DESIGN AND ANALOG VLSI IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK

Prof. Bapuray.D.Yammenavar[1], Vadiraj.R.Gurunaik[2], Rakesh.N.Bevinagidad[3]and Vinayak.U.Gandage[4]

[1,2,3,4]Dept of Electronics & Communication, BLDEA's College of Engg & Tech, Bijapur, Visvesvaraya Technological University, Karnataka, India.

bapurayg@gmail.com[1], vadirajgurunaik@yahoo.com[2], rbevinagidad@yahoo.com[3] and vinayakug@gmail.com[4]

ABSTRACT

Nature has evolved highly advanced systems capable of performing complex computations, adoption and learning using analog computations. Furthermore nature has evolved techniques to deal with imprecise analog computations by using redundancy and massive connectivity. In this paper we are making use of Artificial Neural Network to demonstrate the way in which the biological system processes in analog domain.

We are using 180nm CMOS VLSI technology for implementing circuits which performs arithmetic operations and for implementing Neural Network. The arithmetic circuits presented here are based on MOS transistors operating in subthreshold region. The basic blocks of artificial neuron are multiplier, adder and neuron activation function.

The functionality of designed neural network is verified for analog operations like signal amplification and frequency multiplication. The network designed can be adopted for digital operations like AND, OR and NOT. The network realizes its functionality for the trained targets which is verified using simulation results. The schematic, Layout design and verification of proposed Neural Network is carried out using Cadence Virtuoso tool.

KEYWORDS

Neural Network Architecture (NNA), Artificial Neural Network (ANN), Back Propagation Algorithm (BPA), Artificial Intelligence (AI), Neuron Activation Function (NAF).

1. INTRODUCTION

Neural Computers mimic certain processing capabilities of the human brain. Computing is an information processing paradigm inspired by biological system composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.

When we speak of intelligence it is actually acquired, learned from the past experiences. This intelligence though a biological word, is realized based on the mathematical equations, giving rise to the science of Artificial Intelligence (AI). To implement this intelligence artificial neurons are used.

Artificial Neural Networks (ANNs) learn by example. An ANN is configured for a specific application, such as pattern recognition function approximation or data classification through a learning process, learning in biological systems involves adjustments to the synaptic

connections that exist between the neurons. These artificial neurons, in this paper are realized by Analog components like multipliers, adders and differentiators. This is true of ANNs as well.

1.1 Brain versus Computers

- There are approximately 10 billion neurons in the human cortex, compared with 10 of thousands of processors in the most powerful parallel computers.
- Each biological neuron is connected to several thousands of other neurons, similar to the connectivity in powerful parallel computers.
- Lack of processing units can be compensated by speed. The typical operating speeds of biological neurons is measured in milliseconds (10^{-3} s), while a silicon chip can operate in nanoseconds (10^{-9} s).
- The human brain is extremely energy efficient, using approximately 10^{-16} joules per operation per second, whereas the best computers today use around 10^{-6} joules per operation per second.
- Brains have been evolving for tens of millions of years, computers have been evolving for tens of decades

2. Biological Neuron Model

The human brain consists of a large number [2]; more than a billion of neural cells that process information. Each cell works like a simple processor. The massive interaction between all cells and their parallel processing only makes the brain's abilities possible.

Dendrites: are branching fibers that extend from the cell body or soma. Soma or cell body of a neuron contains the nucleus and other structures, support chemical processing and production of neurotransmitters.

Axon: It is a singular fiber carries information away from the soma to the synaptic sites of other neurons (dendrites and somas), muscles, or glands. Axon hillock is the site of summation information. At any for incoming moment, the collective influence of all neurons that conduct impulses to a given neuron will determine whether or not an action potential will be initiated at the axon hillock and propagated along the axon.

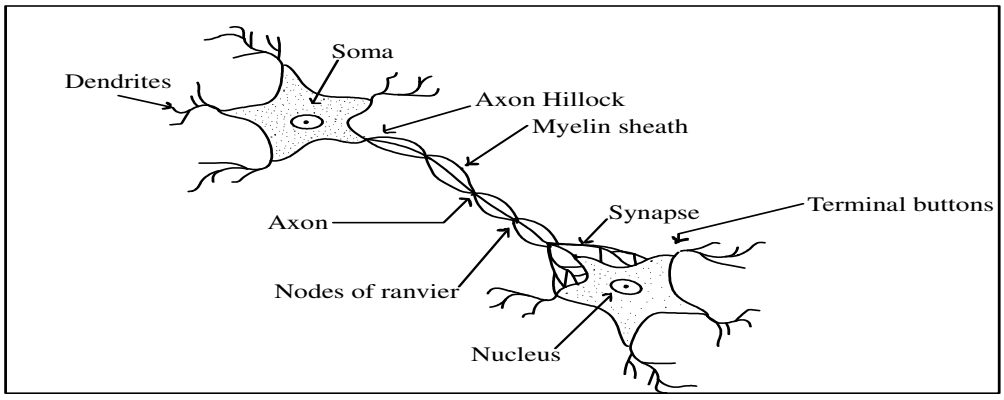

Fig.1 Structure of Biological Neuron

Myelin Sheath: consists of fat-containing cells that insulate the axon from electrical activity. This insulation acts to increase the rate of transmission of signals. A gap exists between each

myelin sheath cell along the axon. Since fat inhibits the propagation of electricity, the signals jump from one gap to the next.

Nodes of Ranvier : are the gaps (about 1μm) between myelin sheath cells long axons are since fat serves as a good insulator, the myelin sheaths speed the rate of transmission of an electrical impulse along the axon.

Synapse: is the point of connection between two neurons or a neuron and a muscle or a gland. Electrochemical communication between neurons takes place at these junctions.

Terminal Buttons: of a neuron are the small knobs at the end of an axon that release chemicals called neurotransmitters.

2.1 Artificial Neuron Model

An artificial neuron [2] is a mathematical function conceived as a simple model of a real (biological) neuron. This is a simplified model of real neurons, known as a Threshold Logic Unit.

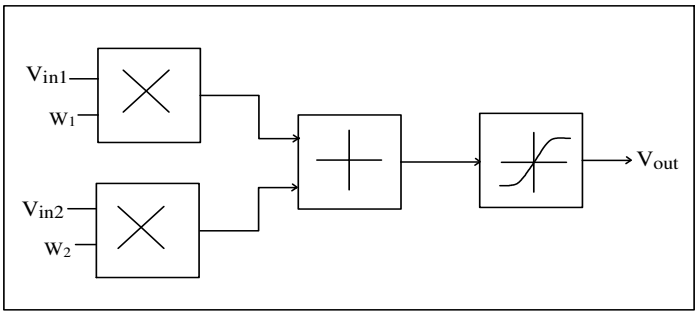

Fig.2 Mathematical model of Neuron

- A set of input connections brings in activations from other neurons.
- A processing unit sums the inputs, and then applies a non-linear activation function (i.e. squashing / transfer / threshold function).
- An output line transmits the result to other neurons.

2.1.1 Gilbert cell multiplier

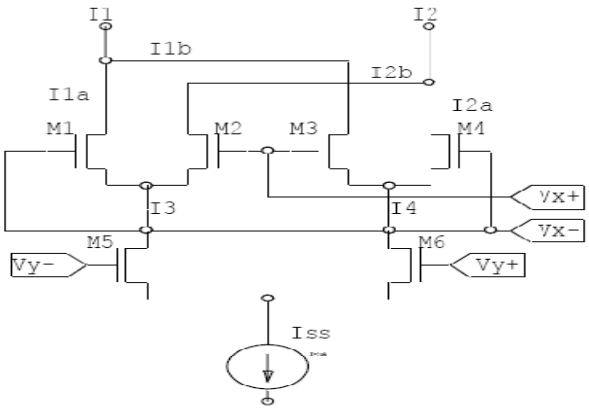

Fig.3 Gilbert cell.

In figure 4.3 the basic Gilbert cell structure is presented [1]. Assuming all transistors are biased in the saturation region and obey the ideal square law equation and that devices are sized and matched so that the transconductance parameters satisfy $K_1=K_2=K_3=K_4=Ka$ and $K_5=K6=K_b$.

Defining the output current $I_0=I_2-I_1=-(I_{2b}+I_{2a})-(I_{1a}+I_{1b})$, it can be shown that

$$I_o = \sqrt{2K_a}V_x\left(\sqrt{I_3}\sqrt{1-\frac{K_aV_x^2}{2I_3}}-\sqrt{I_4}\sqrt{1-\frac{K_aV_x^2}{2I_4}}\right)$$

If we demand

$$\frac{K_aV_x^2}{2I_3}\ll 1 \qquad \text{and} \qquad \frac{K_aV_x^2}{2I_4}\ll 1$$

It follows that I_o depends linearly on V_x

$$I_o \cong \sqrt{2K_a}\left(\sqrt{I_3}-\sqrt{I_4}\right)V_x$$

While the currents I_3, I_4 can be expressed as by

$$V_y = \frac{1}{\sqrt{K_b}}\left(\sqrt{I_3}-\sqrt{I_4}\right)$$

Substituting V_y and Io expression, it follows that

$$\boxed{I_o = \sqrt{2K_aK_b}V_xV_y}$$

The output current yields an ideal analog multiplier [10]. Notice that since both I_3 and I_4 are I_{SS} and V_Y dependent, both V_Y and V_X must be kept small to maintain good linearity.

2.1.2 CMOS Differential Amplifier as NAF

A differential amplifier [3] is one that amplifies the difference between two voltages and rejects the average or common mode value of the two voltages.

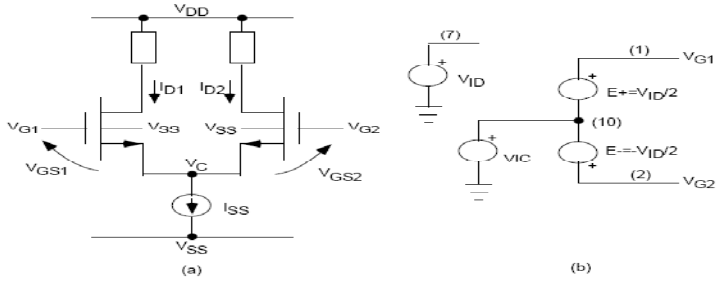

Fig.4 General MOS Differential Amplifier: (a) Schematic Diagram, (b) Input Gate voltage implementation.

The Differential input is given by:

$$V_{ID} = V_{G1} - V_{G2} = (V_{GS1} + V_C) - (V_{GS2} + V_C) \quad --(1)$$

$$V_{ID} = V_{GS1} - V_{GS2} = (V_{GS1} - V_{TN}) - (V_{GS2} - V_{TN}) = \sqrt{\frac{2I_{D1}}{\beta_1}} - \sqrt{\frac{2I_{D2}}{\beta_2}} \quad --(2)$$

The common-mode input signal is given by:

$$V_{IC} = \frac{V_{G1} + V_{G2}}{2} \quad --(3)$$

The input voltages in term of V_{ID} and V_{IC} are given by

$$V_{G1} = V_{IC} + V_{ID}/2 \quad --(4)$$

$$V_{G2} = V_{IC} - V_{ID}/2 \quad --(5)$$

Two special cases of input gate signals are of interests: pure differential and pure common mode input signals. Pure differential input signals mean V_{IC}=0, from equation (4) and (5);

$$V_{G1} = V_{ID}/2$$
$$V_{G2} = -V_{ID}/2$$

This case is of interest when studying the differential gain of differential amplifier, see figure.5 Pure common-mode input signals mean V_{ID}=0, from equation (4) and (5);

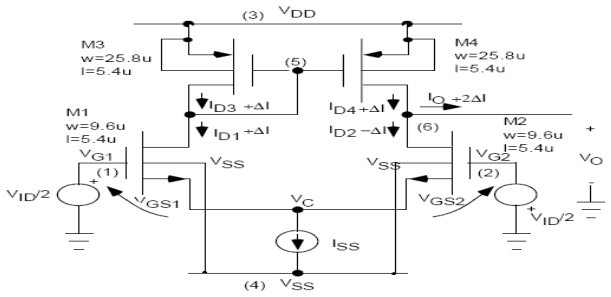

Fig.5 Differential Amplifier Implementation:

An active load acts as a current source. Thus it must be biased such that their currents add up exactly to I_{SS}. In practice this is quite difficult. Thus a feedback circuit is required to ensure this equality. This is achieved by using a current mirror circuit as load. The current mirror consists of transistor M3 and M4. One transistor (M3) is always connected as diode and drives the other transistor (M4). Since V_{GS3}=V_{GS4}, if both transistors have the same β, then the current I_{D3} is mirrored to I_{D4}, i.e., I_{D3}=I_{D4}.

The advantage of this configuration is that the differential output signal is converted to a single ended output signal with no extra components required. In this circuit, the output voltage or current is taken from the drains of M2 and M4. The operation of this circuit is as follows. If a differential voltage V_{ID}=V_{G1}-V_{G2}, is applied between the gates, then half is applied to the gate-source of M1 and half to the gate-source of M2. The result is to increase I_{D1} and decrease I_{D2} by equal increment, ΔI. The ΔI increase I_{D1} is mirrored through M3-M4 as an increase in I_{D4} of ΔI. As a consequence of the ΔI increase in I_{D4} and the ΔI decrease in I_{D2} , the output must sink a

current of 2ΔI. The sum of the changes in I_{D1} and I_{D2} at the common node V_C is zero. That is, the node V_C is at an ac ground. From Eq (4) and Eq (5) for pure differential input signal means the common-mode signal V_{IC} is zero. That is, the input signals are $V_{G1}=V_{ID}/2$ and $V_{G2}=-V_{ID}/2$. This is shown in Figure.5. The transconductance of the differential amplifier is given by:

$$g_{mD} = \frac{\Delta I_o}{\Delta V_{ID}} = \frac{2\Delta I}{\Delta V_{ID}} = \frac{\Delta I}{\Delta V_{ID}/2} = \frac{\Delta I}{V_{gs1}} = g_{m1}$$

That is the differential amplifier has the same transconductance as a single stage common source amplifier.

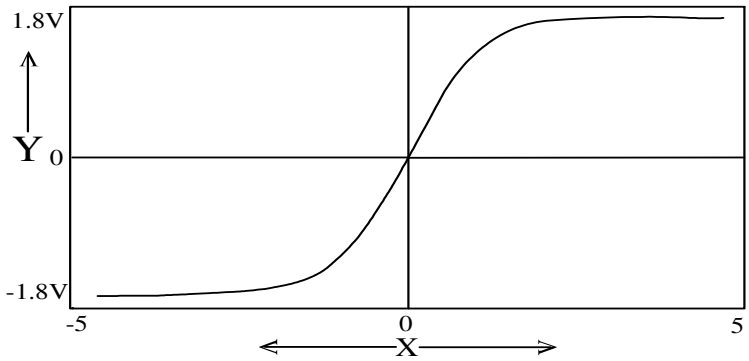

Fig.6 DC response of CMOS Differential Amplifier

3. Back Propagation Algorithm

In this paper we are using back propagation algorithm [5]-[6] as a training Algorithm for the proposed neural network. Back-propagation network (BPN) is the best example of a parametric method for training supervised multi-layer perception neural network for classification. BPN like other SMNN (supervised multi layer feed forward neural network) models has the ability to learn biases and weights. It is a powerful method to control or classify systems that use data to adjust the network weights and thresholds for minimizing the error in its predictions on the training set. Learning in BPN employs gradient-based optimization method in two basic steps: to calculate the gradient of error function and to compute output by the gradient.

BPN compares each output value with its sigmoid function in the input forward and computes its error in BPN backward. This is considerably slow, because biases and weights have to be updated in each epoch of learning. Preprocessing in real world environment focuses on data transformation, data reduction, and pre-training. Data transformation and normalization are two important aspects of pre-processing.

The mathematical equations of back propagation Algorithm are given as follows

$$E = \frac{1}{2}(a_i^2 - d_i)^2 \qquad (1)$$

Where E is the error, a_i is actual output of neural network and d_i is the desired output. This process of computing the error is called a forward pass. How the output unit affects the error in the *ith* layer is given by differentiating equation (1) we get

$$\frac{\partial E}{\partial a_i} = (a_i^2 - d_i) \qquad (2)$$

The equation (2) can be written in the other form as

$$\partial_i - (a_i^2 - d_i)d(a_i^2) \tag{3}$$

Where d(*ai*) is the differentiation of the *ai.* The weight update is given by

$$\Delta w_{ij} = \eta \partial_i a_i^1 \tag{4}$$

Where a^1_i is the output of the hidden layer or input to the output neuron and η is the learning rate. This error has to propagate backwards [7] from the output to the input. The ∂ for the hidden layer is calculated as

$$\partial_{hiddenlaye} = d(a_i^1)\sum w_{ij}\partial_i \tag{5}$$

Weight update for the hidden layer [8] with new, will be done using equation (3). Equations (1)-(5) depend on the number of the neurons present in the layer and the number of layers present in the network. The block diagram of 1:3:1 neural network with back propagation is shown in the following Fig.7

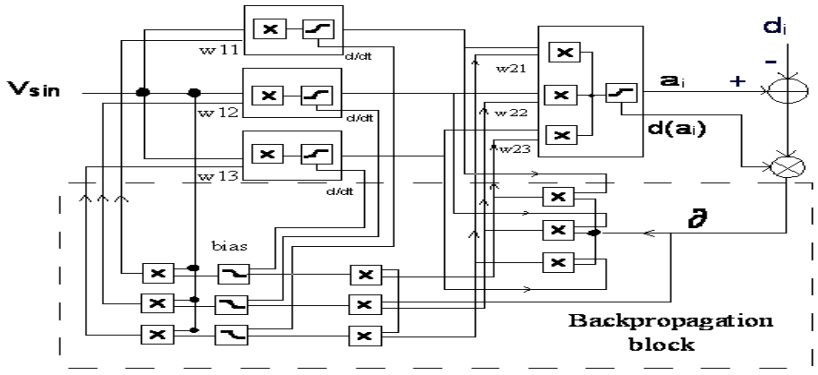

Fig.7 Neural network (1:3:1) with Backpropagation Algorithm

4. Neuron Design

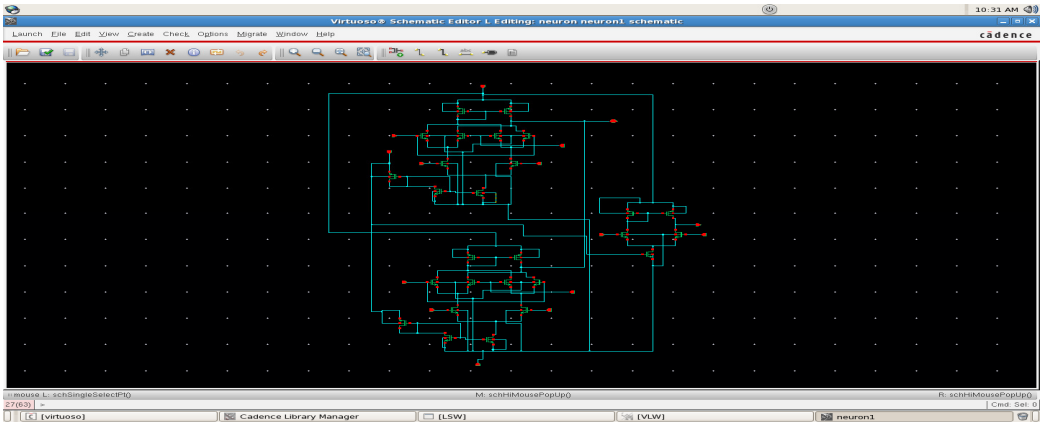

Fig.8 Schematic of Neuron

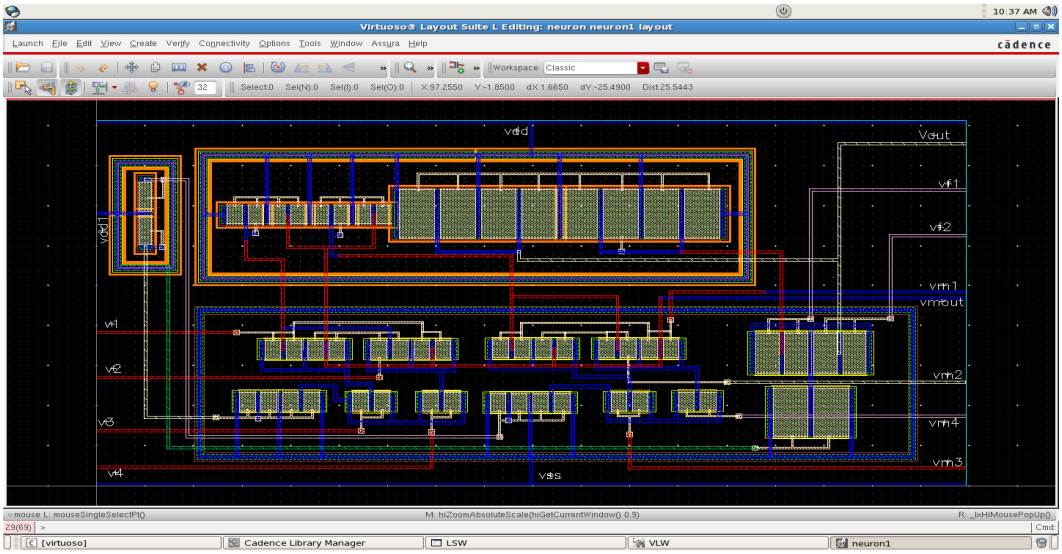

Fig.9 Layout of Neuron

The Fig.9 shows the layout of a neuron. The total size of the Neuron cell is approximately 90x45u.The layout is simulated with parasitic, and its results have been matched with the simulated results of schematic. The obtained various results of neuron are discussed in preceding sections.

4.1 Implementation of 1:3:1 neural network

The proposed 1:3:1 Neural network is shown below has three layers as input, hidden and output layers respectively. An input Vin is connected to the three neurons in the hidden layer through weights w11 to w13. The outputs of the hidden layer are connected to the output layer through weights w21 to w23.

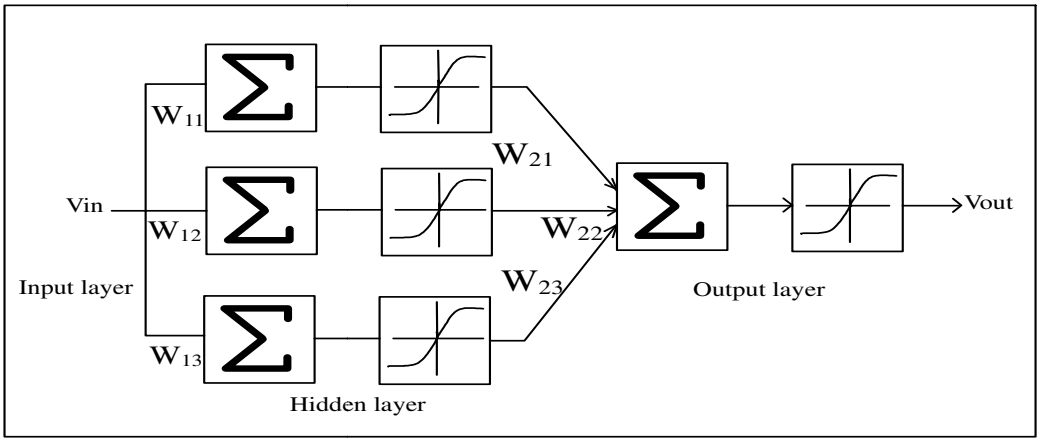

Fig.10 1:3:1 Neural Network

The network is trained with a sine wave of 500KHz frequency and target signal applied was of same frequency, the neural network was able to learn it and has reproduced signal of frequency same as that of target. The figure 12(a) shows input and output waves respectively.

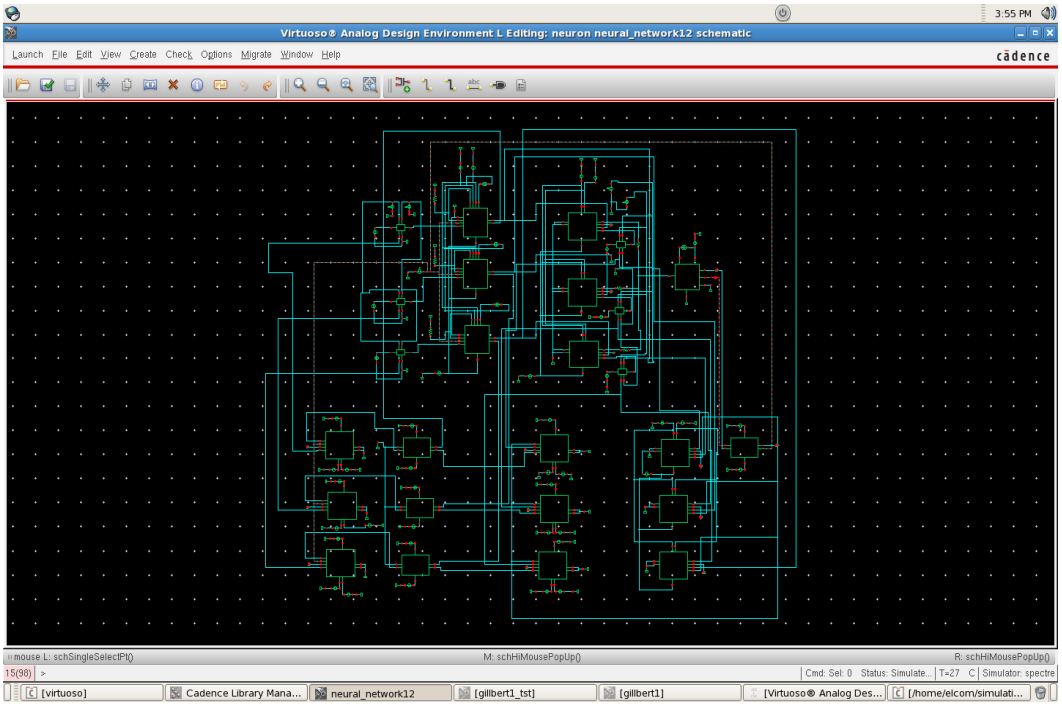

Fig .11 1:3:1 Neural Network

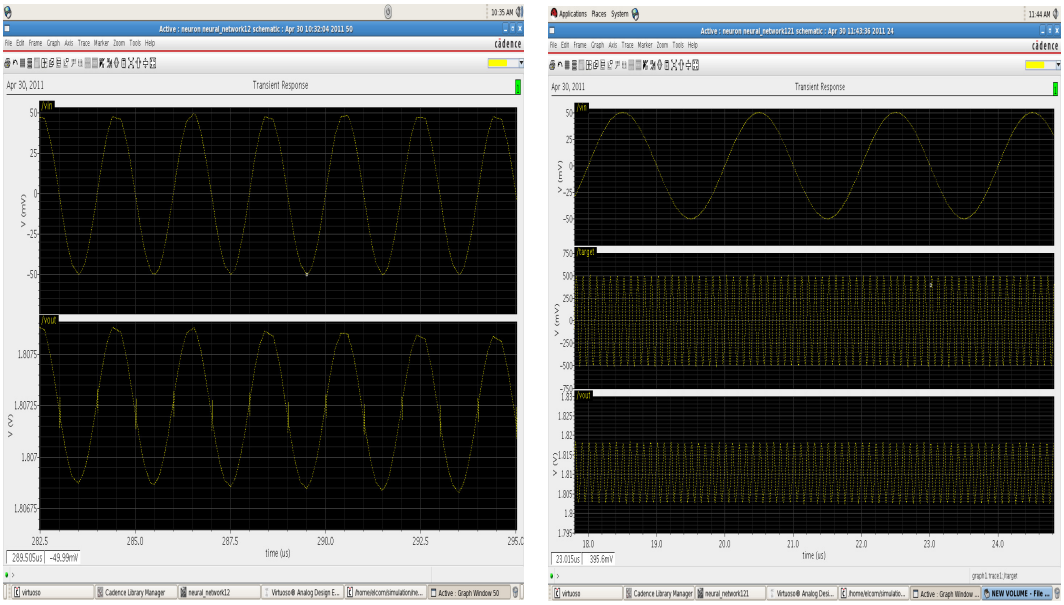

Fig.12: (a) Function approximation I/O wave forms with same frequency. (b)Function Approximations (Frequency Multiplication).

In the second case we trained the neural network with an input frequency of 500 KHz and target with 10MHz frequency. The network produced the learned 10MHz frequency as that of target frequency. This is shown in the figure 12(b), with input, target and output waveforms respectively. This validates the frequency multiplication operation of neural network.

5. Analog Operations

5.1. Signal Amplification

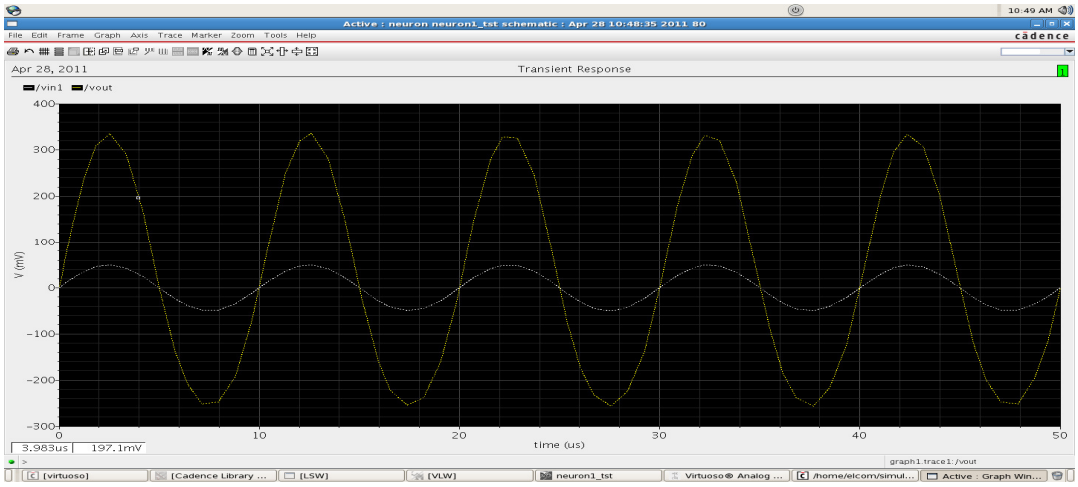

Fig.13: Transient Response of signal amplification

The linearity property of NAF(Fig.6) can be used for the signal amplification, from figure.13, it can be observed that the amplitude of input signal is ±50mV with a frequency of 500KHz, and produced output swing is of 580mV with maintaining the constant frequency same as that of input. The gain of neuron amplifier is "5.8". From this we conclude that neuron can be used for small signal amplification purpose.

5.2.Amplitude Modulation

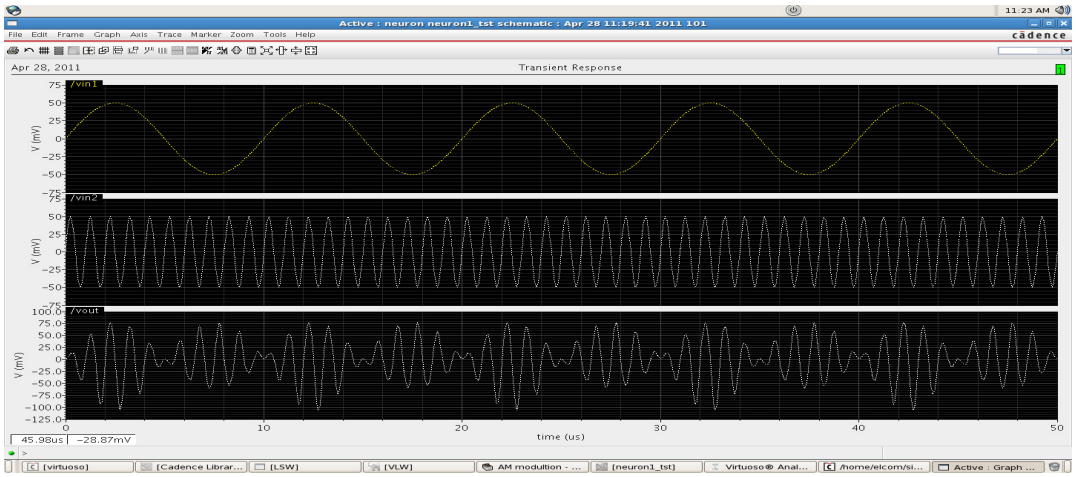

Fig 14: Transient Response of AM modulation

One of the other applications of neuron is Amplitude Modulation, as analog multiplier is the important building block of neuron, so its property can be used for amplitude modulation. The figure.14 shows its transient response and modulation index obtained is "2.5".From the above discussions the analog operation of Neural Network can be validated.

6. Digital Operations

Neural architecture is also adopted and verified for Digital operations like OR, AND and NOT. These operations are obtained by varying three main properties of neuron; they are “weight, bias voltage, and input terminals of NAF”. The digital buffer is used at the output stage of neuron for digital operations and its test bench is shown in figure.15. The simulated results of OR, AND and NOT gates are shown in figure.16.

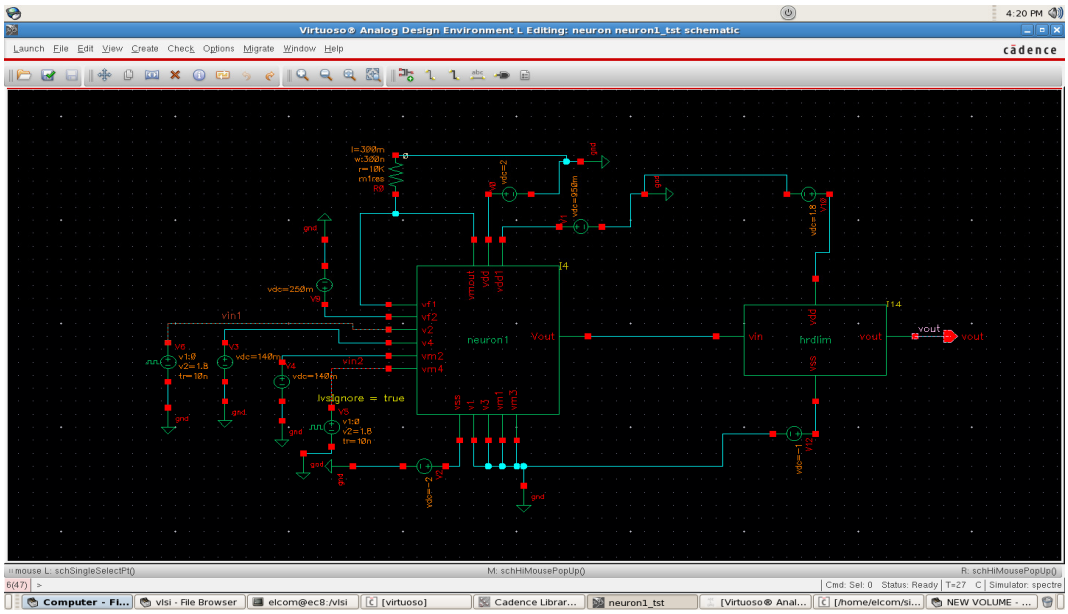

Fig.15 Test bench for digital operation

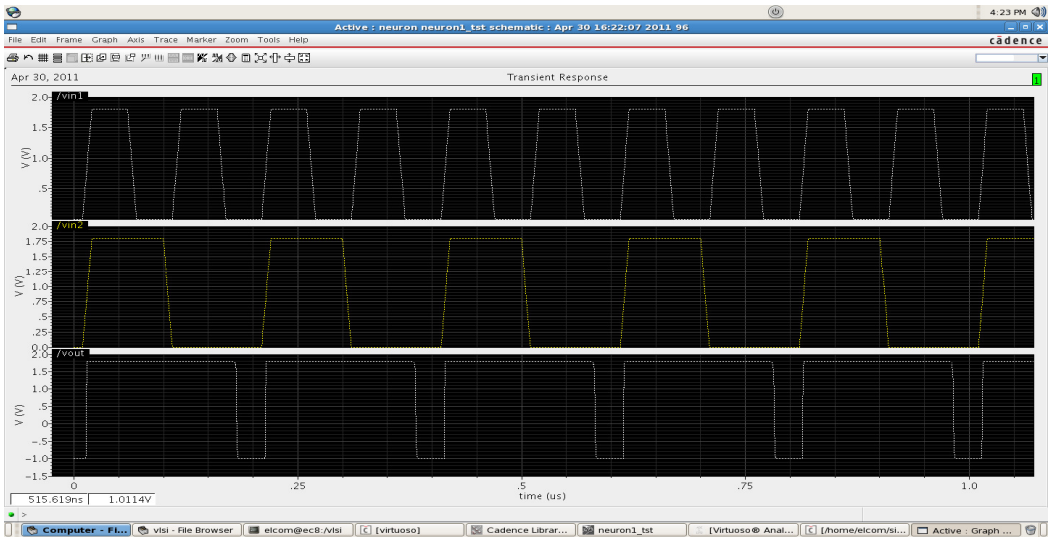

Fig 16(a): OR gate I/O waveforms

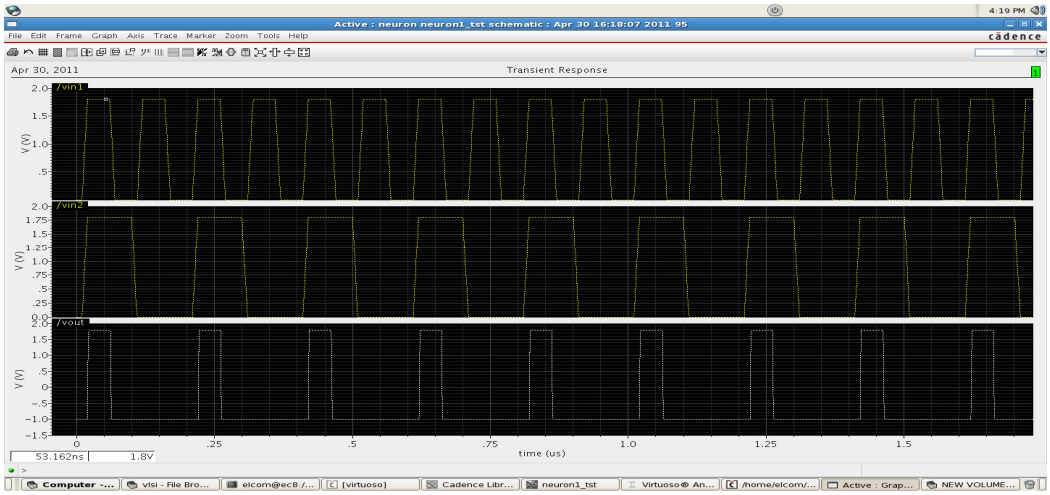

Fig 16(b): AND gate I/O waveforms

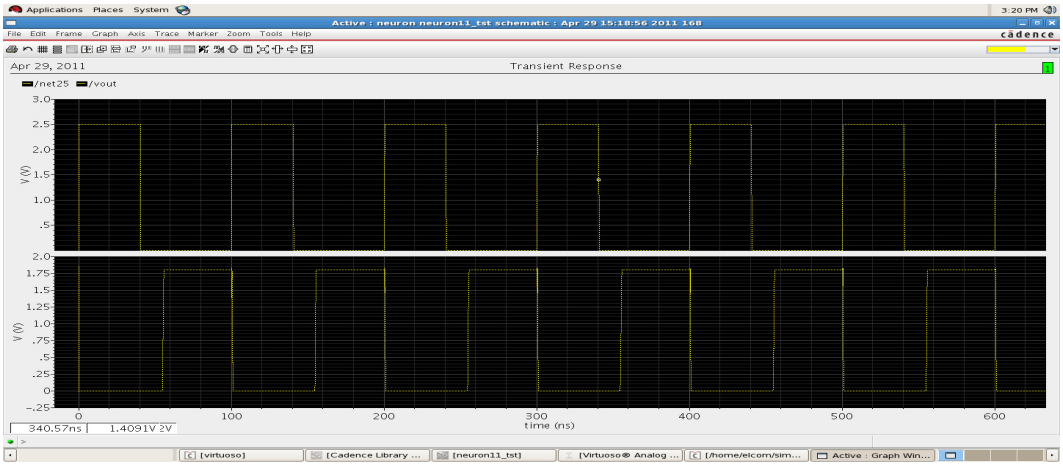

Fig 16(c): NOT gate I/O waveforms

6. Applications of Neural Network

Artificial neural networks are often used for applications in recent days, where it is difficult to state explicit rules. Often it seems easier to describe a problem and its solution by giving examples; if sufficient data is available a neural network can be trained.

Clustering:

A clustering algorithm explores the similarity between patterns and places similar patterns in a cluster. Best known applications include data compression and data mining.

Classification/Pattern recognition:

The task of pattern recognition is to assign an input pattern (like handwritten symbol) to one of many classes. This category includes algorithmic implementations such as associative memory.

Function approximation:

The tasks of function approximation are to find an estimate of the unknown function subject to noise. Various engineering and scientific disciplines require function approximation.

Prediction Systems:

The task is to forecast some future values of a time-sequenced data. Prediction has a significant impact on decision support systems. Prediction differs from function approximation by considering time factor. System may be dynamic and may produce different results for the same input data based on system state (time).

Brain modeling:

The scientific goal of building models of how real brains work. This can potentially help us understand the nature of human intelligence, formulate better teaching strategies, or better remedial actions for brain damaged patients.

Artificial System Building:

The engineering goal of building efficient systems for real world applications. This may make machines more powerful, relieve humans of tedious tasks, and may even improve upon human performance.

7. Future work

The conventional computers are good for fast arithmetic and do what programmer programs, ask them to do. The conventional computers are not so good for interacting with noisy data or data from the environment, massive parallelism, fault tolerance, and adapting to circumstances.

Signal compression can be done in analog domain using neural networks, the main difference between analog and digital signal processing is, analog signal processing does not require analog to digital converter, where as digital signal processing require analog to digital and digital to analog converter. The problem of quantization noise can be avoided by analog signal processing with the help of neural network.

8. Conclusion

A VLSI implementation of a neural network has been demonstrated in this paper. Analog weights are used to provide stable weight storage with refresh circuit. Analog multipliers are used as synapse of neural networks. Although the functions learned were analog, the network is adoptable to accept digital inputs and provide digital outputs for learning other functions. Network designed has been successfully adopted for digital operations like AND, OR and NOT.

The Network proposed has following features.

- Gilbert cell multiplier was designed with maximum input range of 100mV and maximum output swing of 800mV.
- Neuron Activation function was designed for input range of ±1.8V and output range of ±1.7V. A Neural architecture was proposed using these components.
- The Neural Architecture works on the supply voltage ±1.8V with the output swing of ±1.6V.
- Back Propagation algorithm was used for the training of the network.
- The designed neural architecture had a convergence time of 200 ns.
- The Neural network shown to be useful for digital and analog operations.
- The architecture proposed can be used with other existing architecture for neural processing.
- Neural network was able to learn and reproduce the target waves; this validates the on chip learning in analog domain.

REFERENCES

[1]. Gilbert Multiplier by Ari Sharon, aris@cs, Ariel Zentner , relz@cs, Zachi Sharvit, zachi@cs, Yaakov Goldberg, yaakov@cs.

[2]. Bose N. K., Liang P., "Neural Network Fundamentals with graphs, algorithms and Application", Tata McGraw hill, New Delhi, 2002, ISBN 0-07-463529-8

[3]. Razavi Behzad, "Design of Analog CMOS Integrated Circuits", Tata McGrawhill, New Delhi,2002, ISBN 0-07-052903-5

[4]. Bernabe Linares-Barranco et al., "A Modular T-Mode Design Approach for Analog NeuralNetwork Hardware Implementations", IEEE Journal of Solid-state Circuits. Vol. 27, no. 5, May1992, pp. 701-713

[5]. Hussein CHIBLE, "Analysis And Design Of Analog Microelectronic Neural Network Architectures With On-Chip Supervised Learning" Ph.D. Thesis in Microelectronics, University of Genoa, 1997 Isik Aybay et al, "Classification of Neural Network Hardware", Neural Network World, IDG Co.,Vol 6 No 1, 1996, pp. 11-29

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

[7]. Roy Ludvig Sigvartsen, "An Analog Neural Network with On-Chip Learning" Thesis Department of informatics, University of Oslo, 1994 Chun Lu, Bing-xue Shi and Lu Chen, "Hardware Implementation of an Analog Accumulator for On-chip BP Learning Neural Networks" Institute of Microelectronics, Tsinghua University Beijing, China 2002

[8]. Arne Heittmann, "An Analog VLSI Pulsed Neural Network for Image Segmentation using, Adaptive Connection Weights" Dresden University of Technology, Department of Electrical

[9]. European Journal of Scientific Research ISSN 1450-216X Vol.27 No.2 (2009), pp.199-216

[10]. Engineering and Information Technology, Dresden, Germany, 2000 Shai, Cai-Qin. Geiger, Randy L. "A 5-v CMOS Analog Multiplier" IEEE Journal of solid state circuits Vol sc22 No.6 December 1987, pp. 1143-1146