iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84393

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

CMOS Logic Design Using Neural Networks

Columbus Enslen[1], Damu Radhakrishnan[2]
Division of Engineering Programs, State University of New York at New Paltz, New Paltz, New York, USA

Abstract: ***This paper presents the design of digital logic functions using analog neural network principles. The logic gates are implemented as a single-layer perceptron neuron: MOS transistors fed by inputs with widths proportional to synaptic weights add up drain currents at a shared node via Kirchoff's current law, MOS transistors biased by a reference voltage to provide the threshold offset, and a cascaded CMOS inverter pair to perform a step activation function. This neuromorphic approach is demonstrated by designing a NOT gate, a NAND gate and an 1-bit full adder. The full adder resolves the non-linearly separable XOR problem by cascading two neurons; a majority-gate Carry output neuron followed by a Sum output neuron that uses the Carry as a double weighted inhibitory input. TSMC 180nm CMOS process is used for the implementation. Electric VLSI tools are used for making transistor schematics and layout. Simulations are carried out using LTspice. The 23 transistor neural network full adder achieved propagation delays less than 0.5ns and 0.27ns for the sum and carry outputs respectively, with an average power of 1.3mW. For comparison purposes, a fully complementary CMOS full adder using 36 transistors was also designed.***

Keywords: Neural networks, Perceptron, Full adder, CMOS design, Electric VLSI

I. INTRODUCTION

Artificial Neural Networks (ANN) have gained wider applications in the design and optimization of digital circuits in recent years. They are inspired by biological structures in the brain. The use of ANNs has simplified and increased the response time in systems due to parallel data processing. Mcculloch and Pitt studied the nervous system and showed that neural events and the relations among them can be treated by means of propositional logic [1]. Neural networks are used in many applications, such as pattern recognition, human speech diagnosis and identification, conversion of black and white images to color images, earthquake prediction etc. [2-6].

Each neural network consists of a number of layers, the input layer, the hidden layer, and the output layer. Each layer consists of a neural network of a number of artificial neurons [7,8]. Each neuron has a threshold, a weight W, and a bias b. Human brain is made up of cells called neurons. There are a total of 10^{11} neurons which are interconnected with 10^4 connections per neuron [2]. Neurons (or nerve cells) are electrically excitable cells within the nervous system, able to fire electric signals, called action potentials, across a neural network [9]. Each neuron is composed of dendrites, soma (cell body), axons and synapses. Dendrites are made up of nerve filaments that transmit electrical signals to the soma. Soma receives signals and applies a threshold value on them. Ultimately, the axon, which is a long nerve filament, carries these signals from the body to other neurons [2,3]. Synapse is the junction between the axons of one nerve cell and dendrites of other neurons. Figure 1 shows a biological neuron.

The structure of artificial neurons is similar to that of biological neurons, with several inputs and one output [7]. Each artificial neuron contains inputs, weights, bias and threshold values. A perceptron is one of the simplest forms of artificial neuron used in neural networks. The output of a perceptron is given as:

$$y = f\left(\sum_{i=1}^{n}(x_i w_i + b)\right),$$

where n denotes the number of inputs, b is the bias, w_i is the weight, x_i is the input value and f is an activation function. A Classic perceptron uses a step activation function represented as:

$$f(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0 \end{cases}$$

This relationship is illustrated in Figure 2.

In this paper, logic gates and full adders are designed using neural networks with constant weights and a fixed bias. They are implemented in CMOS using Electric VLSI software and simulations are carried out by LTspice.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

II. DESIGN OF LOGIC GATES AND FULL ADDER

In this section, two logic gates, NOT and NAND, will be designed as individual neurons (perceptron gates). This will be followed by the design of a full adder. The weights and bias values of the neural networks are calculated in each case. In the designed networks, the input is fed in binary form. The bias values of each neuron are connected to a tunable reference voltage.

A. NOT gate

A NOT gate can be modeled as:

$$y = \begin{cases} 1 & if\ xw + b > 0 \\ 0 & otherwise \end{cases}$$

A logic diagram of the NOT gate is shown in Figure 3, and its truth table is shown in Table I. From Table I, output = 1 when input is 0, and output = 0 when input = 1.

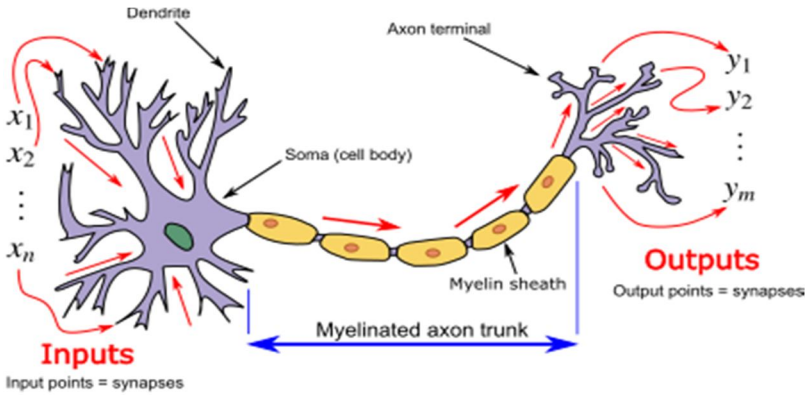


Fig. 1 A Biological neuron [9]

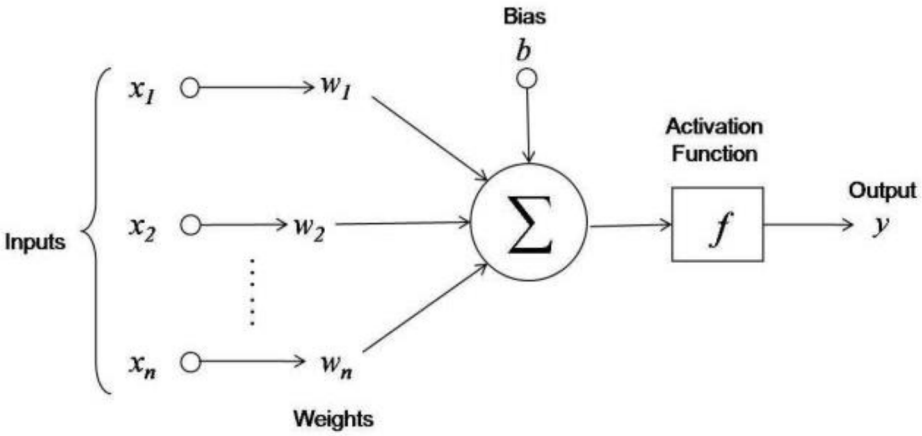


Fig. 2 Mathematical representation of a perceptron

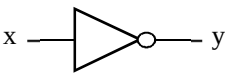


Fig. 3 A NOT gate

TABLE I TRUTH TABLE OF NOT GATE

Input x	Output y
0	1
1	0

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

To design the NOT gate as a perceptron, consider an arbitrary weight w = -1 and bias b = 0.5. Then the perceptron output is given by:

$$y = \begin{cases} 1 & if\ -x + 0.5 > 0 \\ 0 & otherwise \end{cases}$$

Substituting the values for x, we get: When x = 0, $-x + 0.5 = 0.5$, and y = 1, and when x = 1, $-1 + 0.5 = -0.5$, and y = 0. Hence, the perceptron behaves like a NOT gate. It may be noted that many Perceptron implementations are possible since many choices exist for w and b.

B. NAND Gate

A 2-input NAND gate and its truth table are shown in Figure 4 and Table II respectively. This NAND gate can be modeled as:

$$y = \begin{cases} 1 & if\ Aw_1 + Bw_2 + b > 0 \\ 0 & otherwise \end{cases}$$

As a first step let us consider the weights as: $W_1 = -1$, $W_2 = -1$, and b = 1.5.
Substituting the values from each row of Table II, we get:
For 00, $0\times(-1) + 0\times(-1) + 1.5 = 1.5$ and hence, y = 1.
For 01, $0\times(-1) + 1\times(-1) + 1.5 = 0.5$ and hence, y = 1.
For 10, $1\times(-1) + 0\times(-1) + 1.5 = 0.5$ and hence, y = 1.
For 11, $1\times(-1) + 1\times(-1) + 1.5 = -0.5$ and hence, y = 0.
Hence the truth table is satisfied. As in the case of the NOT gate, many perceptron implementations are possible for the NAND gate by simply selecting different weights and bias values.

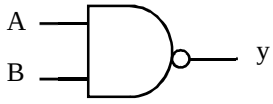


Fig. 4 A Two-input NAND gate

TABLE II. TRUTH TABLE OF TWO-INPUT NAND GATE

Inputs		Output
A	**B**	**y**
0	0	1
0	1	1
1	0	1
1	1	0

C. Full Adder

A full adder has 3 inputs A, B and C and two outputs Sum and Carry. A Spartan-3 FPGA implementation of a full adder as an artificial neural network using Verilog coding is given in [10]. A block diagram for the full adder is shown in Figure 5, and its truth table is shown in Table III.

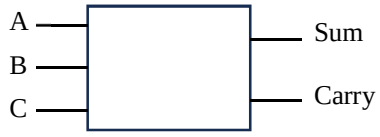


Fig. 5 Full adder

TABLE III. FULL ADDER TRUTH TABLE

Inputs			Outputs	
A	**B**	**C**	**Sum**	**Carry**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

The full adder can be modeled using two equations, one for Sum (f_1), and the other for Carry (f_2) as:

$$\text{Sum} = f_1 = \begin{cases} 1 & if\ Aw_1 + Bw_2 + Cw_3 + b_1 > 0 \\ 0 & otherwise \end{cases}$$

$$\text{Carry} = f_2 = \begin{cases} 1 & if\ Aw_1 + Bw_2 + Cw_3 + b_2 > 0 \\ 0 & otherwise \end{cases}$$

A modified set of equations is presented for the Sum output in [11]. This is based on Table III, where the sum can also be obtained as a function of Carry output as: Sum = A + B + C – 2 × Carry.
This is illustrated in Table IV. Based on Table IV, the Sum equation is modified as:

$$\text{Sum} = f_3 = \begin{cases} 1 & if\ Aw_1 + Bw_2 + Cw_3 - 2 \times Carry + b_3 > 0 \\ 0 & otherwise \end{cases}$$

To design the perceptron, let us select the weights. For simplicity, the weights for both Sum and Carry are selected the same. First, we design the Carry neuron. Consider the weights as: W_1= 1, W_2 = 1, W_3 = 1, and bias b_2 = -1.5.
The Carry output verification can now be done as:
Zero inputs at logic 1: 0×(1) + 0×(1) + 0×(1) - 1.5 = - 1.5 and hence Carry = 0.
Any one input at logic 1: 1×(1) - 1.5 = - 0.5 and hence Carry = 0.
Any two inputs at logic 1: 1×(1) + 1×(1) - 1.5 = + 0.5 and hence Carry = 1.
All three inputs at logic 1: 1×(1) + 1×(1) + 1×(1) - 1.5 = + 1.5 and hence Carry = 1.
Hence, the Carry output is satisfied.

TABLE IV FULL ADDER SUM AS A FUNCTION OF CARRY

Inputs			Outputs		Sum
A	**B**	**C**	**Sum**	**Carry**	**A+ B + C – 2 Carry**
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	1	1

For the Sum, we use the function f_3. A new bias value is needed. Let us use b_3 = - 0.5. Once again W_1 = 1, W_2 = 1, and W_3 = 1. The Sum output verification can now be done as:
All inputs at logic 0: 0×(1) + 0×(1) + 0×(1) - 2 ×(0) - 0.5 = - 0.5 and hence Sum = 0.
Any one input at logic 1: 1×(1) – 2×(0) - 0.5 = + 0.5 and hence Sum = 1.
Any two inputs at logic 1: 1×(1) + 1×(1) – 2×(1) - 0.5 = - 0.5 and hence Sum = 0.
All three inputs at logic 1: 1×(1) + 1×(1) + 1×(1) - 2×(1) - 0.5 = + 0.5 and hence Sum = 1.
The above satisfies the truth table for Sum.

III. CMOS IMPLEMENTATION

To implement the perceptron as a CMOS circuit, the inputs and the bias values are applied to the gates of MOS transistors. The CMOS circuit is used as a current-mode perceptron. Instead of operating the circuit in the digital mode (saturation mode), the circuit is operated in the analog mode as a voltage controlled current source under linear operating conditions [11]. Since the PMOS transistor Source is connected to the supply source VDD, it behaves like a current source injecting current (excitatory device with positive weight) at the output node whenever it is conducting. Similarly, the NMOS transistor behaves like a current sinking circuit (inhibitory device with negative weight) from the output node since its source is connected to GND.

Since the current through the transistors is a linear function of their aspect ratios (W/L), they can be used as weights in the perceptron. A wider transistor has a higher device transconductance which generates proportionately more drain current for the same gate voltage, thereby implementing multiplication of the input x_i by the weight w_i.

The drain leads of all transistors (both NMOS and PMOS) connect to a single shared node called the Z-node. By Kirchhoff's current law, all branch currents, both injected and sink, sum algebraically at this node. A separate transistor (dedicated bias transistor) with its gate tied to a tunable analog reference voltage (V_{bias}) injects or drains a fixed current from the Z-node, thereby implementing the bias term b. The reference voltage is adjusted to set the precise firing threshold of the neuron.

The Z-node output voltage feeds into a pair of CMOS inverters. The first inverter acts as the hard threshold, snapping the analog voltage to a digital level when it crosses the inverter switching point (≈ ½Vdd). The second inverter restores the output polarity and drives the output to full rail-to-rail swing thereby generating digital outputs.

A. NOT Gate Perceptron

From Section III A, the Z-node output for the NOT gate is given by: Z = x w + 0.5, where w = -1. A negative weight is achieved by an NMOS transistor connected between ground and Z-node. By selecting the NMOS transistor aspect ratio W/L = 1, we can obtain the weight of -1. Similarly, the positive bias 0f 0.5 is obtained by connecting PMOS transistor with aspect ratio W/L = 0.5 across Z-node and VDD. As per the MOSIS design rules, the minimum aspect ratio is 3/2. To satisfy this, the equation for Z can be multiplied by 3. This provides an NMOS transistor size W/L = 6/2, and the PMOS transistor size W/L = 3/2. To compensate for the lower hole mobility values in the PMOS transistor, its size is doubled to W/L = 6/2 (assuming that $\mu_n/\mu_p \approx 2$). A transistor schematic for the NOT gate is shown in Figure 6. Two inverters are connected at node Z to convert it to a digital output at y.

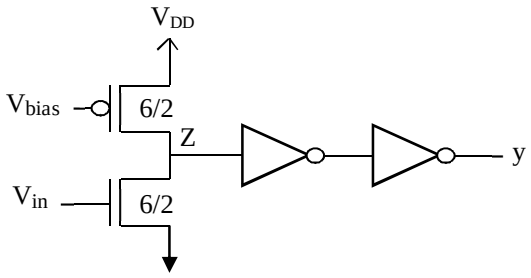


Fig. 6 NOT Gate designed as a perceptron

The transistor schematics were laid out in Electric VLSI using TSMC 180nm CMOS process. This is shown in Figure 7. The simulations were carried out at a supply voltage of 1.8V and an input signal frequency of 500MHz. Correct waveforms were observed up to a maximum V_{bias} = 0.9V, which implies that the PMOS transistor minimum V_{GS} = 0.9V. But the output delays are asymmetric at this bias voltage. Almost symmetrical delays (tPLH = tPHL = 0.11ns) were obtained for V_{bias} = 0V.

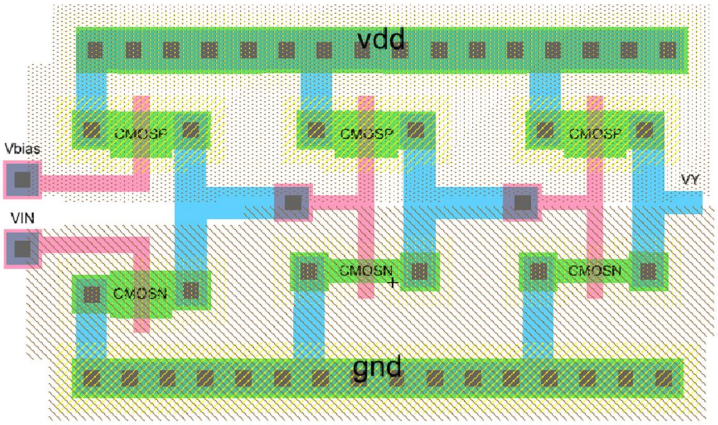


Fig. 7 Electric VLSI layout of CMOS NOT gate

B. NAND Gate Perceptron

From Sec III B, for the NAND gate Z = Aw_1 +B w_2 + b, where w_1 = -1, w_2 = -1 and b = 1.5. This represents two pull down NMOS transistors with W/L = 1 and a pull-up PMOS transistor with W/L = 1.5. The bias voltage is applied to the PMOS transistor. To satisfy MOSIS Design rules, we multiply the transistor sizes by 2. The transistor sizes used in our implementation are shown in Figure 8. The Correct simulation waveforms were observed for supply voltages of 1.8V and a maximum V_{bias} = 0.3V. This gives V_{GS} = -1.5V for the PMOS transistor. Once again, the delays are asymmetric at this bias voltage. Almost symmetrical delays (0.13ns) were obtained for V_{bias} = 0.05V.

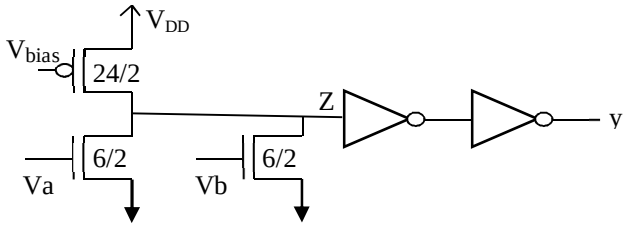


Fig. 8 Two-input NAND gate designed as a perceptron

C. Full Adder Perceptron

The perceptron equations for the full adder were obtained in Sec. II C and are given as:
Carry = $Aw_4 + Bw_5 + Cw_6 + b_2$, and Sum = $Aw_1 + Bw_2 + Cw_3 - 2 \times Carry + b_3$.
One set of values to satisfy these equations (from Sec. II C) are:
W_1 = 1, W_2 = 1, W_3 = 1, and b_2 = -1.5 for Carry, and W_1 = 1, W_2 = 1, W_3 = 1, and b_3 = -0.5 for Sum.

Transistor sizes are carefully chosen for proper functioning of the circuit. Many choices exist. All PMOS transistor sizes used on both Sum and Carry networks are 12/2. Only the NMOS transistor sizes are different. A 6/2 size NMOS transistor is used on the carry side for feeding the bias voltage. The bias voltage (Vb2) used is 1.8V. The NMOS transistor sizes used for the Sum are as follows:
The Carry output was fed to a 10/2 size NMOS transistor. The bias voltage (Vb3) is fed to the other NMOS transistor sized at 12/2, and the bias voltage used is 1.0V. The completed transistor schematics for the full adder is shown in Figure 9. This adder uses a total of 23 transistors. The transistor schematics and layout using Electric VLSI are shown in Figures 10 and 11 respectively. For the circuit to work correctly based on the transistor sizes used, the bias voltage Vb3 must be in the range from 0.6V to 1.1V.

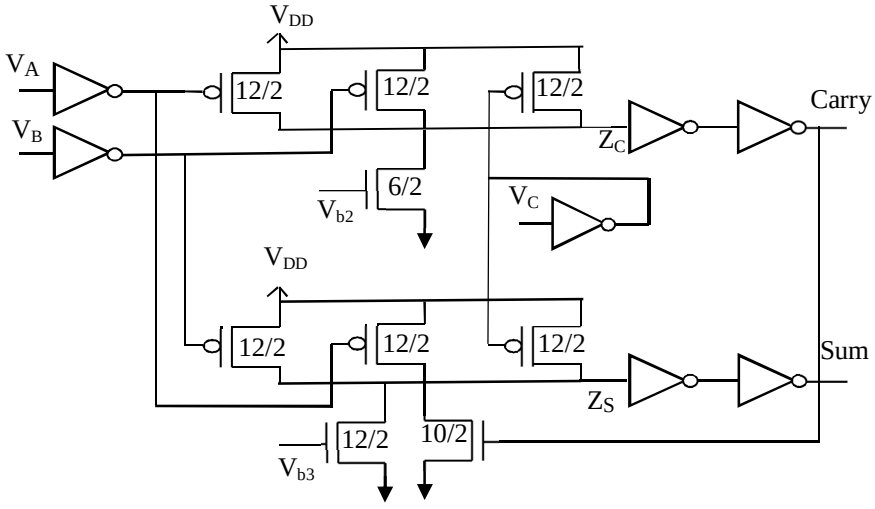


Fig. 9 Full adder designed as two perceptrons

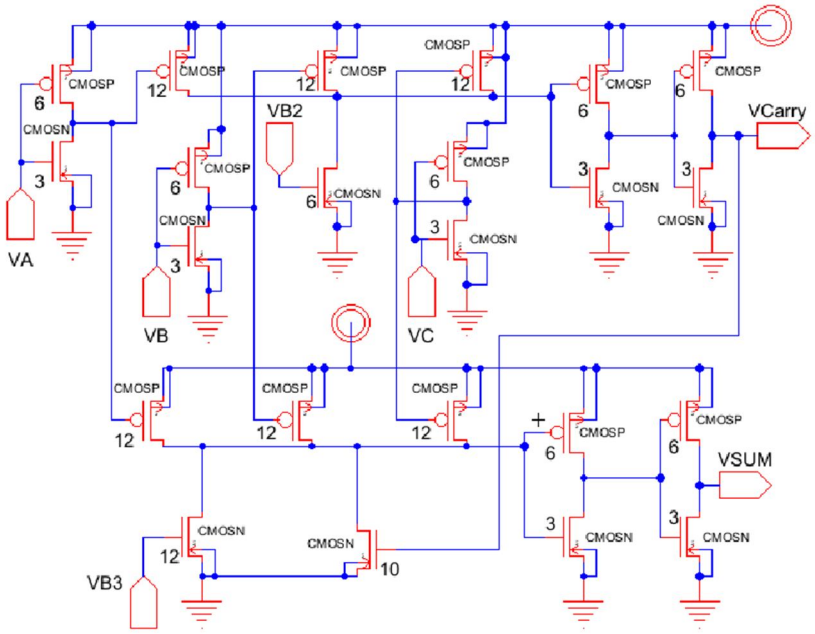


Fig. 10 Perceptron full adder transistor schematics in Electric VLSI

For comparison purposes, a fully complementary CMOS full adder (referred here as CMOS full adder) is also designed and laid out using Electric VLSI and simulated using LTspice. This adder uses a total of 36 transistors and its layout is shown in Figure 12.

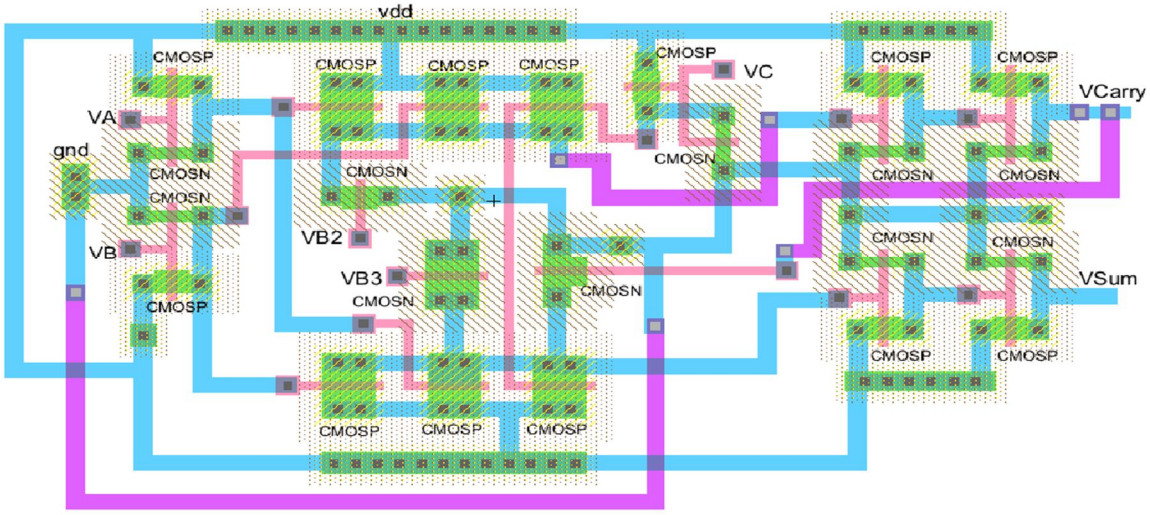


Fig. 11 CMOS layout of percepron full adder in Electric VLSI

IV. SIMULATION RESULTS

The LTSpice simulation waveforms for the NAND gate are shown in Figure 13. A 500MHz and 250MHz waveforms are fed to the two inputs. The perceptron based full adder layout waveforms are shown in Figure 14. The simulations were carried out at a frequency of 500MHz. The first three waveforms (Va, Vb and Vc) are the input waveforms, and the last two are output waveforms for Carry and Sum respectively. The simulation waveforms of the CMOS full adder are shown in Figure 15.

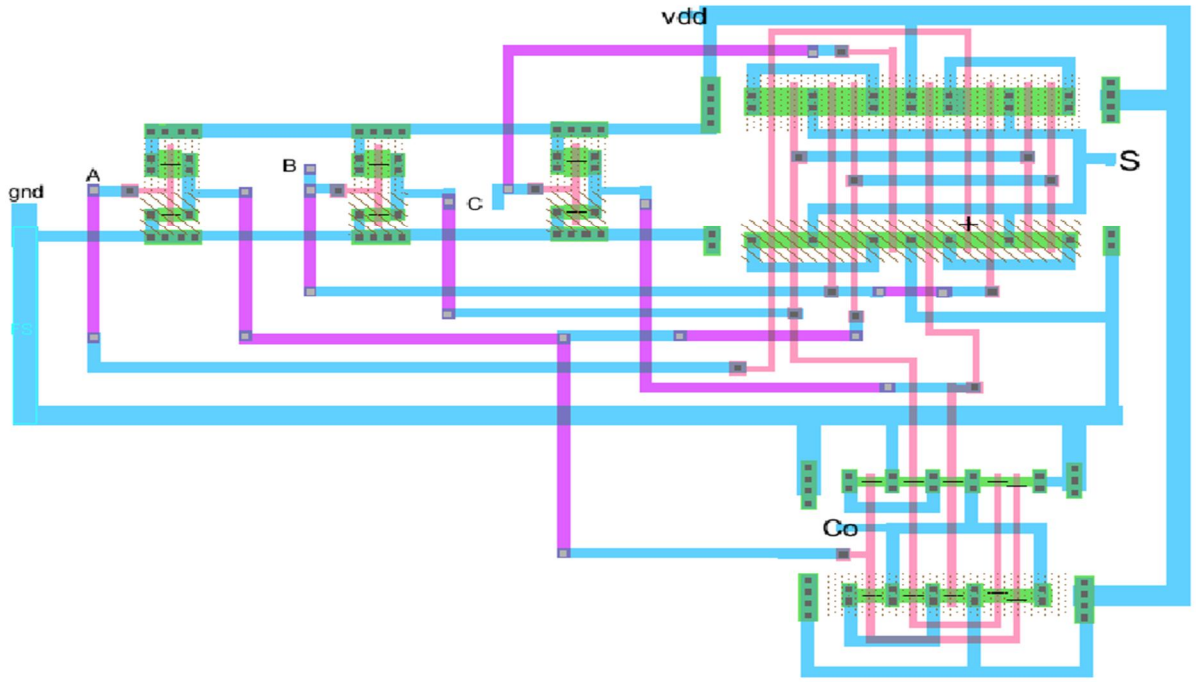


Fig. 12 Electric VLSI layout of CMOS full adder

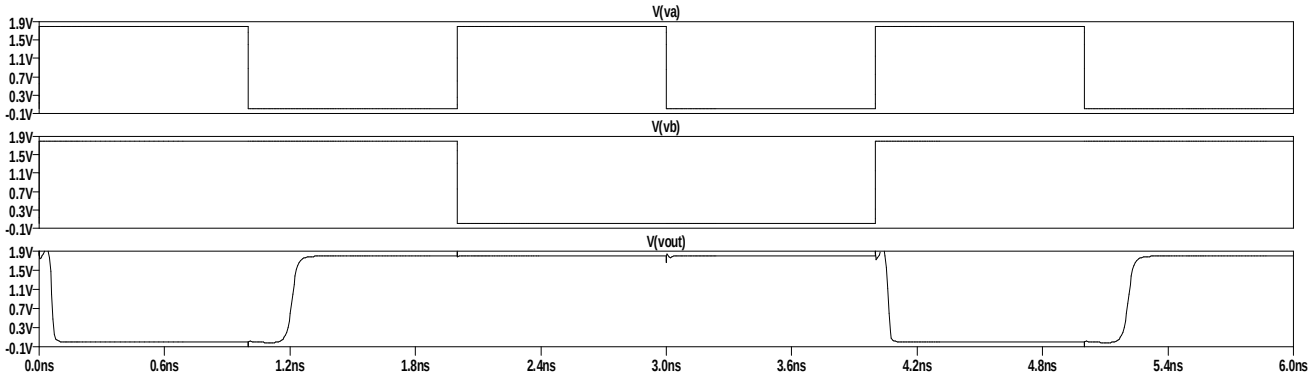


Fig. 13 Two-input NAND gate simulation waveforms

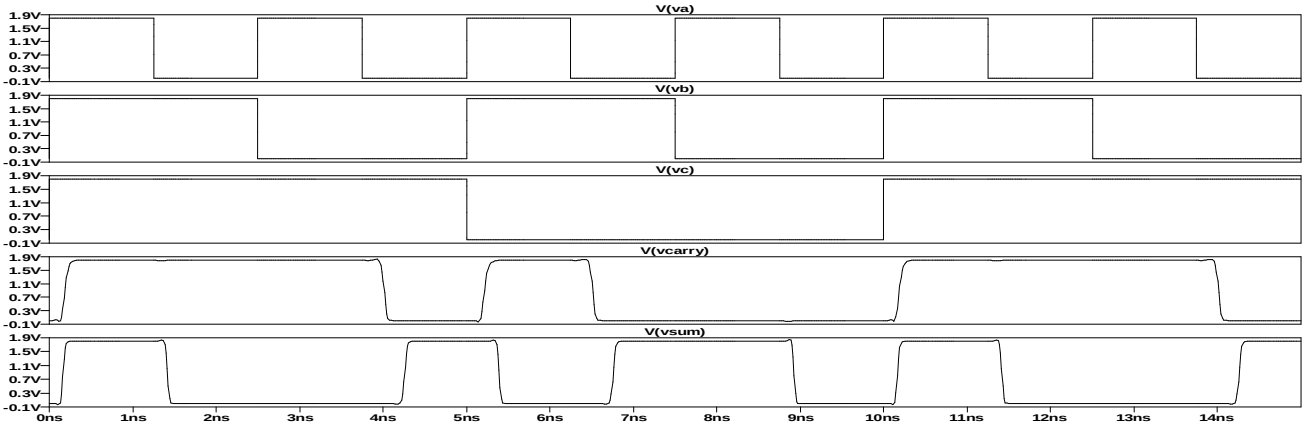


Fig. 14 Perceptron full adder layout simulation waveforms

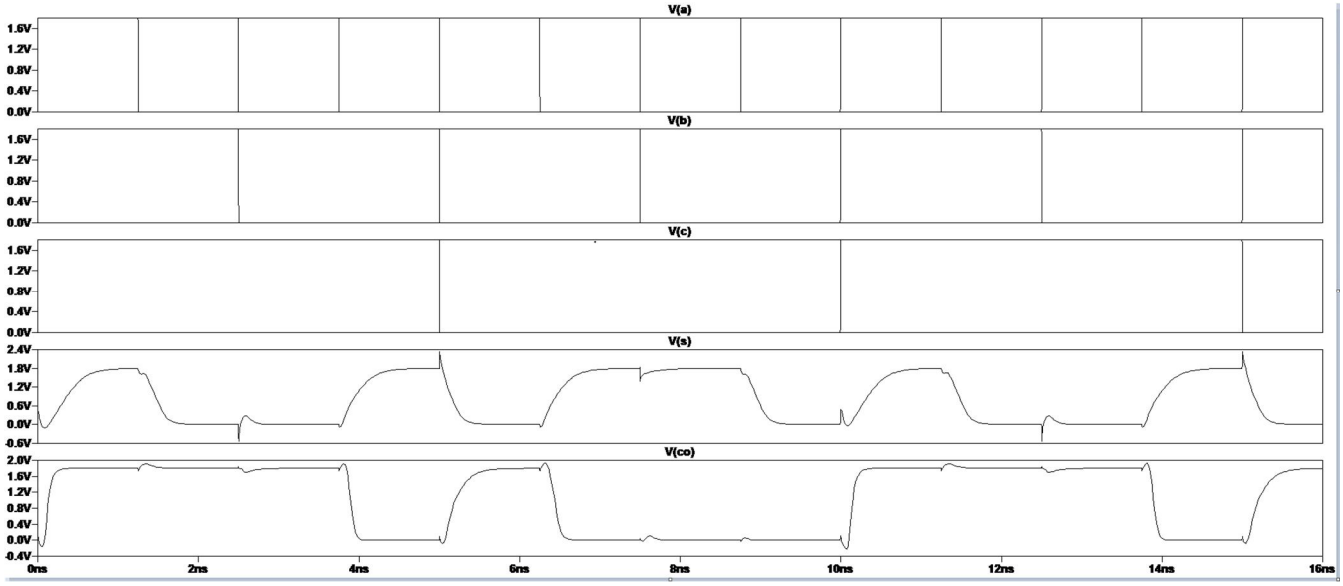


Fig. 15 CMOS full adder layout simulation waveforms

Table V shows the average power and propagation delays measured using simulation. These values are compared to the CMOS full adder. The delays vary based on the inputs to the adder. From Table V, a reasonable value for the bias voltage Vb3 is found to be around 0.85V. The CMOS adder exhibited the lowest delay and power. The dynamic power for the perceptron adder is much larger compared to the CMOS adder because of the simultaneous conduction of both NMOS and PMOS transistors for certain input combinations.

TABLE V. FULL ADDER TIMIMG AND POWER MEASUREMENT

		Neuron Adder Worst delay (ns)						CMOS Full Adder Delay (ns)
Metric		Vb3						
		0.6V	0.7V	0.8V	0.9V	1.0V	1.1V	
Sum	tplh	0.389	0.399	0.418	0.448	0.51	0.78	0.33
	tphl	0.983	0.533	0.456	0.417	0.392	0.373	0.22
Carry	tplh	0.215						0.18
	tphl	0.264						0.2
Average Power	mW	1.06	1.09	1.13	1.17	1.22	1.27	0.077
Static Power	nW	0.44						0.15
Transistor Count		23						36

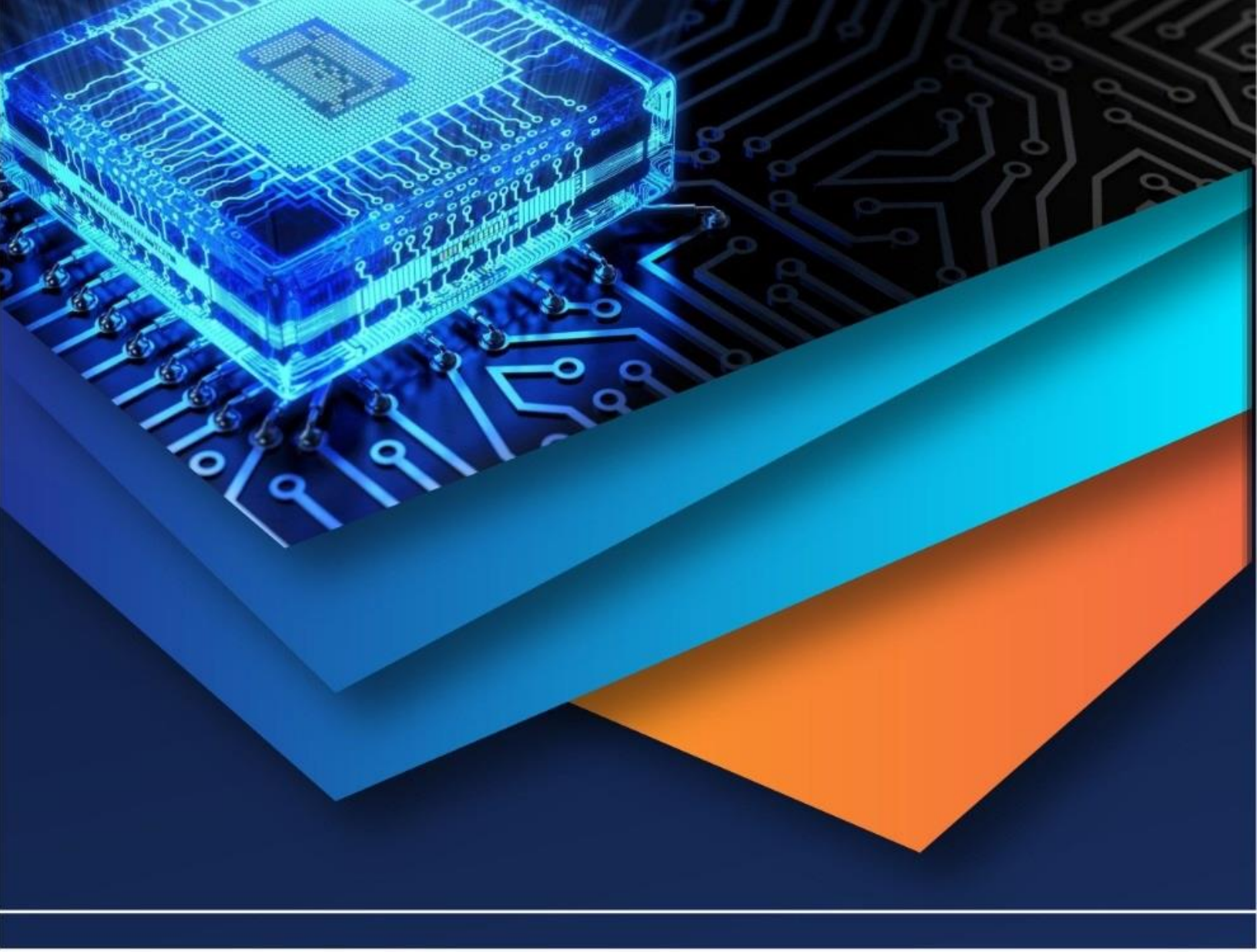
V. CONCLUSIONS

Neural network based design of logic gates and full adder are presented in this paper. All designs are based on tsmc180nm CMOS process at a supply voltage of 1.8V. Electric VLSI tools are used for making the CMOS layouts. LTspice simulations were performed on the NOT gate, two-input NAND gate and the full adder. The bias voltages are to be selected properly for the correct functioning of these circuits. For the full adder, the bias voltage for the carry side NMOS transistor is fixed at 1.8V. For the sum side NMOS transistor it is variable between 0.6V to 1.1V.

Propagation delays and power dissipation varies with the bias voltage. A bias voltage of 0.85V on the sum side transistor keeps the propagation delays less than 0.5ns. The average power consumption is little over 1mW. This is much higher compared to the normal CMOS adder due to the simultaneous conduction of both NMOS and PMOS transistors under certain input conditions. The number of transistors used in the neuron adder is only 23 as compared to 36 for the CMOS adder.

REFERENCES

[1] W.S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics. vol. 5, pp. 115-133, 1943.
[2] V.K. Singh, "Proposing Solution to XOR problem using Minimum Configuration MLP", Procedia Computer Science, Elsevier, pp. 263-270, 2016. vol. 85.
[3] D.P. Sharma, "Neural Network Simulation of Digital Circuits", International Journal of Computer Applications (0975 – 8887), vol. 79, no. 6, pp. 7-13, 2013.
[4] C.S. Rai and A.P. Singh, "A Review of Implementation Techniques for Artificial Neural Networks", University School of Information Technology, GGS Indraprastha University, Delhi, 2006.
[5] U. A. Belorkar and A. E. Bhande, "VLSI Implementation of Back Propagated Neural Network for Signal Processing", International Journal of Electrical and Electronics Engineering Research, vol. 4, issue 2, pp. 131-140, 2014.
[6] J. Han, Z. Li, W. Zheng, Y. Zhang, "Hardware Implementation of Spiking Neural Networks on FPGA," Tsinghua Science and Technology, vol. 25, no. 4, pp. 479-486, Aug. 2020.
[7] S. Moukhlis, E.R Abdessamad and A. Hamdoun, "FPGA Implementation of Artificial Neural Networks", International Journal of Computer Science Issues, vol. 11, issue 2, no 1, pp. 237-239, 2014.
[8] M. Subadra, K.P Lakshmi, J. Sundar and K. MathiVathani, "Design and Implementation of Multilayer Perceptron with On-Chip Learning in Virtex-E", in Proc. 2nd AASRI Conference on Computational Intelligence and Bioinformatics, AASRI Procedia 6, 2014, pp. 82 – 88.
[9] [Online] https://en.wikipedia.org/wiki/Biological_Neuron_model.
[10] N. Farha, A. L. Paul, N. Kousar, S. Devika and R. Divakaran, "Design and Implementation of Logic Gates and Adder Circuits on FPGA Using ANN", International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 4 (5), pp. 623-629, 2016.
[11] C. Enslen, "Transistor Neural Network – NAND Gate and Full Adder using Analog Neural Networks", State University of New York, New Paltz, NY, CE Tech. Rep., May 2026.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY