Decompression using Neural Architecture
The Neural Architecture is extended for the application of image compression and decompression [13, 16]. The simulation result for image compression and decompression are shown in the figure 20. The input 1 was a sine wave with 16 Vpp voltage 10 MHz frequency and input 2 was a sine wave with 9 Vpp voltage and 100 MHz frequency. The compressed output was a DC signal of 4 Vpp.

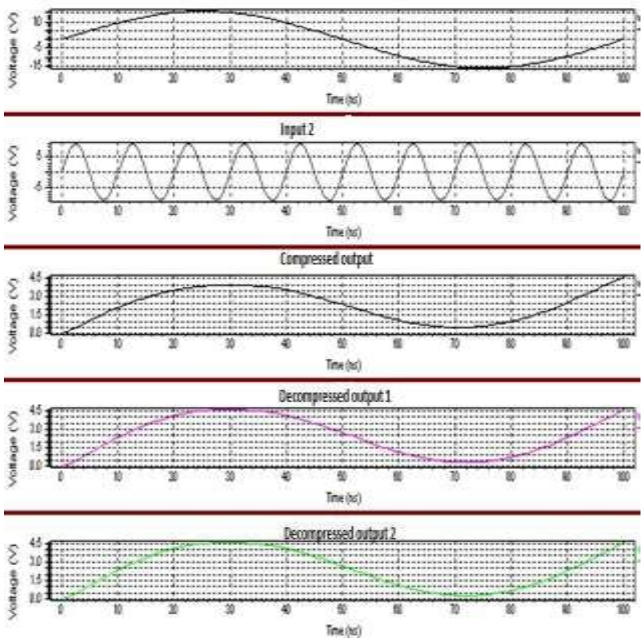

Figure 20: Image compression and Decompression Simulation

The decompressed output for input 1 was a 4.5 Vpp sine wave and for input 2 was a 4.5 Vpp sine wave. The compression we achieved is 97.22 % and the average decompression achieved is 60.94% [15, 18, 20].

V. CONCLUSION

Neural network with their remarkable ability to derive meaning from complicated or imprecise data can be used to extract patterns and to detect trends that are too complex to be noticed by either humans or other computer techniques [6]. Due to its adaptive learning, self-organization, real time operations and fault tolerance via redundant information coding properties it can be used in Modelling and Diagnosing the Cardiovascular System and in Electronic noses which has several potential applications in telemedicine. Another application developed was "Instant Physician" which represents the "best" diagnosis and treatment. This work can be further extended to implement neuro fuzzy system with high speed low power, CMOS circuit (Layout extraction) in current mode as well as for nano scale circuit simulation using Double Gate MOSFET (DG MOSFET) modeling [6, 13].

REFERENCES

[1] Cyril Prasanna Raj P & S.L. Pinnae "DESIGN AND ANALOG VLSI IMPLEMENTATION OF NEURAL NETWORK ARCHITECTURE FOR SIGNAL PROCESSING" European Journal of Scientific Research ISSN 1450-216X Vol.27No.2 (2009), pp.199-216

[2] George Papadourakis "INTRODUCTION TO NEURAL NETWORKS" (2009)

[3] Anita Wasilewska "NEURAL NETWORKS" State University of New York at Stony Brook.

[4] Ranjeet Ranade & Sanjay Bhandari & A.N. Chandorkar "VLSI Implementation of Artificial Neural Digital Multiplier and Adder" pp.318-319

[5] Rafid Ahmed Khalil & Sa'ad Ahmed Al-Kazzaz "Digital Hardware Implementation of Artificial Neurons Models Using FPGA"pp.12-24

[6] Bose N.K., Liang P., Neural Network Fundamentals With graphs, algorithms and Application, Tata McGraw hill, New Delhi, 2002,ISBN0-07-463529-8.

[7] Razavi Behzad, Design of Analog CMOS Integrated Circuits, Tata McGraw hill, New Delhi , 2002, ISBN0-07-052903-5.

[8] Roy Ludvig Sigvartsen, An Analog Neural Network With On-Chip Learning Thesis Department of informatics, University of Oslo, 1994

[9] Isik Aybayetal , Classification of Neural Network Hardware, Neural Network World ,IDG Co., Vol6 No1, 1996, pp.11-29 ,

[10] Vincent F. Koosh Analog Computation and Learning in VLSI PhD thesis California institute of technology, Pasadena, California.2001.

[11] E. Vittoz, et al.," The design of high performance analog circuits on digital CMOS chips," IEEE J. Solid State Circuit 20, 1985, pp.657-665

[12] S. Orcioni G. Biagetti, M. Conti, "A mixed signal fuzzy controller using current model circuits, " Analog Integrated Circuits Process. 38, 2004 , pp.215-231.

[13] F. Djeffal et al., "Design and Simulation of Nanoelectronic DG MOSFET Current Source using Artificial Neural Networks," Materials Science and Engineering C, vol. 27, 2007, pp. 1111-1116

[14] H. Ben Hommounda et al., "Neural Based Models Of Semiconductor Devices for Spice Simulator,"American J. of Applied Science, vol. 5, 2008, pp. 385-391

[15] Arne Heittmann, "An Analog VLSI Pulsed Neural Network for Image Segmentation using Adaptive Connection Weights" Dresden University of Technology, Department of Electrical Engineering and Information Technology,Dresden, Germany, 2000

[16] Wai-Chi Fang et al, "A VLSI Neural Processor for Image Data Compression using Self-Organisation Networks" IEEE Transactions on Neural Networks, Vol. 3, No. 3, May 1992, pp. 506-517

[17] R. Rojas, Neural Networks, Springer-Verlag, Berlin, 1996

[18] D. Nguyen a and B. Wid row, Improving the learning speed of 2-layer neural network by choosing initial values of the adaptive weights, IEEE First International Joint Conference on Neural Networks , 3, 21–26, (1990).

[19] R.A. Jacobs, Increased rates of convergence through learning rate adaptat ion , Neural Networks , 1 , 295–307, (1988).

[20] T. P. Vogl, J. K. Man gis, J.K. Rigler, W. T. Z ink and D .L. Alkon, Accelerating the convergence of the back-propagation method , Biological Cyberne tics, 59 , 257–263,(1988).