Neuromorphic Computing(Neuron based memory computing)

Thesis submitted in partial fulfillment for the award of the

degree of

B.Tech in Electronics & Communication Engineering

Submitted By

Athish SA(B21EC040)

Under the guidance of

Dr. Anup Dandapat (Professor, ECE)

Department of Electronics and Communication

National Institute of Technology Meghalaya

Meghalaya, India - 793003

07 May 2024

ABSTRACT

The Artificial Neural Network (ANN) is a widely used computing mechanism that tries to replicate the operation of the human brain. Nowadays ANNs are used in various fields like image processing,

arti ficial intelligence, medical diagnosis, pattern recognition, and so on. It attempts to mimic the human brain. It tries to replicate the behaviour of the human brain but the main constrain is ‘the memory’ concept. The human brain has memory. It can store a piece of information in memory and retrieve that information when required. On the other hand, ANN can not store data in them. It can neither create a mem ory nor store data in itself. It does not have its own memory. The project work undertaken attempts to create a memory using ANN. It also endeavours to search for created memory. Initially, a digital neuron is to be created by using Multiplier, Summation block, and Activation block.Consequently, this novel research proposes computer-based mem ory with the help of the concept of human memory.

LIST OF FIGURE

Outline

CHAPTER 1

INTRODUCTION

1.1 Introduction

Neuromorphic computing is an approach to artificial intelligence (AI) and computing that is inspired by the structure and functioning of the human brain. The term ”neuromorphic” comes from ”neuron” (nerve cell) and ”morphing” (changing form), reflecting the idea of cre ating computing systems that mimic the neurobiological architecture of the nervous system of the human brain. Neuromorphic computing, inspired by the human brain’s architecture, aims to emulate cognitive processes using hardware that mimics neurons and synapses. One key concept is neuron-based memory computing, where memory and computation are intertwined within individual neurons. Unlike traditional von Neumann architecture, where memory and processing are separate, neuromorphic systems leverage the inherent parallelism and energy efficiency of neural networks to perform tasks such as pattern recog nition and learning. Neuron-based memory computing stores information in the strength of synaptic connections, enabling efficient and fault-tolerant

processing. This approach holds promise for applications in artificial intelligence, robotics, and cognitive computing, offering solutions for tasks that require real-time processing and adaptation to dynamic environments. Through the integration of memory and computation at the neuron level, neuromorphic computing represents a novel paradigm for efficient and brain-like information processing.

.

1.2 Motivation

- The motivation behind exploring neuron-based memory computing within the realm of neuromorphic computing stems from the limitations of traditional computing ar chitectures in emulating the brain's remarkable efficiency and adaptability. While conventional von Neumann architectures excel in sequential processing and precise numerical calculations, they struggle with tasks that demand parallelism, real-time adaptation, and energy efficiency.

- The brain, on the other hand, processes information in a massively parallel manner, with memory and computation seamlessly integrated within neurons and synapses.This architecture enables cognitive capabilities such as pattern recognition, learning, and robustness to noisy and uncertain data. By mimicking the brain's structure and prin ciples, neuromorphic computing aims to overcome the bottlenecks of conventional computing systems.

- Neuron-based memory computing specifically addresses the need for efficient storage and processing of information within the same computational units, thereby enabling low-power, fault-tolerant, and adaptive computing systems. This approach is moti vated by the desire to develop intelligent systems that can learn from experience, rec ognize complex patterns, and operate efficiently in real-world environments, ultimately paving the way for advancements in artificial intelligence, robotics, and autonomous systems.

1.3 Objective

1. Develop hardware architectures: Design and implement neuromorphic computing

hardware that mimics the structure and function of neurons and synapses. This involves engineering physical components capable of emulating neural processing, in cluding mechanisms for neuron-based memory storage and computation.

2. Design algorithms: Develop algorithms and software frameworks optimized for neuron based memory computing architectures. These algorithms should leverage the unique

properties of neuromorphic hardware to enable tasks such as pattern recognition, learning, and adaptation. The focus is on creating efficient and scalable algorithms that harness the parallelism and energy efficiency of neuromorphic systems to achieve cognitive capabilities akin to the human brain.

CHAPTER 2

ARTIFICIAL NEURAL NETWORK

2.1 Artificial Neural Network

Artificial Neural Network (ANN) tries to form networks to solve computational problems of the kind that biology does effortlessly. The human brain is the best computing system. ANN tries to mimic the behaviour of the human brain. It tries to replicate the information analyzing and processing ability of the human brain. Like the human brain, ANNs are made up of simple processing elements interconnected to form a network. These simple computing elements represent a simplified model of the neuron. ANN can be a huge parallel architecture or a small network as per the requirements. The human brain is a massively parallel, fault-tolerant, adaptive network. It has the ability to learn, to memorize. It can generalize things with the least possible energy consumption. Inspired by the human brain, ANNs can be massively parallel. They also can learn things and to a certain extent fault-tolerant too . ANNs are widely used in pattern recognition, optimization, prediction, function approximation . It is also used in satellite communication , GPS , human speech identification , earthquake prediction, conversion of black and white images to color images , robotics due to their massive parallel, adaptive, fault-tolerant

nature . It's also utilized in wireless communication, medical applications, and satellite communication to improve logic circuits. . It has been used in a range of medical studies, such as the automatic detection of spike-wave complexes in EEG signals, the detection of cancer tumors in X-ray images, etc. In , a non-invasive blood glucose sensing system was presented using near-infra-red (NIR) spectroscopy where ANN was used to process the signals from NIR optodes. In this system, the inverse delayed model of a neuron was used. Apart from these applications, ANN had also been employed in the identification of dynamic patterns in signatures, recognition of vehicle number plates, data recognition in machine learning, machine condition monitoring, etc. It has gained astonishing success in engineering fields also. Such as intelligence control, robotics, information processing, computer vision, pattern identifying, etc. It's also booming in sectors like economics and management .

CHAPTER 3

Formation of Neural Networks

3.1 ANN and BNN

The neuron serves as the fundamental unit of the brain, facilitating communication through neural pathways. It comprises four main parts: dendrites receive signals, which are then transmitted as electrical impulses to the cell body. Here, signals are summed and pro cessed, with the axon subsequently carrying them to synapses. Synapses act as functional connections between neurons. While the human brain contains billions of neurons, forming extensive neural networks, it is organized into functional areas to manage complexity. Ar tificial Neural Networks (ANNs) emulate this structure, employing artificial neurons with inputs, weights, biases, activation functions, and outputs. These artificial neurons process inputs through weighted sums and activation functions to generate outputs, mimicking the functionality of biological neurons within the brain's communication system.

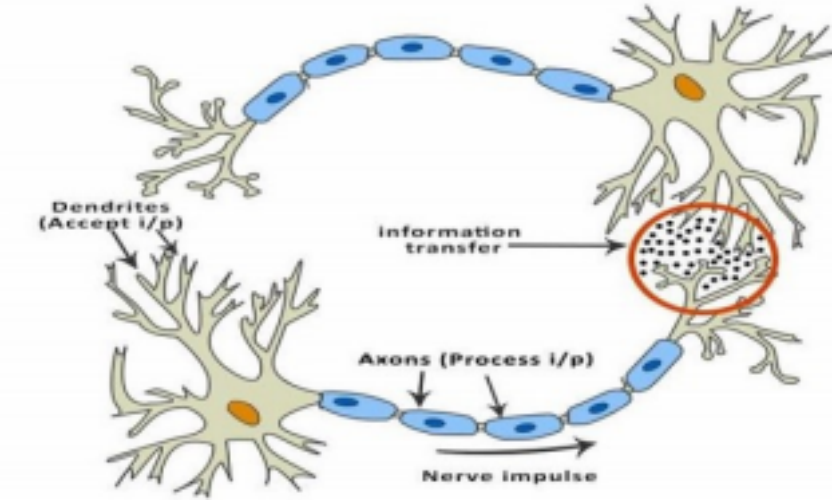

Figure 1: BIOLOGICAL NEURAL NETWORK (BNN)

The output equation of an artificial neuron is as follows: $Y = f\left(\sum_{(i=1)}^{N}(x_i w_i + b)\right)$ In artificial neurons, 'Y' represents the output, 'n' signifies the number of inputs, 'x' and 'w' denote input and weight respectively, while 'b' stands for bias. The activation function 'f' governs the output generation. Weights can be positive (excitatory input) or

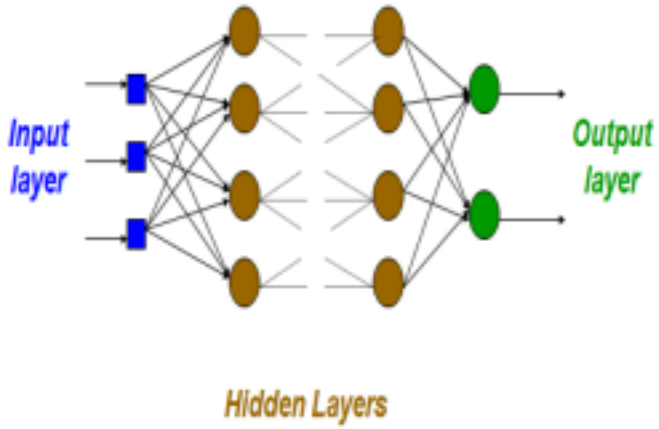

Figure 2: Basic structure of ANN

negative (inhibitory input), akin to synaptic behavior in biological neurons. The wiring and interconnections in artificial neurons parallel the axon and dendrites in biological neurons. Activation functions in artificial neurons mimic the thresholding process in the cell body of biological neurons, where inputs are processed and transformed into output signals. This analogy underscores the foundational similarities between artificial and biological neural networks.

Artificial Neural Networks (ANNs) are constructed by linking artificial neurons, forming weighted graphs where the neurons represent nodes. ANNs are categorized based on their connections into Feed-forward Neural Networks and Recurrent Neural Networks. Feed forward networks lack feedback paths, rendering them static, while Recurrent networks incorporate feedback loops, making them dynamic. ANNs typically comprise three layers: input, hidden, and output. The input layer receives inputs, while the output layer generates outputs. Hidden layers, situated between the input and output layers, process and transform the input data through interconnected neurons. This layered architecture enables ANNs to perform complex tasks such as pattern recognition and classification, with each layer contributing to the network's ability to learn and make predictions.

CHAPTER 4

DESIGN FLOW OF ANN

4.1 Components of artificial neural network

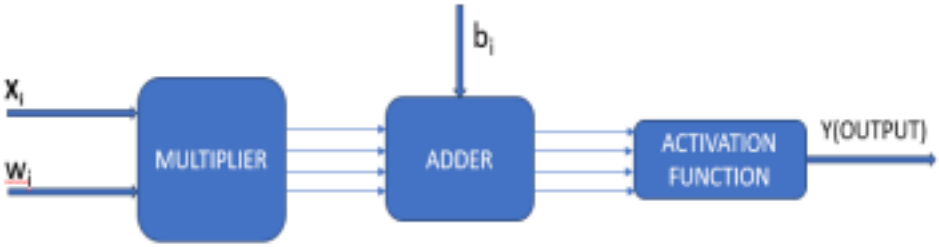

Figure 3: Basic structure of ANN

Weights and bias are both learnable parameters inside the network. A teachable neural network will randomize both the weight and bias values before learning initially begins. Activation function is just a thing function that you use to get the output of node. It is also known as Transfer Function. It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function). This can be achieved by usage of multipliers ,adders and the design of activation function which are designed using the simulation software Cadence

Virtuoso.

1. Weights Initialization: Before training begins, weights are often initialized randomly to break symmetry and prevent neurons from learning the same features. Common initialization techniques include random sampling from a normal distribution or uni form distribution.

2. Bias Initialization: Similar to weights, biases are initialized randomly to introduce flexibility and enable the model to learn optimal representations. Biases provide each neuron with its own activation threshold, affecting the overall behavior of the network.

3. Activation Function Properties: Different activation functions have distinct properties that influence network behavior. For example, the ReLU activation function is com-

putationally efficient and mitigates the vanishing gradient problem, while the sigmoid function provides outputs bounded between 0 and 1, suitable for binary classification tasks.

4. Activation Function Selection: Choosing the appropriate activation function depends on the task and the characteristics of the data. For instance, the tanh activation func tion is commonly used in hidden layers to capture non-linearities, while the softmax function is utilized in the output layer for multi-class classification.

5. Activation Function Derivatives: During backpropagation, the derivatives of activa tion functions are crucial for updating weights and biases. Derivatives dictate the rate at which weights and biases are adjusted based on the error gradient, influencing the network's learning dynamics.

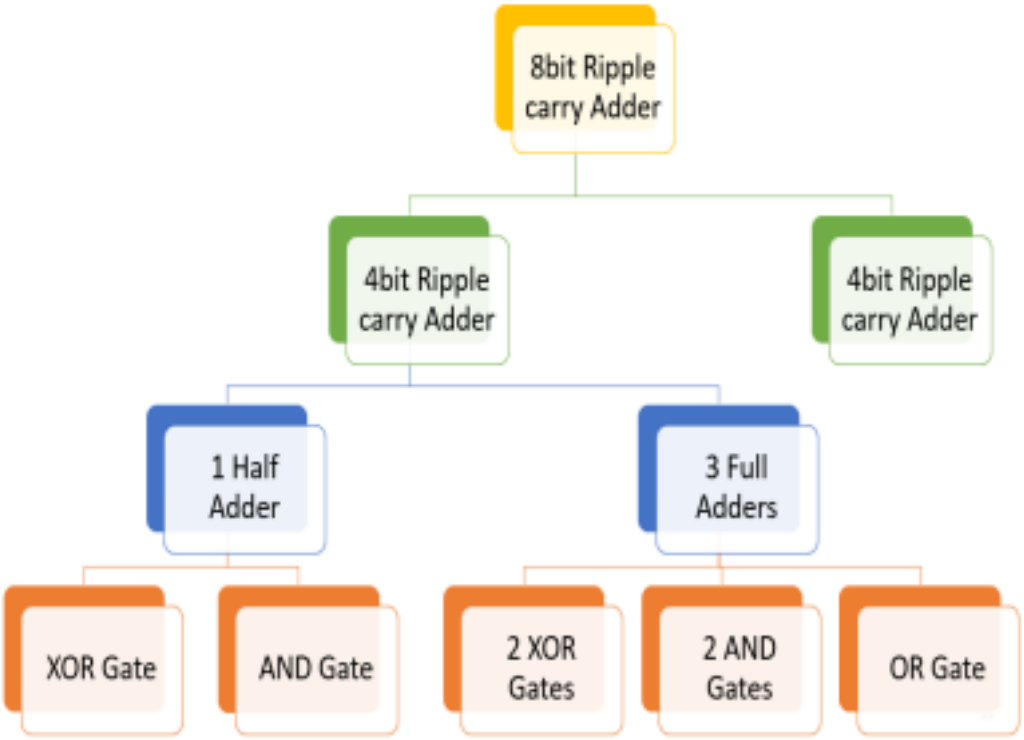

Figure 4: Design flow of 8bit Ripple Carry Adder.

CHAPTER 5

Results and Discussion

5.1 Simulation Results

We have designed 8-bit ripple carry adder which is an important component in the im plementation of the artificial neural network. This design of 8-bit ripple carry adder was performed using the simulation software Cadence Virtuoso. The 8-bit ripple carry adder consists of one half adder and seven full adders.

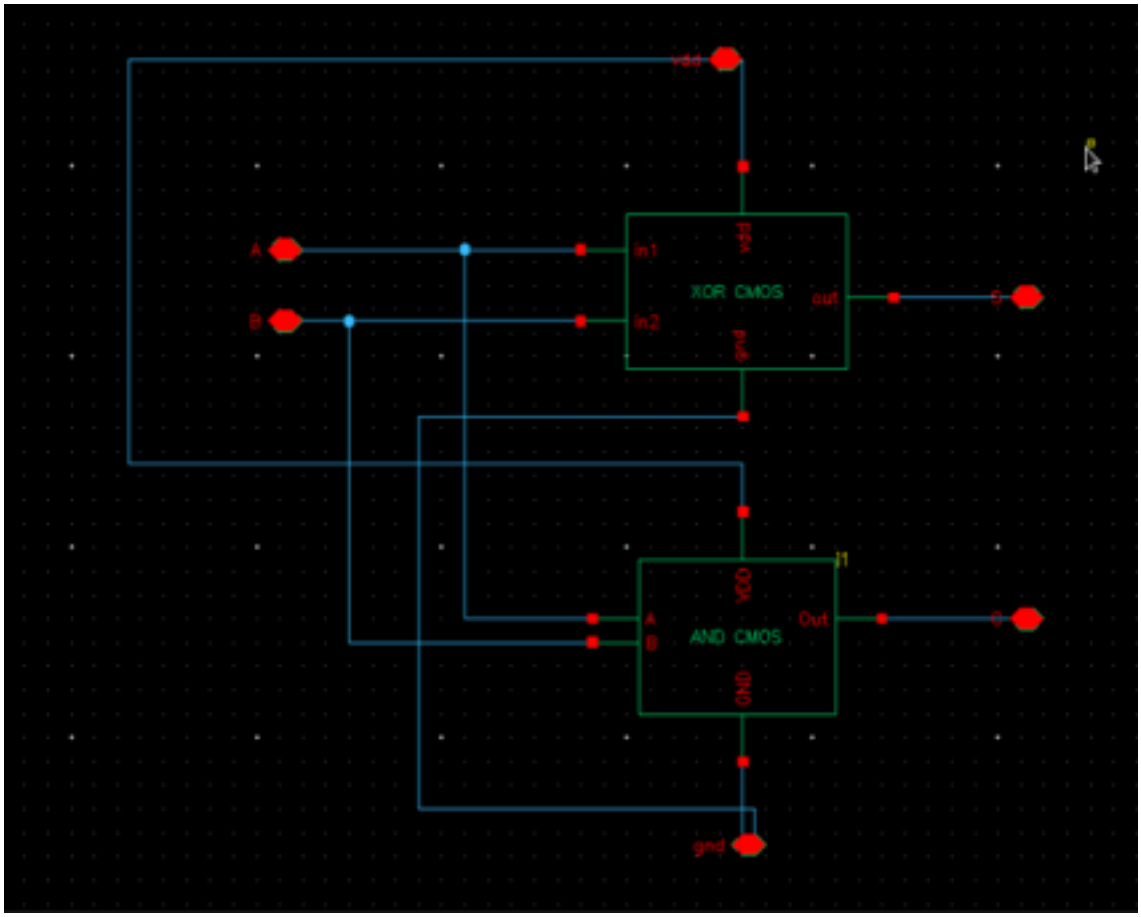

Figure 5: Half Adder Schematic Circuit.

The Figure 5 shows the Cadence design of Half adder schematic circuit which consists of ’XOR CMOS’ gate and ’AND CMOS’ gate, which were designed using nmos and pmos transistors using the gpdk045nm technology.

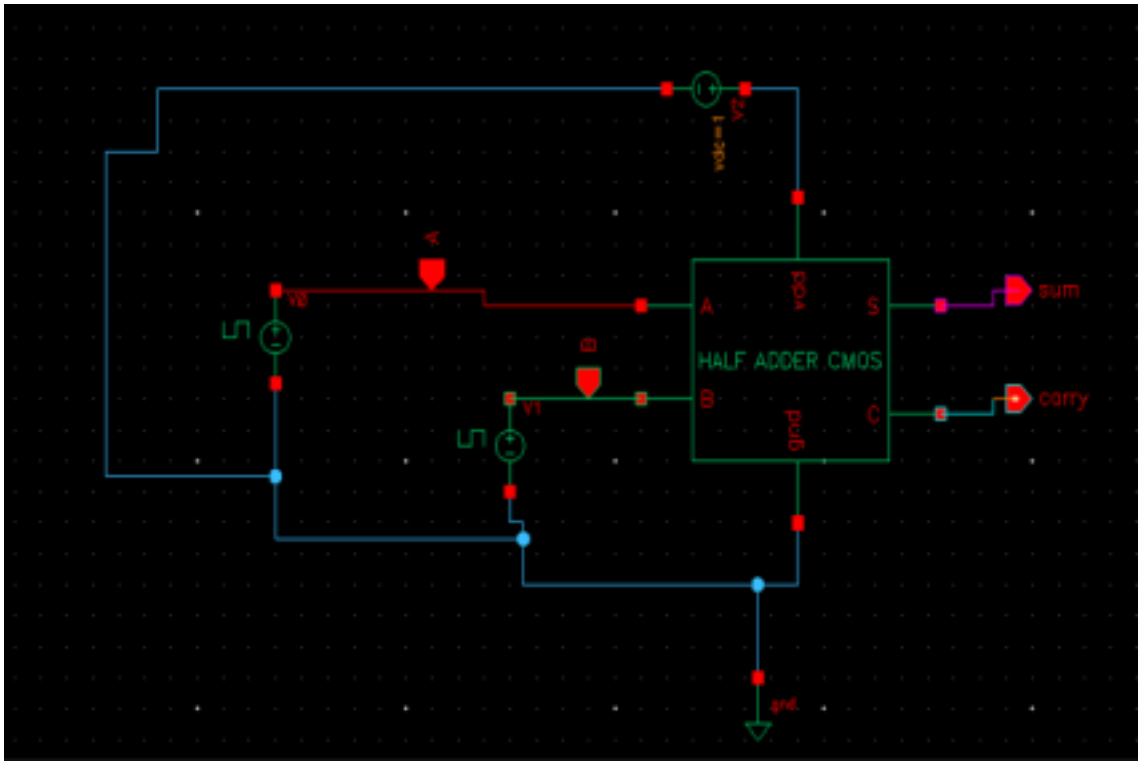

Figure 6: Half Adder Symbolic Circuit.

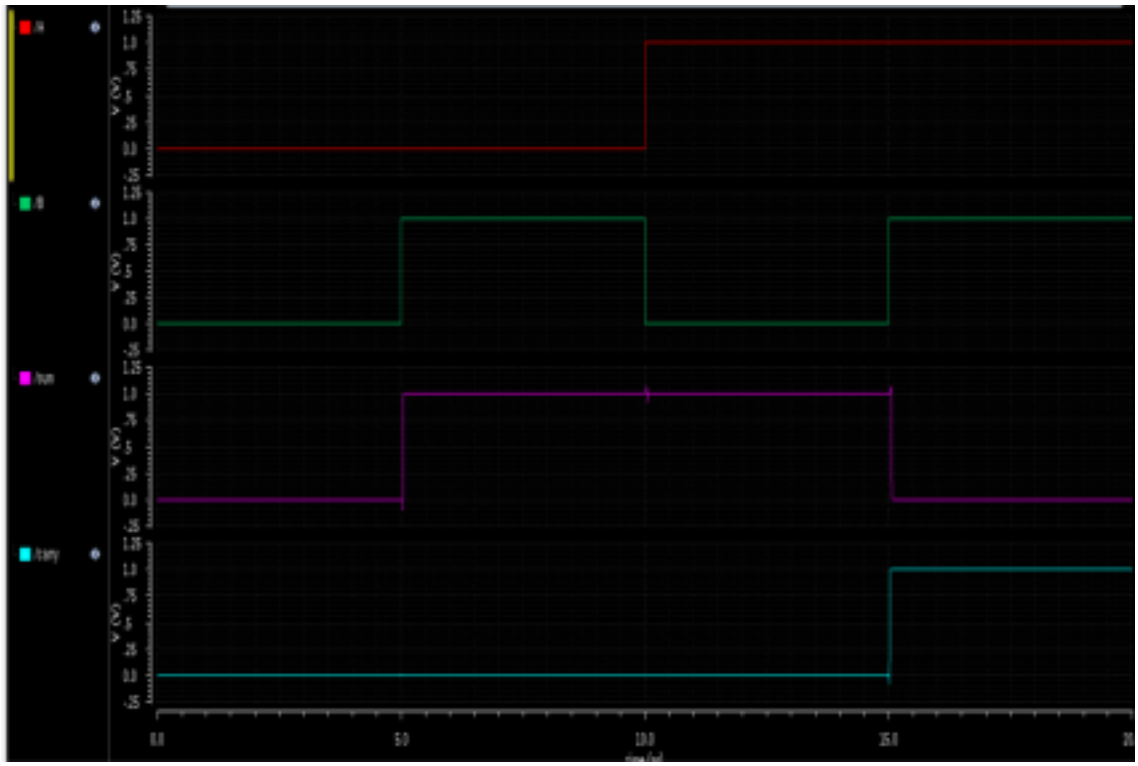

Figure 7: Half Adder Output.

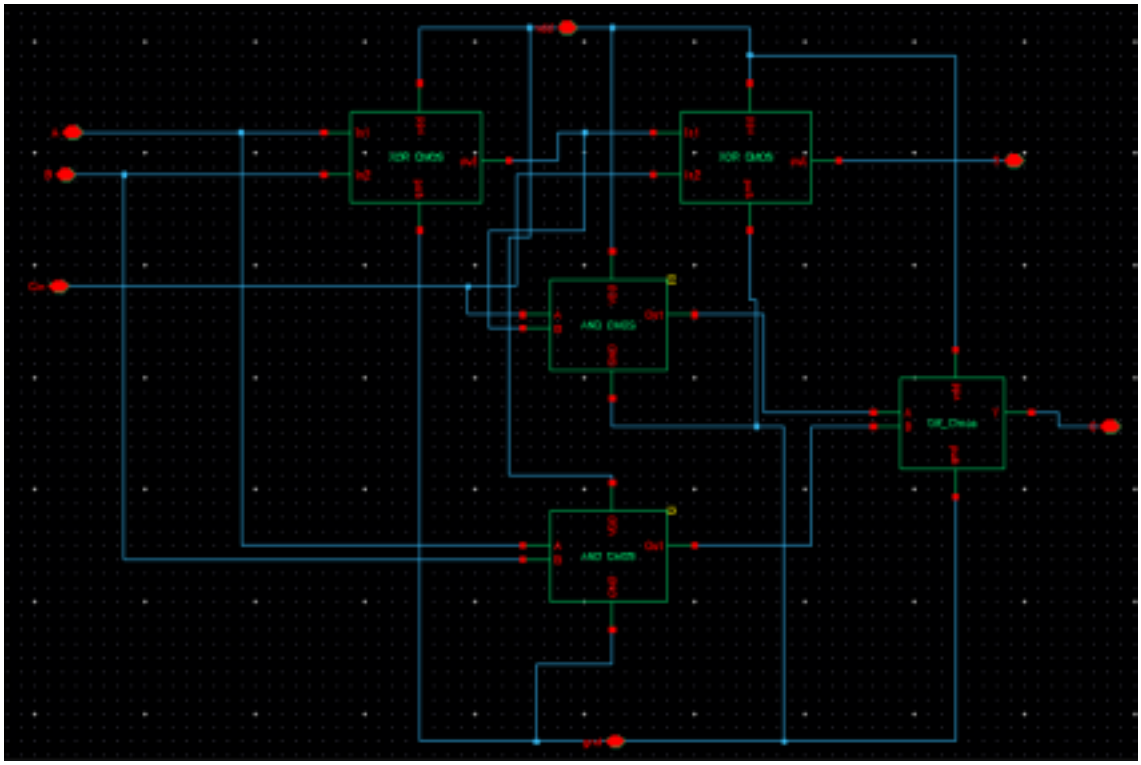

Figure 8: Full Adder Schematic Circuit.

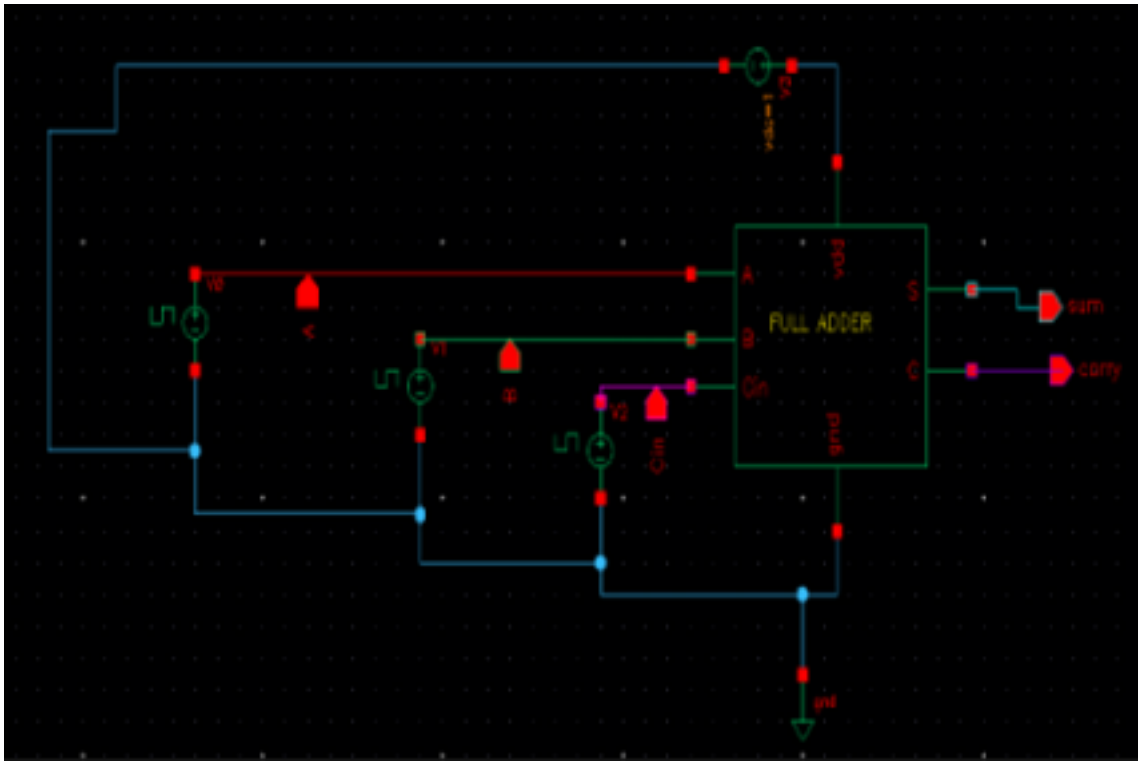

Figure 9: Full Adder Symbolic Circuit.

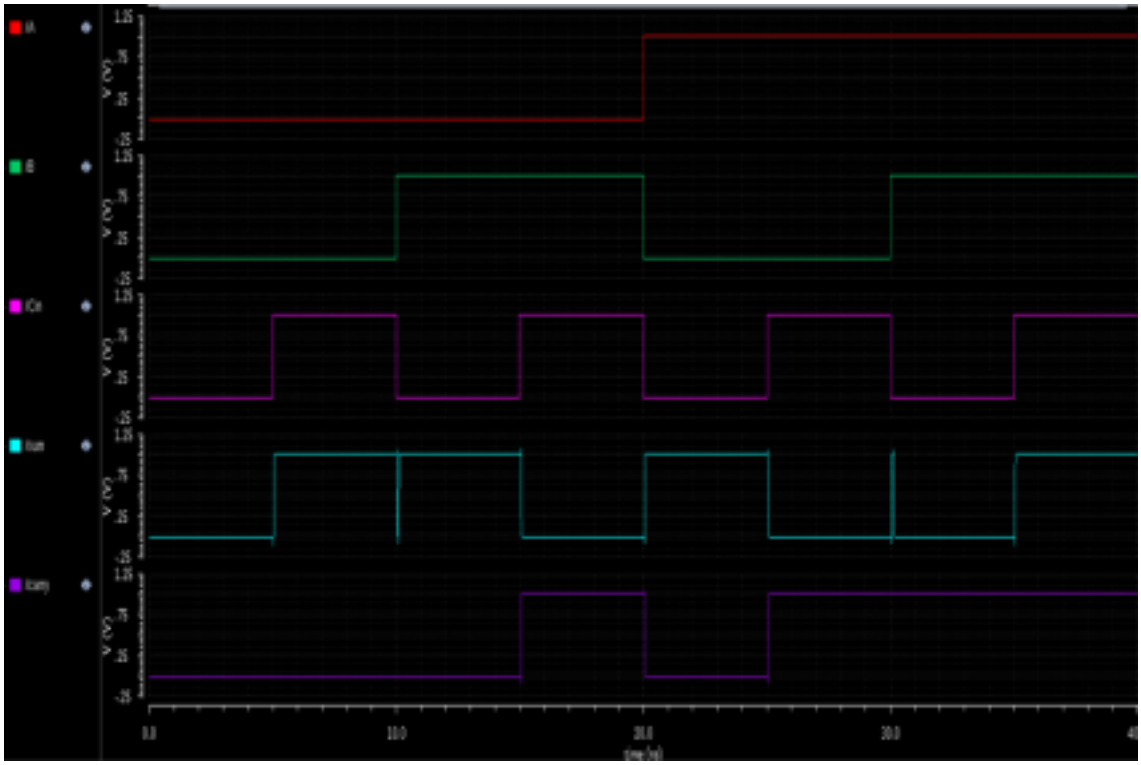

Figure 10: Full Adder Output.

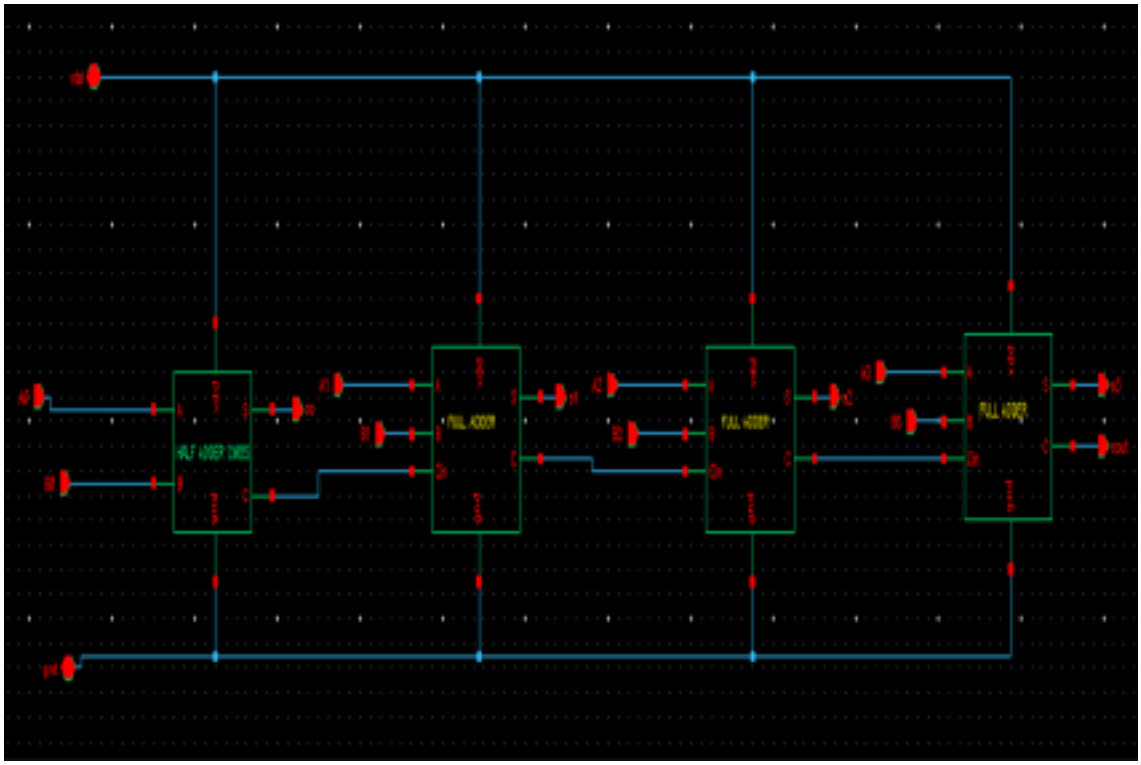

Figure 11: 4-Bit Ripple Carry Adder Schematic Circuit.

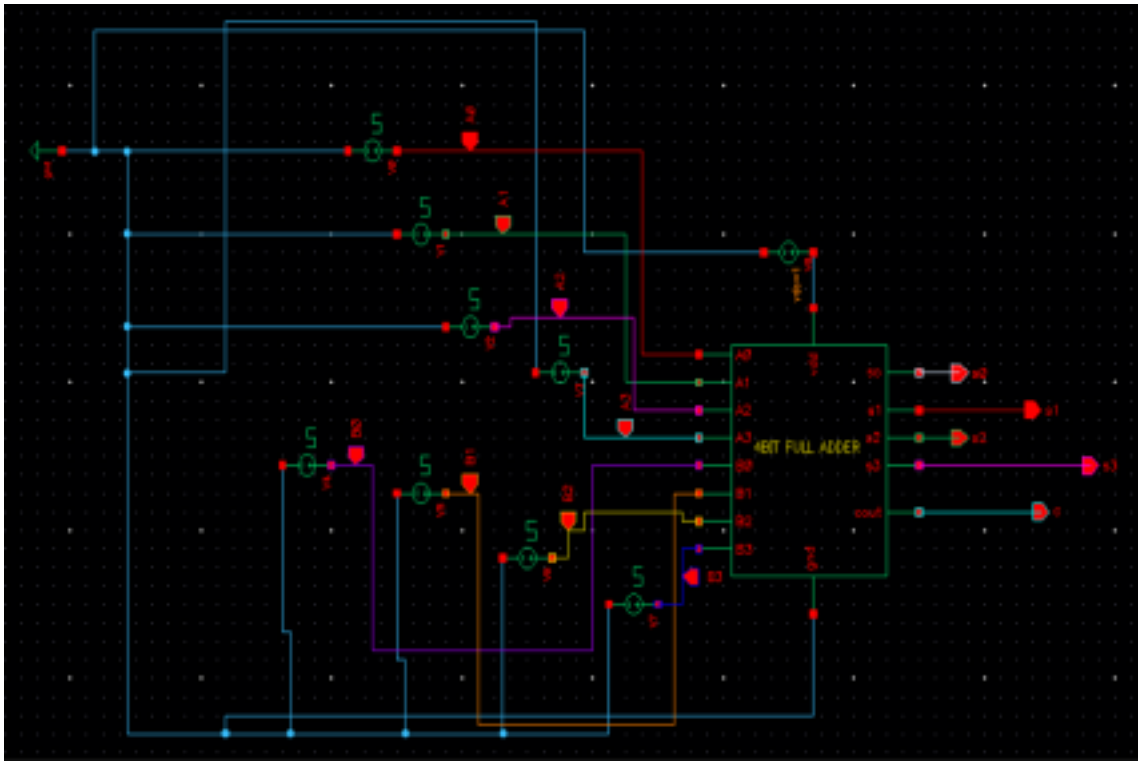

Figure 12: 4-Bit Ripple Carry Adder Symbolic Circuit.

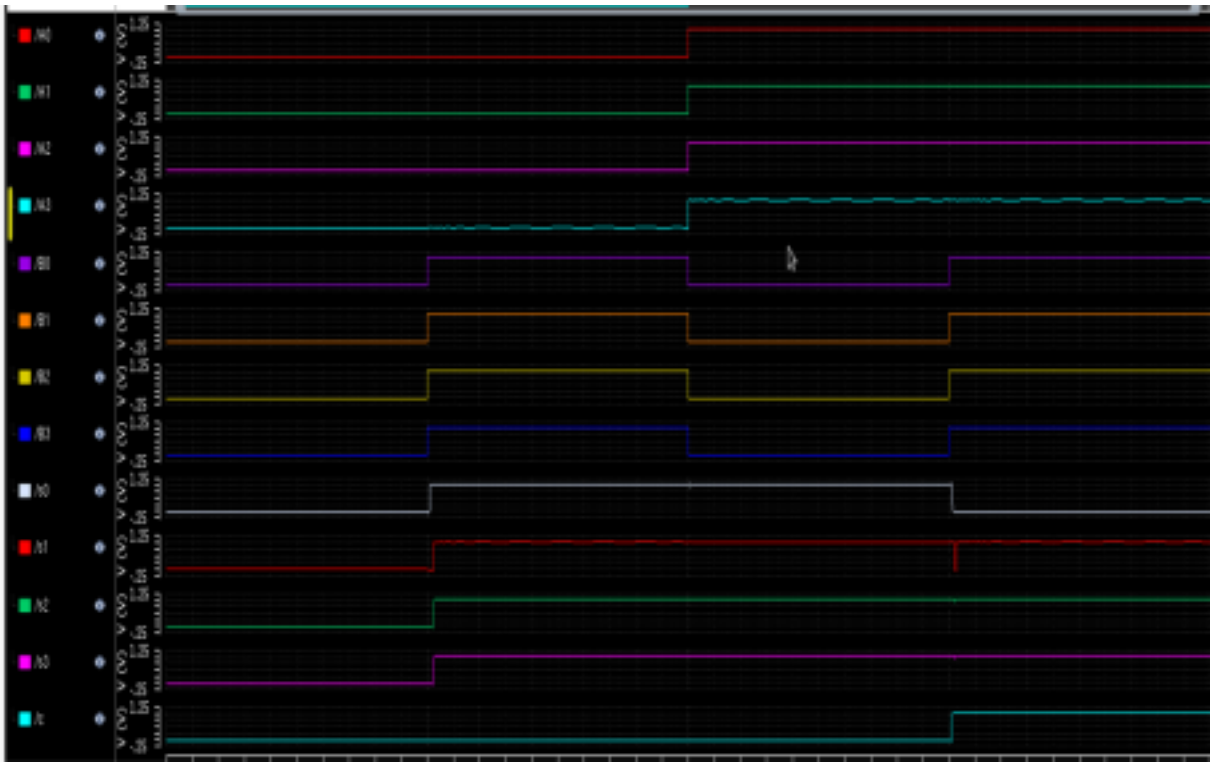

Figure 13: 4-Bit Ripple Carry Adder Output.

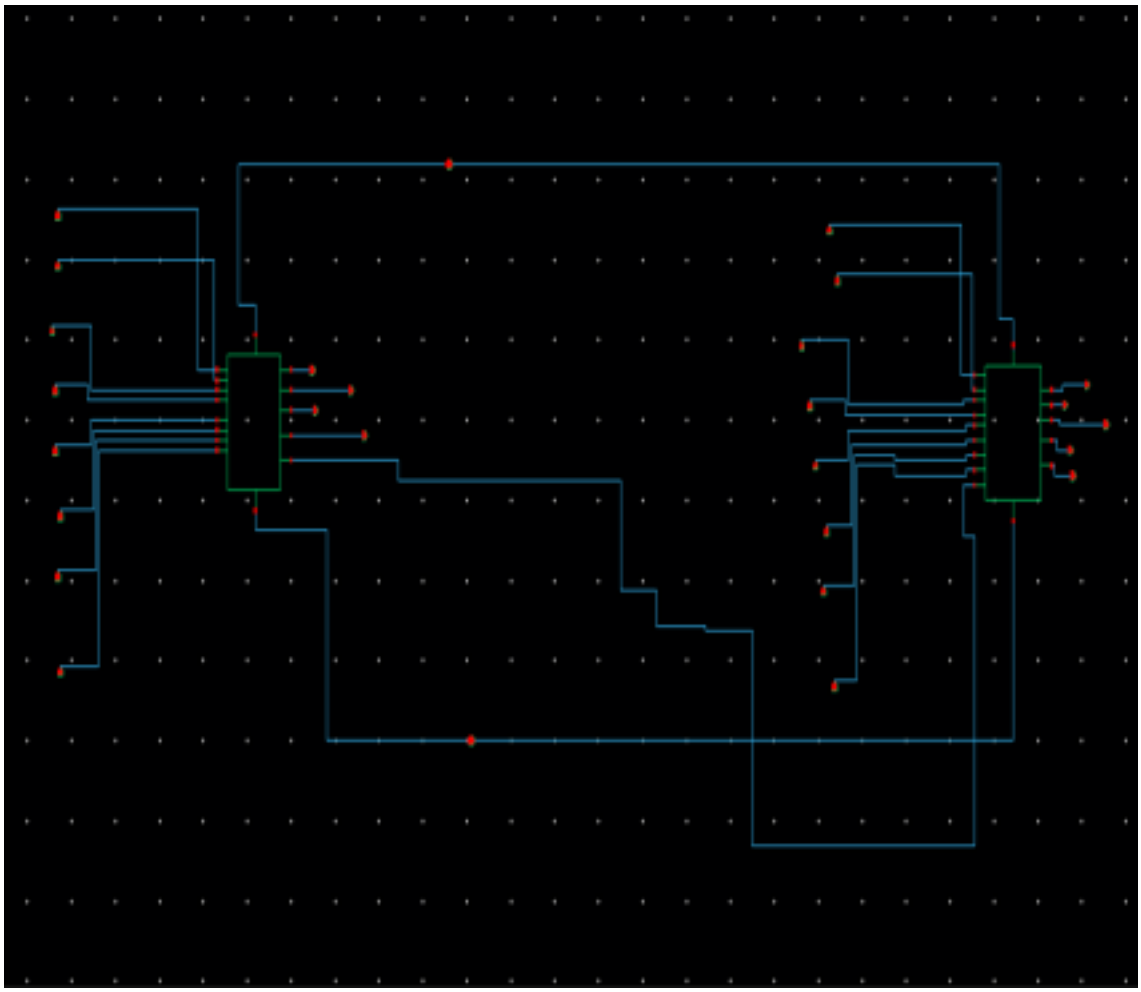

Figure 14: 8-Bit Ripple Carry Adder Schematic Circuit.

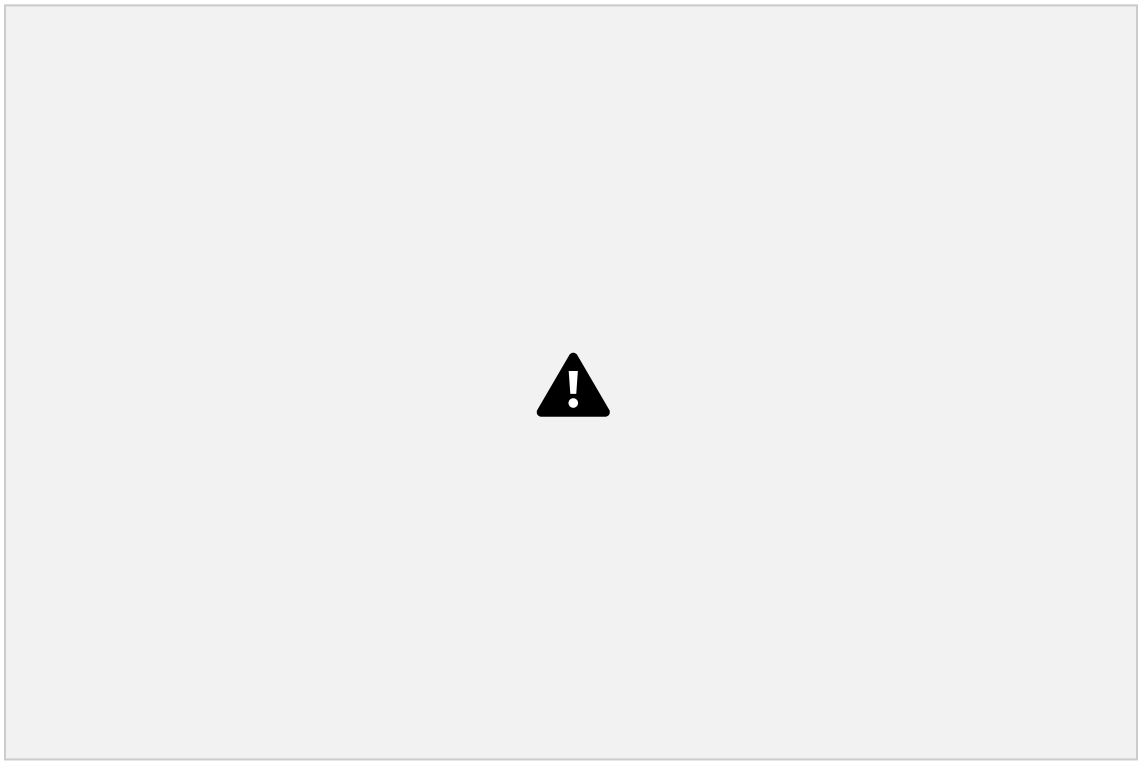

Figure 15: 8-Bit Ripple Carry Adder Symbolic Circuit.

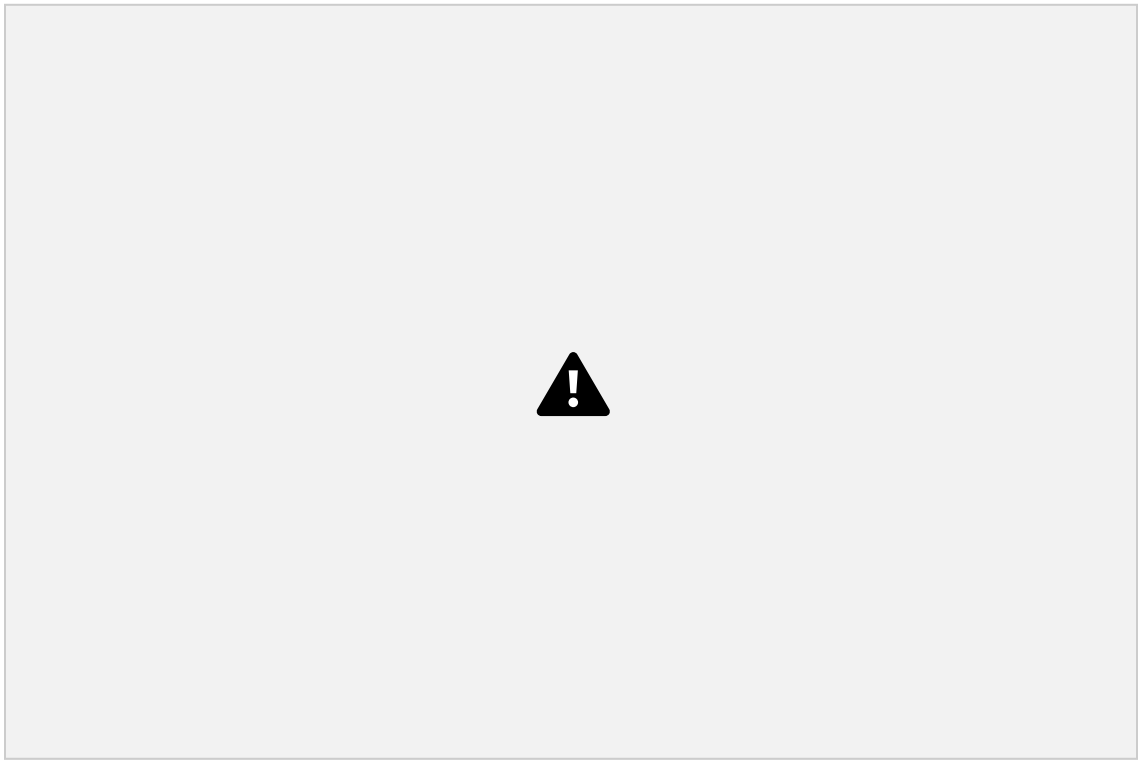

Figure 16: 8-Bit Ripple Carry Adder Output.

The Figure 6 shows the Cadence design of Half adder symbolic circuit which consists

of two inputs A,B and two outputs Sum and Carry.

The Figure 7 shows the Cadence output for Half adder simulated in Cadence . The Figure 8 shows the Cadence design of Full adder schematic circuit which consists of ’2-XOR CMOS’ gate, ’1-OR CMOS’ gate and ’2-AND CMOS’ gate, which were designed using nmos and pmos transistors using the gpdk045nm technology.

The Figure 9 shows the Cadence design of Full adder symbolic circuit which consists of three inputs A,B and C, and two outputs Sum and Carry.

The Figure 10 shows the Cadence output for Full adder simulated in Cadence . The Figure 11 shows the Cadence design of 4-bit ripple carry adder schematic circuit which consists of ’1-HALF ADDER’ and ’3-FULL ADDERS’, which were designed using nmos and pmos transistors using the gpdk045nm technology.

The Figure 12 shows the Cadence design of 4-bit ripple carry adder symbolic circuit which consists of 8-inputs and 5-outputs .

The Figure 13 shows the Cadence output for 4-bit ripple carry adder simulated in Cadence .

The Figure 14 shows the Cadence design of 8-bit ripple carry adder schematic circuit which consists of ’1-HALF ADDER’ and ’7-FULL ADDERS’, which were designed using the gpdk045nm technology.

The Figure 15 shows the Cadence design of 8-bit ripple carry adder symbolic circuit which consists of 16-inputs and 9-outputs .

The Figure 16 shows the Cadence output for 8-bit ripple carry adder simulated in Cadence .

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

In conclusion, the integration of CMOS-based memory cells into artificial neural networks represents a significant step towards bridging the gap between biological brains and compu tational systems. By leveraging the principles of gpdk45nm technology, we have

successfully designed and analyzed the implementation procedures for artificial neuron structures. This endeavor not only enhances our understanding of neural computation but also paves the way for developing more efficient and biologically inspired computational models. The schematics of gates and adders produced serve as valuable assets for future endeavors in neural network design, facilitating the realization of increasingly sophisticated and brain-like computational systems.

6.2 Future Work

We would design multipliers and use it in the design of the artificial neuron. We would design the most suitable Activation function which is best suitable for the implementa tion of artificial neural networks. We would implement the proposed neuron design using CMOS technology. We would explore the reduction in power consumption with the usage of ANN technology. We would study about SNN's, CNN's and RNN's for providing further advancements in the designed artificial neuron technology .

References

[1] Y. L. Sung-Mo Kang, "Cmos digital integrated circuits."

[2] S. K. J.S.Srinivas Raju and L. Sneha, "Realization of logic gates using mcculloch-pitts neuron mode in ijeet," vol. 45, no. 2, pp. 135–158, 2017.

[3] B. B. J.Vijaychandra, B.Sesha Sai and P.Jagannadh, "A comprehensive review on mcculloch-pitts neuron model," *International Journal of Innovative*.

[4] J. Hopfield, "Artificial neural networks," *IEEE Circuits and Devices Magazine*, vol. 4, no. 5, pp. 3–10, 1988.

[5] E. Brunvand, "Digital vlsi chip design with cadence and synopsys cad tools: Vlsi cad lab manual ssp_4."