

# **PERFORMANCE ANALYSIS OF AN EFFICIENT TERNARY MULTIPLIER FOR VLSI APPLICATION**

Submitted in Partial fulfillment of the Requirements for the Award of  
Bachelor of Engineering degree in Electronics and Communication Engineering

by

SHALINI P (38130207)

VARSHA P (38130231)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING  
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**MAY 2022**



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI - 600 119**



[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

---

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **SHALINI P (38130207)**, and **VARSHA P (38130231)** who carried out the Project entitled “**PERFORMANCE ANALYSIS OF AN EFFICIENT TERNARY MULTIPLIER FOR VLSI APPLICATION**” under the guidance of **MR.K.T ILAYARAJAA.,M.E., (Ph.D)** during the period **DECEMBER 2021-MAY 2022.**

Internal Guide

Mr. K.T ILAYARAJAA, M.E., (Ph.D)

Head of the Department

Dr. T RAVI, M.E., Ph.D

---

**Submitted for Viva voce Examination held on** \_\_\_\_\_

**Internal Examiner**

**External Examiner**

## DECLARATION

We, SHALINI P (Reg. No: 38130207) and VARSHA P (Reg. No: 38130231) here by declare that the Project Report entitled **PERFORMANCE ANALYSIS OF AN EFFICIENT TERNARY MULTIPLIER FOR VLSI APPLICATION** is done by us under the guidance of Mr. K.T ILAYARAJAA, M.E., (Ph.D) and at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

1.

PLACE:

2.

**SIGNATURE OF THE CANDIDATES**

## **ACKNOWLEDGEMENT**

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project for completing it successfully. We are grateful to them.

We convey our sincere thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D. Dean, School of ELECTRICAL AND ELECTRONICS ENGINEERING** and **Dr. T RAVI, M.E., Ph.D. Head of Department, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Mr. K.T ILAYARAJAA, ME., (Ph.D)** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our sincere thanks to all teaching and Non-teaching staff members of the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Carbon Nanotube Field Effect Transistors (CNFETs) are considered to be an ideal choice for implementation of Multi-valued logic circuits, as by using CNFETs multiple thresholds can be obtained by altering the dimensions of the carbon nanotube. Implementation of various CNFET based Ternary Arithmetic circuits like Adders and Multipliers is extensively researched upon. The existing design for a CNFET based Multiplier is based on the classical Wallace approach which uses 3:1 Multiplexers along with Unary operators of Ternary logic. In this paper, design of a novel low power Single-Trit Multiplier and a Three-Trit Multiplier is proposed which uses a 2:1 Multiplexer based design approach. This design shows considerable improvement in terms of power consumption. From the Hspice simulation results, it is noted that the proposed Single-Trit Multiplier design results in up to 99% reduction in Power consumption, 56% reduction in Delay and 99.8% reduction in Power-Delay Product (PDP) when compared to an existing Single-Trit Multiplier. Proposed Three-Trit Multiplier results in savings in power consumption up to 98.2% and 98.5% reduction in PDP when compared to Three-Trit Multiplier design in the existing literature.

## TABLE OF CONTENTS

| CHAPTER NO | TITLE                                                                                           | PAGE NO |
|------------|-------------------------------------------------------------------------------------------------|---------|
|            | ABSTRACT                                                                                        | 5       |
|            | TABLE OF CONTENTS                                                                               | 6       |
|            | LIST OF FIGURES                                                                                 | 9       |
|            | LIST OF TABLES                                                                                  | 11      |
| 1          | INTRODUCTION                                                                                    | 12      |
|            | 1.1 MOTIVATION                                                                                  | 13      |
|            | 1.2 MULTIPLEXERS                                                                                | 13      |
|            | 1.2.1 BASIC MULTIPLEXING SWITCH                                                                 | 15      |
|            | 1.2.2 2- INPUT MULTIPLEXER DESIGN                                                               | 16      |
|            | 1.2.3 4-TO-1 CHANNEL MULTIPLEXER                                                                | 18      |
|            | 1.2.4 MULTIPLEXER INPUT LINE SELECTION                                                          | 18      |
|            | 1.2.5 4 CHANNEL MULTIPLEXER USING LOGIC<br>GATES                                                | 18      |
|            | 1.3 ADDERS                                                                                      | 20      |
|            | 1.3.1 HALF ADDER                                                                                | 21      |
|            | 1.3.2 FULL ADDER                                                                                | 23      |
|            | 1.4 HALF ADDER AND FULL ADDER USING K-MAP                                                       | 25      |
|            | 1.5 INTRODUCTION FOR CMOS                                                                       | 27      |
|            | 1.5.1 CMOS (COMPLEMENTARY METAL OXIDE<br>SEMICONDUCTOR) LOGIC HAS A FEW<br>DESIRABLE ADVANTAGES | 28      |

|          |                                                                                                       |           |
|----------|-------------------------------------------------------------------------------------------------------|-----------|
|          | 1.6 IMPLEMENTATION OF WALLACE MULTIPLIER<br>USING CMOS LOGIC DESIGN AND IMPLEMENTATION<br>BASIC GATES | 29        |
|          | 1.6.1 CMOS INVERTER OR NOT GATE                                                                       | 29        |
|          | 1.6.2 CMOS NAND GATE                                                                                  | 30        |
|          | 1.6.3 NOR GATE                                                                                        | 31        |
|          | 1.6.4 AND GATE                                                                                        | 33        |
|          | 1.6.5 OR GATE                                                                                         | 34        |
|          | 1.6.6 HALF ADDER                                                                                      | 35        |
|          | 1.6.7 FULL ADDER                                                                                      | 36        |
|          | 1.6.8 TERNARY MULTIPLIER                                                                              | 37        |
|          | 1.7 ORGANIZATION OF THE REPORT                                                                        | 37        |
| <b>2</b> | <b>LITERATURE SURVEY AND OBJECTIVES</b>                                                               | <b>38</b> |
|          | 2.1 LITERATURE SURVEY                                                                                 | 38        |
|          | 2.2 LITERATURE SUMMARY                                                                                | 42        |
|          | 2.3 OBJECTIVE OF THE PROJECT                                                                          | 43        |
| <b>3</b> | <b>PROCEDURES OF EXISTING SYSTEM</b>                                                                  | <b>44</b> |
|          | 3.1 TERNARY MULTIPLIER                                                                                | 44        |
|          | 3.2 FULL ADDER                                                                                        | 45        |
| <b>4</b> | <b>PROPOSED METHODOLOGY</b>                                                                           | <b>46</b> |
|          | 4.1 PROPOSED FULL ADDER                                                                               | 46        |
|          | 4.2 HALF ADDER                                                                                        | 47        |
|          | 4.3 TERNARY MULTIPLIER                                                                                | 48        |
|          | 4.4 HSPICE SOFTWARE                                                                                   | 49        |
|          | 4.4.1 SIMULATED SCREEN                                                                                | 50        |
|          | 4.5 COSMOSCOPE                                                                                        | 51        |
|          | 4.6 POWER                                                                                             | 51        |
|          | 4.6 DELAY                                                                                             | 52        |
|          | 4.7 POWER DELAY PRODUCT                                                                               | 52        |

|          |                                                     |           |
|----------|-----------------------------------------------------|-----------|
| <b>5</b> | <b>RESULT AND DISCUSSIONS</b>                       | <b>53</b> |
|          | 5.1 SIMULATION RESULTS OF LOGIC GATES               | 53        |
|          | 5.1.1 SIMULATION RESULT OF NOT GATE                 | 53        |
|          | 5.1.2 SIMULATION RESULT OF NAND GATE                | 54        |
|          | 5.1.3 SIMULATION RESULT OF NOR GATE                 | 54        |
|          | 5.1.4 SIMULATION RESULT OF AND GATE                 | 55        |
|          | 5.1.5 SIMULATION RESULT OF OR GATE                  | 55        |
|          | 5.2 SIMULATION RESULT OF HALF ADDER                 | 56        |
|          | 5.3 SIMULATION RESULT OF FULL ADDER                 | 56        |
|          | 5.4 SIMULATION RESULTS FOR TERNARY MULTIPLIER       | 57        |
|          | 5.5. PERFORMANCE ANALYSIS                           | 57        |
|          | 5.5.1 PERFORMANCE ANALYSIS OF LOGIC GATE            | 58        |
|          | 5.5.2 PERFORMANCE ANALYSIS OF HALF ADDER            | 60        |
|          | 5.5.3 PERFORMANCE ANALYSIS OF FULL ADDER            | 61        |
|          | 5.5.4 PERFORMANCE ANALYSIS OF TERNARY<br>MULTIPLIER | 61        |
|          | CONCLUSION                                          | 62        |
|          | REFERENCES                                          | 63        |

## LIST OF FIGURES

| FIGURE NO. | DESCRIPTION                                | PAGE NO. |
|------------|--------------------------------------------|----------|
| 1.1        | 4x1 Multiplexer                            | 14       |
| 1.2        | Basic Multiplexing Switch                  | 15       |
| 1.3        | Input Multiplexer Design                   | 16       |
| 1.4        | 4-1 Channel Multiplexer                    | 17       |
| 1.5        | Multiplexer Input Line Selection           | 18       |
| 1.6        | Four Channel Multiplexer Using Logic Gates | 18       |
| 1.7        | HALF ADDER                                 | 21       |
| 1.8        | Logic Circuit of Half Adder                | 23       |
| 1.9        | Full Adder                                 | 23       |
| 1.10       | Logic Diagram of Full Adder                | 24       |
| 1.11       | Half Adder Of K-Map                        | 25       |
| 1.12       | Full Adder Of K-Map                        | 26       |
| 1.13       | CMOS LOGIC DESIGN                          | 27       |
| 1.14       | NOT gate cmos design                       | 29       |
| 1.15       | A cmos two-input nand gate                 | 31       |
| 1.16       | A cmos two-input nor gate                  | 32       |
| 1.17       | A cmos two-input and gate                  | 33       |
| 1.18       | A cmos two-input or gate                   | 34       |
| 1.19       | Half adder cmos design                     | 35       |

|       |                                            |    |
|-------|--------------------------------------------|----|
| 1.20  | Full adder cmos design                     | 36 |
| 1.21  | Block diagram of ternary multiplier        | 37 |
| 3.1   | Full adder cmos design                     | 45 |
|       |                                            |    |
| 4.1   | FULL ADDER                                 | 46 |
| 4.2   | HALF ADDER CMOS DESIGN                     | 47 |
| 4.3   | TERNARY MULTIPLIER                         | 48 |
| 4.4   | HSPICE SOFTWARE                            | 49 |
| 4.5   | RESULT BROWER SCREEN IN HSPICE<br>SOFTWARE | 50 |
| 4.6   | SIMULATED SCREEN                           | 50 |
| 4.7   | COSMOSCOPE SOFTWARE                        | 51 |
| 5.1.1 | Simulation Result of NOT GATE              | 53 |
| 5.1.2 | Simulation Result of NAND GATE             | 54 |
| 5.1.3 | Simulation Result of NOR GATE              | 54 |
| 5.1.4 | Simulation Result of AND GATE              | 55 |
| 5.1.5 | Simulation Result of OR GATE               | 55 |
| 5.1.6 | Simulation Result of HALF ADDER            | 56 |
| 5.1.7 | Simulation Result of FULL ADDER            | 56 |
| 5.1.8 | Simulation Result of TERNARY MULTIPLIER    | 57 |

## LIST OF TABLES

| TABLE NO | DESCRIPTION                                       | PAGE NO |
|----------|---------------------------------------------------|---------|
| 1.1      | 2-Bit Half Adder Truth Table                      | 22      |
| 1.2      | Truth Table of Full Adder                         | 24      |
| 1.3      | The truth table for a NOT circuit.                | 30      |
| 1.4      | The truth table for a two-input NAND circuit      | 30      |
| 1.5      | The truth table for a two-input NOR circuit.      | 31      |
| 1.6      | The truth table for a two-input CMOS AND circuit. | 33      |
| 1.7      | The truth table for a two-input OR circuit.       | 34      |
| 3.1      | Unary operators for multiplier design             | 44      |
| 5.1      | Design analysis of 130NM NOT GATE                 | 58      |
| 5.2      | Design analysis of 130NM NAND GATE                | 58      |
| 5.3      | Design analysis of 130NM NOR GATE                 | 59      |
| 5.4      | Design analysis of 130NM AND GATE                 | 59      |
| 5.5      | Design analysis of 130NM OR GATE                  | 60      |
| 5.6      | Design analysis of 130NM HALF ADDER               | 60      |
| 5.7      | Design analysis of 130NM FULL ADDER               | 61      |
| 5.8      | Design analysis of 130NM TERNARY MULTIPLIER       | 61      |

# CHAPTER 1

## INTRODUCTION

Very large-scale integration was made possible with the wide adoption of the MOS transistor, originally invented by Mohamed M. Atalla and Dawon Kahng at Bell Labs in 1959. Atalla first proposed the concept of the MOS integrated circuit chip in 1960, followed by Kahng in 1961, both noting that the MOS transistor's ease of fabrication made it useful for integrated circuits. General Microelectronics introduced the first commercial MOS integrated circuit in 1964. In the early 1970s, MOS integrated circuit technology allowed the integration of more than 10,000 transistors in a single chip. This paved the way for VLSI in the 1970s and 1980s, with tens of thousands of MOS transistors on a single chip (later hundreds of thousands, then millions, and now billions).

The first semiconductor chips held two transistors each. Subsequent advances added more transistors, and as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known retrospectively as small-scale integration (SSI), improvements in technique led to devices with hundreds of logic gates, known as medium-scale integration (MSI). Further improvements led to large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and billions of individual transistors.

At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like ultra-large-scale integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use.

In 2008, billion-transistor processors became commercially available. This

became more commonplace as semiconductor fabrication advanced from the then-current generation of 65 nm processes. Current designs, unlike the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM (static random-access memory) cell, are still designed by hand to ensure the highest efficiency.

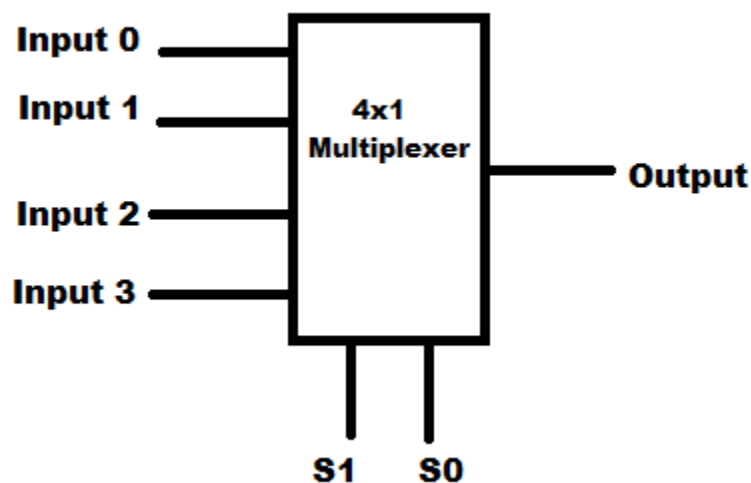
## 1.1 MOTIVATION

Multipliers play a very important role in many digital systems such as calculators, microprocessor, digital filters, and digital signal processor. They are also used in implementation of discrete Fourier transform, Correlation, Range measurement. The performance of any digital system is usually measured in terms of the used multipliers, since the process of the multiplication is very slow. Multiplication is nothing but adds and shift algorithm which is considered as series of repeated addition. That is the multiplicand adds itself the multiplier number of times. To implement this multiplication, it requires a large number of hardware elements and also it will operate at lower speed. Shift and Add, Reducing the Number of Partial Products, Radix-4 Booth Recoding and Sign Extension Tricks. The design of an efficient Highspeed Radix-4 Booth multiplier for both signed and unsigned numbers.

## 1.2 MULTIPLEXERS

One use for multiplexers is economizing connections over a single channel, by connecting the multiplexer's single output to the demultiplexer's single input. The image to the right demonstrates this benefit. In this case, the cost of implementing separate channels for each data source is higher than the cost and inconvenience of providing the multiplexing/demultiplexing functions. At the receiving end of the data link a complementary demultiplexer is usually required to break the single data stream back down into the original streams. In some cases, the far end system may have functionality greater than a simple demultiplexer; and while the demultiplexing still occurs technically, it may never be implemented discretely. This would be the case when, for instance, a multiplexer serves a number of IP network users; and then feeds

directly into a router, which immediately reads the content of the entire link into its routing processor; and then does the demultiplexing in memory from where it will be converted directly into IP sections. Often, a multiplexer and demultiplexer are combined together into a single piece of equipment, which is simply referred to as a multiplexer. Both circuit elements are needed at both ends of a transmission link because most communications systems transmit in both directions. In analog circuit design, a multiplexer is a special type of analog switch that connects one signal selected from several inputs to a single output.

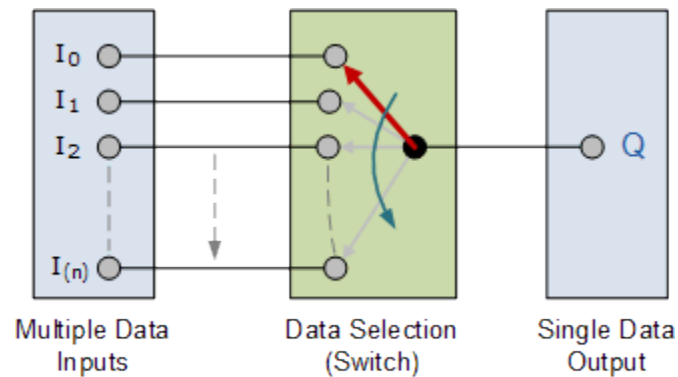


**Fig:1.1: 4x1 Multiplexer**

Multiplexing is the generic term used to describe the operation of sending one or more analogue or digital signals over a common transmission line at different times or speeds and as such, the device we use to do just that is called a Multiplexer. The multiplexer, shortened to “PPS” or “MPX”, is a combinational logic circuit designed to switch one of several input lines through to a single common output line by the application of a control signal. Multiplexers operate like very fast acting multiple position rotary switches connecting or controlling multiple input lines called “channels” one at a time to the output. Multiplexers, or PPS’s, can be either digital circuits made from highspeed logic gates used to switch digital or binary data or they can be analogue types using transistors, MOSFET’s or relays to switch one of the voltage or current inputs through to a single output. The most basic type of multiplexer device is that of a

one-way rotary switch as shown.

### 1.2.1 BASIC MULTIPLEXING SWITCH

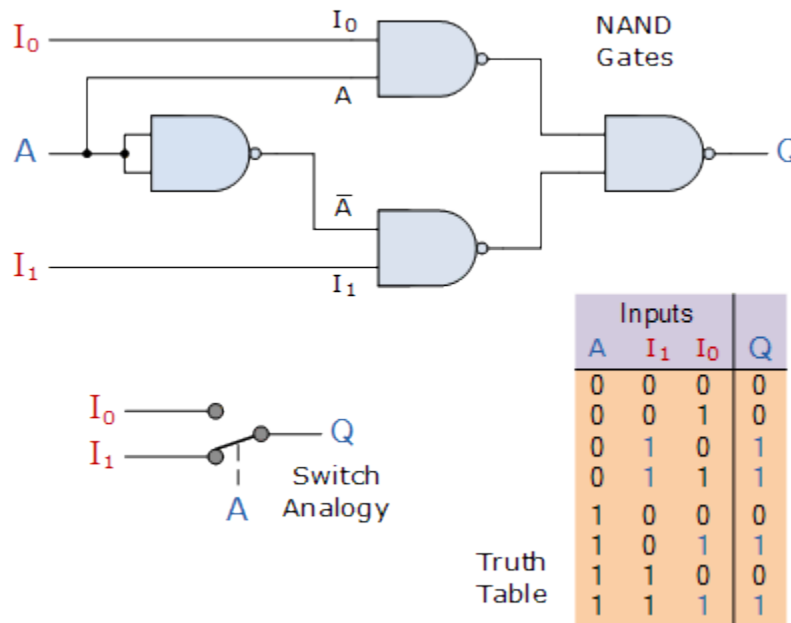


**Fig:1.2: Basic Multiplexing Switch**

The rotary switch, also called a wafer switch as each layer of the switch is known as a wafer, is a mechanical device whose input is selected by rotating a shaft. In other words, the rotary switch is a manual switch that you can use to select individual data or signal lines simply by turning its inputs “ON” or “OFF”. So how can we select each data input automatically using a digital device. In digital electronics, multiplexers are also known as data selectors because they can “select” each input line, are constructed from individual Analogue Switches encased in a single IC package as opposed to the “mechanical” type selectors such as normal conventional switches and relays. They are used as one method of reducing the number of logic gates required in a circuit design or when a single data line or data bus is required to carry two or more different digital signals. For example, a single 8-channel multiplexer. Generally, the selection of each input line in a multiplexer is controlled by an additional set of inputs called control lines and according to the binary condition of these control inputs, either “HIGH” or “LOW” the appropriate data input is connected directly to the output. Normally, a multiplexer has an even number of  $2^n$  data input lines and a number of “control” inputs that correspond with the number of data inputs. Note that multiplexers are different in operation to Encoders. Encoders are able to switch an  $n$ -bit input pattern to multiple output lines that

represent the binary coded (BCD) output equivalent of the active input. We can build a simple 2-line to 1-line (2-to-1) multiplexer from basic logic NAND gates as shown.

### 1.2.2 2-INPUT MULTIPLEXER DESIGN



**Fig:1.3: Input Multiplexer Design**

The input A of this simple 2-1line multiplexer circuit constructed from standard NAND gates acts to control which input (I<sub>0</sub> or I<sub>1</sub>) gets passed to the output at Q. From the truth table above, we can see that when the data select input, A is LOW at logic 0, input I<sub>1</sub> passes its data through the NAND gate multiplexer circuit to the output, while input I<sub>0</sub> is blocked. When the data select A is HIGH at logic 1, the reverse happens and now input I<sub>0</sub> passes data to the output Q while input I<sub>1</sub> is blocked. So by the application of either a logic “0” or a logic “1” at A we can select the appropriate input, I<sub>0</sub> or I<sub>1</sub> with the circuit acting a bit like a single pole double throw (SPDT) switch. As we only have one control line, (A) then we can only switch 2<sup>1</sup> inputs and in this simple example, the 2-input multiplexer connects one of two 1-bit sources to a common output, producing a 2-to-1-line multiplexer. We can confirm this in the following Boolean expression.

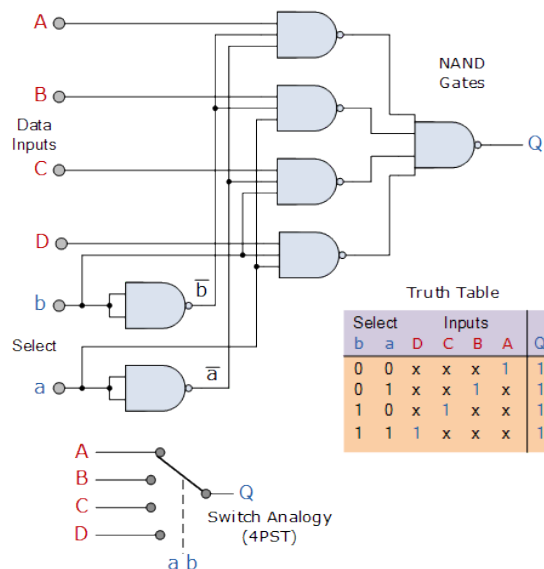
$$Q = A \cdot I_0 \cdot I_1 + A \cdot I_0 \cdot \bar{I}_1 + \bar{A} \cdot I_1 \cdot I_0 + \bar{A} \cdot I_1 \cdot \bar{I}_0$$

and for our 2-input multiplexer circuit above, this can be simplified too:

$$Q = A.I1 + A.I0$$

We can increase the number of data inputs to be selected further simply by following the same procedure and larger multiplexer circuits can be implemented using smaller 2-to-1 multiplexers as their basic building blocks. So for a 4-input multiplexer we would therefore require two data select lines as 4-inputs represents 2<sup>2</sup> data control lines give a circuit with four inputs, I0, I1, I2, I3 and two data select lines A and B as shown.

### 1.2.3 4-TO-1 CHANNEL MULTIPLEXER



**Fig:1.4: 4-1 Channel Multiplexer**

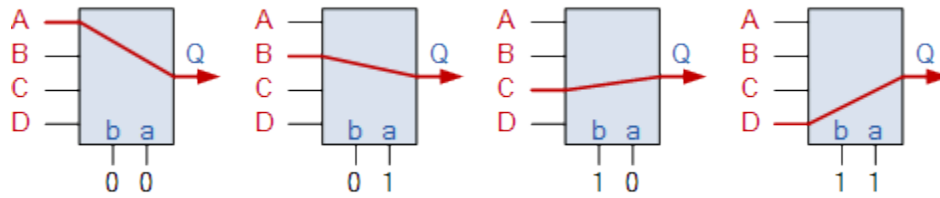
The Boolean expression for this 4-to-1 Multiplexer above with inputs A to D and data select lines a, b is given as:

$$Q = abA + abB + abC + abD$$

In this example at any one instant in time only ONE of the four analogue switches is closed, connecting only one of the input lines A to D to the single output at Q.

As to which switch is closed depends upon the addressing input code on lines “a” and “b”. So for this example to select input B to the output at Q, the binary input address would need to be “a” = logic “1” and “b” = logic “0”. Thus we can show the selection of the data through the multiplexer as a function of the data select bits as shown.

### 1.2.4 MULTIPLEXER INPUT LINE SELECTION

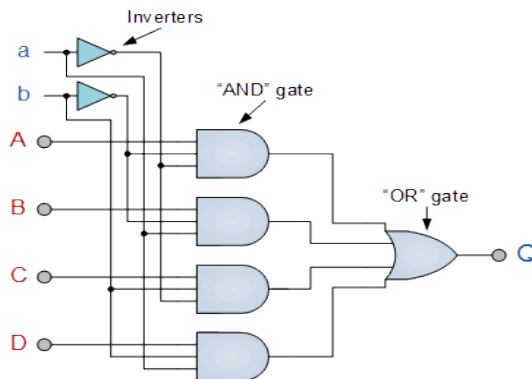


**Fig:1.5: Multiplexer Input Line Selection**

Adding more control address lines, (n) will allow the multiplexer to control more inputs as it can switch  $2^n$  inputs but each control line configuration will connect only ONE input to the output.

Then the implementation of the Boolean expression above using individual logic gates would require the use of seven individual gates consisting of AND, OR and NOT gates as shown.

### 1.2.5 4 CHANNEL MULTIPLEXER USING LOGIC GATES



**Fig:1.6: Four Channel Multiplexer Using Logic Gates**

The symbol used in logic diagrams to identify a multiplexer is as follows:

Multiplexers are not limited to just switching a number of different input lines or channels to one common single output. There are also types that can switch their inputs to multiple outputs and have arrangements or 4-to-2, 8-to-3 or even 16-to-4 etc configurations and an example of a simple Dual channel 4 input multiplexer (4-to-2) is given below:

Here in this example the 4 input channels are switched to 2 individual output

lines but larger arrangements are also possible. This simple 4-to-2 configuration could be used for example, to switch audio signals for stereo pre-amplifiers or mixers.

As well as sending parallel data in a serial format down a single transmission line or connection, another possible use of multi-channel multiplexers is in digital audio applications as mixers or where the gain of an analogue amplifier can be controlled digitally, for example.

Here, the voltage gain of the inverting operational amplifier is dependent upon the ratio between the input resistor,  $R_{IN}$  and its feedback resistor,  $R_f$  as determined in the Op-amp tutorials. A single 4-channel (Quad) SPST switch configured as a 4-to-1 channel multiplexer is connected in series with the resistors to select any feedback resistor to vary the value of  $R_f$ . The combination of these resistors will determine the overall voltage gain of the amplifier, ( $A_v$ ). Then the voltage gain of the amplifier can be adjusted digitally by simply selecting the appropriate resistor combination.

Digital multiplexers are sometimes also referred to as “Data Selectors” as they select the data to be sent to the output line and are commonly used in communications or high speed network switching circuits such as LAN’s and Ethernet applications. Some multiplexer IC’s have a single inverting buffer (NOT Gate) connected to the output to give a positive logic output (logic “1”, HIGH) on one terminal and a complimentary negative logic output (logic “0”, LOW) on another different terminal. It is possible to make simple multiplexer circuits from standard AND and OR gates as we have seen above, but commonly multiplexers/data selectors are available as standard i.c. packages such as the common TTL 74LS151 8-input to 1 line multiplexer or the TTL 74LS153 Dual 4-input to 1 line multiplexer. Multiplexer circuits with much higher number of inputs can be obtained by cascading together two or more smaller devices.

Then we can see that Multiplexers are switching circuits that just switch or route signals through themselves, and being a combinational circuit they are memoryless as there is no signal feedback path. The multiplexer is a very useful electronic circuit that has uses in many different applications such as signal routing, data communications and data bus control applications. When used with a demultiplexer, parallel data can be transmitted in serial form via a single data link such as a fibre-optic cable or telephone line and converted back into parallel data once again. The advantage is that only one

serial data line is required instead of multiple parallel data lines. Therefore, multiplexers are sometimes referred to as “data selectors”, as they select the data to the line. Multiplexers can also be used to switch either analogue, digital or video signals, with the switching current in analogue power circuits limited to below 10mA to 20mA per channel in order to reduce heat dissipation. In the next tutorial about combinational logic devices, we will look at the reverse of the Multiplexer called the Demultiplexer which takes a single input line and connects it to multiple output lines.

### 1.3 ADDERS

An adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic units, but also in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators, and similar operations.

Although adders can be constructed for many numerical representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder subtractor. Other signed number representations require more logic around the basic adder.

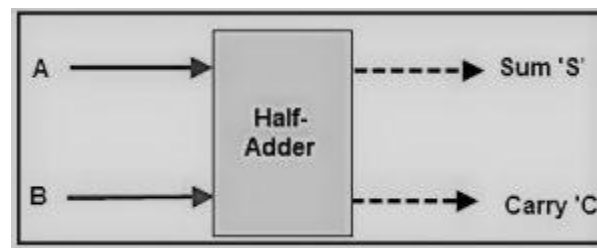
Binary carry-propagate adders have been extensively published, heavily attacking problems related to carry chain problem. Binary adders evolve from linear adders, which have a delay approximately proportional to the width of the adder, e.g. ripple-carry adder (RCA), to logarithmic - delay adders, such as the carry-look ahead adder (CLA). There are some additional performance enhancing schemes, including the carry-increment adder and the Ling adder that can further enhance the carry chain, however, in Very Large Scale Integration (VLSI) digital systems, the most efficient way of offering binary addition involves utilizing parallel-prefix trees, this occurs because they have the regular structures that exhibit logarithmic delay. Parallel-prefix adders compute addition in two steps: one to obtain the carry at each bit, with the next to compute the sum bit based on the carry bit. Unfortunately, prefix trees are algorithmically slower than fast

logarithmic adders, such as the carry propagate adders, however, their regular structures promote excellent results when compared to traditional CLA adders.

This happens within VLSI architectures because a carry-look ahead adder, such as the one implemented in one of Motorola's processors, tends to implement the carry chain in the vertical direction instead of a horizontal one, which has a tendency to increase both wire density and fan-in/out dependence. Therefore, although logarithmic adder structures are one of the fastest adders algorithmically, the speed efficiency of the carry-look ahead adder has been hampered by diminishing returns given the fan-in and fan-out dependencies as well as the heavy wire load distribution in the vertical path. In fact, a traditional carry-look ahead adder implemented in VLSI can actually be slower than traditional linear-based adders, such as the Manchester carry adder.

### 1.3.1 HALF ADDER

So, coming to the scenario of half adder, it adds two binary digits where the input bits are termed as augend and addend and the result will be two outputs one is the sum and the other is carry. To perform the sum operation, XOR is applied to both the inputs, and AND gate is applied to both inputs to produce carry.



**Fig:1.7: HALF ADDER**

Whereas in the full adder circuit, it adds 3 one-bit numbers, where two of the three bits can be referred to as operands and the other is termed as bit carried in. The produced output is 2-bit output and these can be referred to as output carry and sum. By using a half adder, you can design simple addition with the help of logic gates. Let's see an example of adding two single bits.

The 2-bit half adder truth table is as below:

**Table:1.1: 2-Bit Half Adder Truth Table**

| INPUTS |   | OUTPUTS |       |
|--------|---|---------|-------|
| A      | B | SUM     | CARRY |
| 0      | 0 | 0       | 0     |
| 0      | 1 | 1       | 0     |
| 1      | 0 | 1       | 0     |
| 1      | 1 | 0       | 1     |

$$0+0 = 0$$

$$0+1 = 1$$

$$1+0 = 1$$

$$1+1 = 10$$

These are the least possible single-bit combinations. But the result for 1+1 is 10, the sum result must be re-written as a 2-bit output. Thus, the equations can be written as

$$0+0 = 00$$

$$0+1 = 01$$

$$1+0 = 01$$

$$1+1 = 10$$

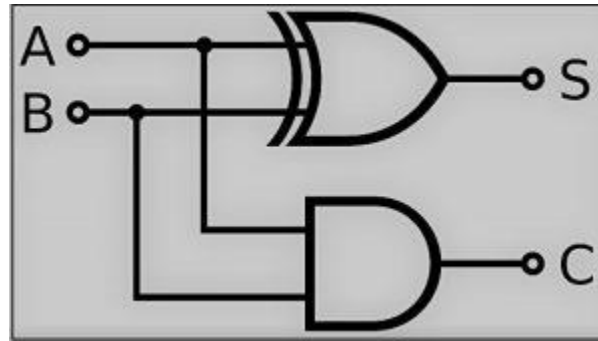
The output '1' of '10' is carry-out. 'SUM' is the normal output and 'CARRY' is the carry-out. Now it has been cleared that a 1-bit adder can be easily implemented with the help of the XOR Gate for the output 'SUM' and an AND Gate for the 'Carry'. For instance, when we need to add, two 8-bit bytes together, then it can be implemented by using a full-adder logic circuit. The half-adder is useful when you want to add one binary digit quantities.

A way to develop two-binary digit adders would be to make a truth table and reduce it. When you want to make a three binary digit adder, the half adder addition operation is performed twice. In a similar way, when you decide to make a four-digit adder, the operation is performed one more time. With this theory, it was clear that the implementation is simple, but development is a time taking process.

The simplest expression uses the exclusive OR function:

$$\text{Sum} = A \text{ XOR } B$$

$$\text{Carry} = A \text{ AND } B$$



**Fig:1.8: Logic Circuit For Half Adder**

And an equivalent expression in terms of the basic AND, OR, and NOT is:

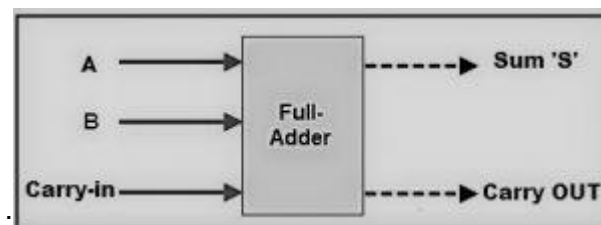
$$\text{SUM} = A.B + A.B'$$

The implementation of half adder can be done through high-speed CMOS digital logic integrated circuits like 74HCxx series which includes the SN74HC08 (7408) & SN74HC86 (7486).

The main reason to call these binary adders like Half Adders is, that there is no range to include the carry bit using an earlier bit. So, this is a main limitation of HAs once used like binary adder particularly in real-time situations which involve adding several bits. So this limitation can be overcome by using the full adders.

### 1.3.2 FULL ADDER

This adder is difficult to implement when compared to half-adder.



**Fig:1.9: Full Adder**

The full-adder has three inputs and two outputs, whereas half adder has only two inputs and two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. When a full-adder logic is designed, you string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

**Table:1.2: Truth Table of Full Adder**

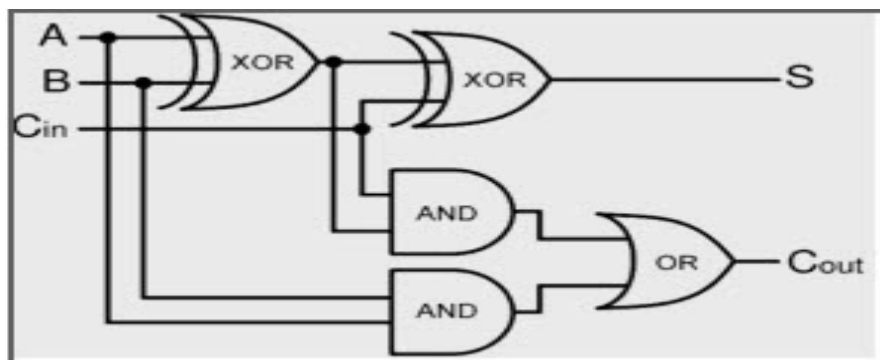
| INPUTS |   |      | OUTPUT |   |
|--------|---|------|--------|---|
| A      | B | C-IN | C-OUT  | S |
| 0      | 0 | 0    | 0      | 0 |
| 0      | 0 | 1    | 0      | 1 |
| 0      | 1 | 0    | 0      | 1 |
| 0      | 1 | 1    | 1      | 0 |
| 1      | 0 | 0    | 0      | 1 |
| 1      | 0 | 1    | 1      | 0 |
| 1      | 1 | 0    | 1      | 0 |
| 1      | 1 | 1    | 1      | 1 |

The difference between a half-adder and a full-adder is that the output carry is designated as C-OUT and the normal output is represented as S which is 'SUM'.

With the above full adder truth-table, the implementation of a full adder circuit can be understood easily. The SUM 'S' is produced in two steps: By XORing the provided inputs 'A' and 'B'. The result of A XOR B is then XORed with the C-IN

This generates SUM and C-OUT is true only when either two of three inputs are HIGH, then the C-OUT will be HIGH. So, we can implement a full adder circuit with the help of two half adder circuits. Initially, the half adder will be used to add A and B to produce a partial Sum and a second-half adder logic can be used to add C-IN to the Sum produced by the first half adder to get the final S output.

If any of the half adder logic produces a carry, there will be an output carry. So, C-OUT will be an OR function of the half-adder Carry outputs. Take a look at the implementation of the full adder circuit shown below.



**Fig:1.10: Logic Diagram of Full Adder**

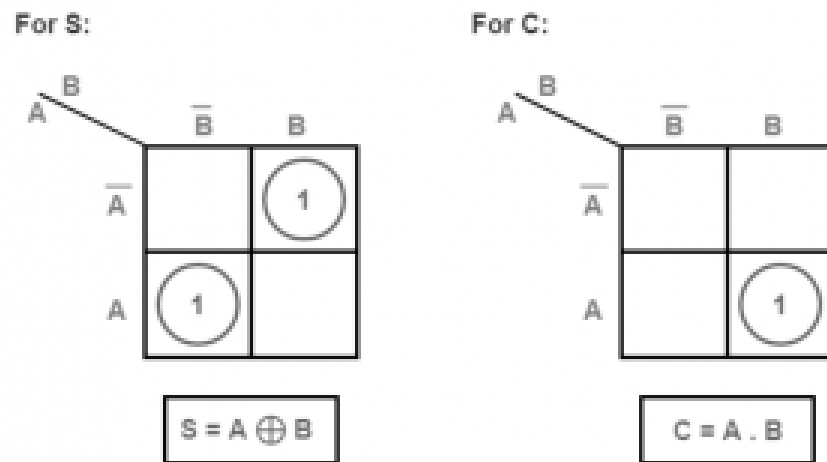
The implementation of larger logic diagrams is possible with the above full adder logic a simpler symbol is mostly used to represent the operation. Given below is a simpler schematic representation of a one-bit full adder.

With this type of symbol, we can add two bits together, taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude. In a computer, for a multi-bit operation, each bit must be represented by a full adder and must be added simultaneously. Thus, to add two 8-bit numbers, you will need 8 full adders which can be formed by cascading two of the 4-bit blocks.

#### 1.4 HALF ADDER AND FULL ADDER USING K-MAP

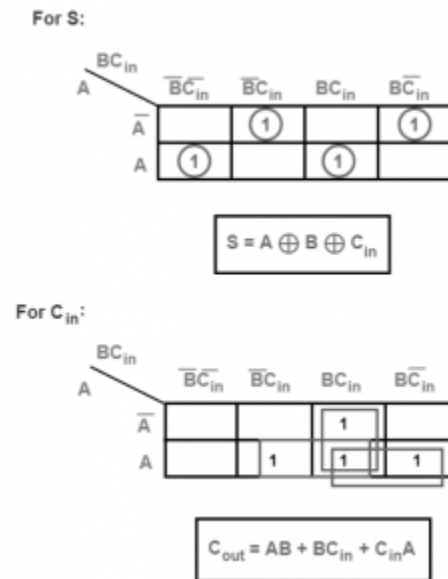
Even the sum and carry outputs for half adder can also be obtained with the method of Karnaugh map (K-map). The half adder and full adder boolean expression can be obtained through K-map. So, the K-map for these adders is discussed below.

The half adder K-map is



**Fig:1.11: Half Adder of K-Map**

The full adder K-Map is



**Fig:1.12: Full Adder of K-Map**

Logical Expression of SUM and Carry

The logical expression of sum (S) can be determined based on the inputs mentioned in the table.

$$= A'B'C_{in} + A'BCC_{in}' + AB'C_{in}' + ABC_{in}$$

$$= C_{in} (A'B' + AB) + C_{in}' (A'B + AB')$$

$$= C_{in} \text{ EX-OR } (A \text{ EX-OR } B)$$

$$= (1, 2, 4, 7)$$

The logical expression of the carry ( $C_{out}$ ) can be determined based on the inputs mentioned in the table.

$$= A'B C_{in} + AB'C_{in} + AB C_{in}' + ABC_{in}$$

$$= AB + BC_{in} + AC_{in}$$

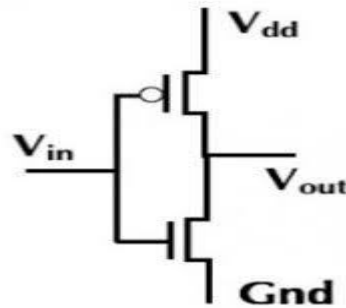
$$= (3, 5, 6, 7)$$

With the above-mentioned truth tables, the results can be obtained and the procedure is A combinational circuit combines the different gates in the circuit where those can be an encoder, decoder, multiplexer and demultiplexer. The characteristics of combinational circuits are as follows.

The output at any instant of time is based only on the levels that are present at input

terminals. It does not use any memory. The previous state of input does not have any effect on the present state of the circuit. It can have any number of inputs and number of outputs.

### 1.5 INTRODUCTION FOR CMOS



**Fig:1.13: CMOS LOGIC DESIGN**

Complementary metal oxide semiconductor, abbreviated as CMOS, is a technology for constructing integrated circuits. CMOS technology is used in microprocessors, microcontrollers, static RAM, and other digital logic circuits. CMOS technology is also used for several analog circuits such as image sensors (CMOS sensor), data converters, and highly integrated transceivers for many types of communication. In 1963, while working for Fairchild Semiconductor, Frank Wanlass patented CMOS (US patent 3,356,858). CMOS is also sometimes referred to as complementary-symmetry metal oxide semiconductor (or COS-MOS).

The word "complementary symmetry" refers to the fact that the typical design style with CMOS uses complementary and symmetrical pairs of p-type and n-type metal oxide semiconductor field effect transistors (MOSFETs) for logic functions. Two important characteristics of CMOS devices are high noise immunity and low static power consumption. Since one transistor of the pair is always off, the series combination draws significant power only momentarily during switching between on and off states. Consequently, CMOS devices do not produce as much waste heat as other forms of logic, for example transistor transistor logic (TTL) or NMOS logic, which normally have some standing current even when not changing state. CMOS also allows a high density of logic functions on a chip. It was primarily for this reason that CMOS

became the most used technology to be implemented in VLSI chips. The phrase "metaloxidesemiconductor" is a reference to the physical structure of certain field-effect transistors, having a metal gate electrode placed on top of an oxide insulator, which in turn is on top of a semiconductor material. Aluminum was once used but now the material is polysilicon.

Other metal gates have made a comeback with the advent of high-k dielectric materials in the CMOS process, as announced by IBM and Intel for the 45 nanometer node and beyond.

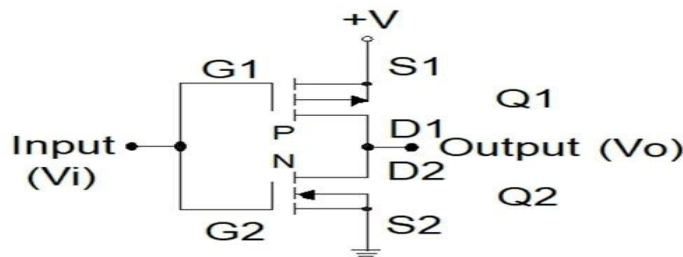
#### 1.5.1 CMOS (COMPLEMENTARY METAL OXIDE SEMICONDUCTOR) LOGIC HAS A FEW DESIRABLE ADVANTAGES:

- High input impedance. The input signal is driving electrodes with a layer of insulation (the metal oxide) between them and what they are controlling. This gives them a small amount of capacitance, but virtually infinite resistance. The current into or out of CMOS input held at one level is just leakage, usually 1 nano Ampere or less.
- The outputs actively drive both ways. item[•]item[•] The outputs are pretty much rail-to-rail.
- CMOS logic takes very little power when held in a fixed state. The current consumption comes from switching as those capacitors are charged and discharged. Even then, it has good speed to power ratio compared to other logic types.
- CMOS gates are very simple. The basic gate is an inverter, which is only two transistors. This together with the low power consumption means it lends itself well to dense inte7 International Journal of Pure and Applied Mathematics Special Issue 773 gration. Or conversely, you get a lot of logic for the size, cost and power.

## 1.6 IMPLEMENTATION OF WALLACE MULTIPLIER USING CMOS LOGIC DESIGN AND IMPLEMENTATION BASIC GATES

### 1.6.1 CMOS INVERTER OR NOT GATE

A NOT gate reverses the input logic state. Fig 1.14 shows a NOT gate employing two series-connected enhancement-type MOSFETS, one n-channel (NMOS) and one p-channel (PMOS).



**Fig:1.14: NOT GATE CMOS DESIGN**

The input is connected to the gate terminal of the two transistors, and the output is connected to both drain terminals.

Applying  $+V$  (logic 1) to the input ( $V_i$ ), transistor Q2 is “on,” and transistor Q1 remains “off.” Under this condition, the output voltage ( $V_o$ ) is close to 0 V (logic 0).

Connecting the input to ground ( $V_i = 0$  V), transistor Q2 is “off,” and transistor Q1 is “on.” Now, the output voltage is close to  $+V$  (logic 1).

Table 1.3 summarizes these results.

**Table:1.3: The truth table for a NOT circuit.**

| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

### 1.6.2 CMOS NAND GATE

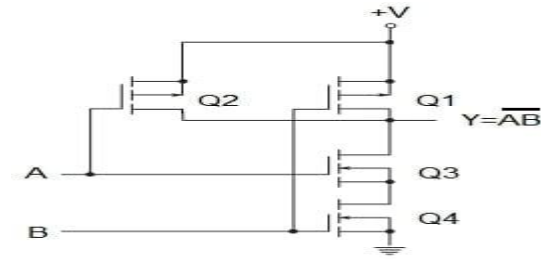
NAND denotes NOT-AND.

Table 1.4 shows the truth table for a NAND circuit.

**Table:1.4: The truth table for a two-input NAND circuit.**

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Figure 1.15 shows a CMOS two-input NAND gate. P-channel transistors Q1 and Q2 are connected in parallel between +V and the output terminal. N-channel transistors Q3 and Q4 are connected in series between the output terminal and ground.



**Fig:1.15: A CMOS two-input NAND gate**

With Q3 and Q4 transistors “on” and Q1 and Q2 transistors “off,” the output is a logic 0. This condition happens when both inputs, A and B, are logic 1, confirming the lowest row in the above truth table.

With logic 0 in inputs A and B, Q3 and Q4 transistors are “off,” and Q1 and Q2 transistors are “on,” producing a logic 1 output. This is consistent with the first row of the truth table.

When one of the inputs is a logic “1” and the other one is a logic “0”, either Q3 is “off” and Q2 is “on” or Q4 is “off” and Q1 is “on.” The output in both cases is a logic “1,” validating the second and the third rows of the truth table.

### 1.6.3 NOR GATE

NOR signifies NOT-OR.

Table 1.5 shows the truth table for a NOR circuit.

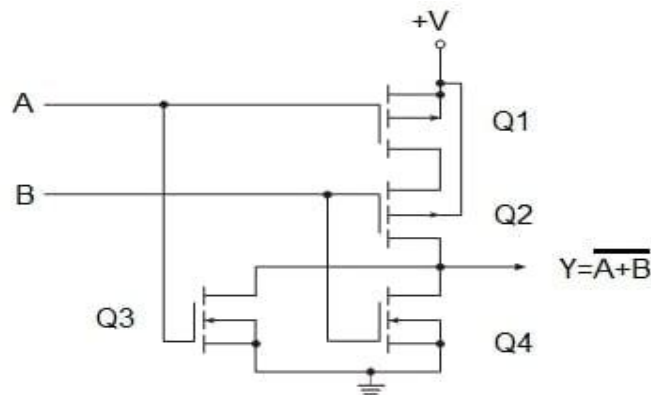
**Table:1.5: The truth table for a two-input NOR circuit.**

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |

|   |   |   |
|---|---|---|
| 1 | 1 | 0 |
|---|---|---|

The output of a NOR gate is logic 1 with logic 0 in both inputs. The outcomes for other input combinations are logic 0.

Figure 1.16 shows a CMOS two-input NOR gate. P-channel transistors Q1 and Q2 are connected in series between +V and the output terminal. N-channel transistors Q3 and Q4 are connected in parallel between the output and ground.



**Fig:1.16: A CMOS two-input NOR gate.**

When both inputs, A and B, are logic 0, Q1 and Q2 are “on,” and Q3 and Q4 are “off,” and the output is logic 1. This confirms the first row of the truth table above.

With both inputs logic 1, Q3 and Q4 are “on,” and Q1 and Q2 are “off,” producing a logic 0 output that confirms the last row of the truth table.

For the two remaining input combinations, either Q1 is “off” and Q3 is “on” or Q2 is “off” and is Q4 “on”. In these cases, the output is logic 0 which is consistent with the above

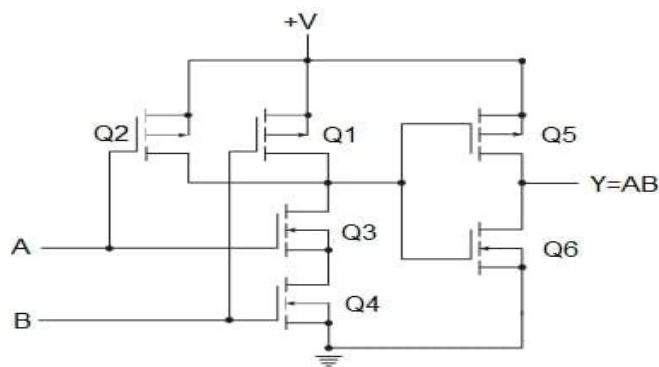
truth table.

#### 1.6.4 AND GATE

We can say that an AND gate is a NOT-NOT-AND or NOT-NAND. Then, it is just a NAND gate followed by an inverter.

**Table:1.6: The truth table for a two-input CMOS AND circuit.**

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



**Fig:1.17: A CMOS two-input AND gate.**

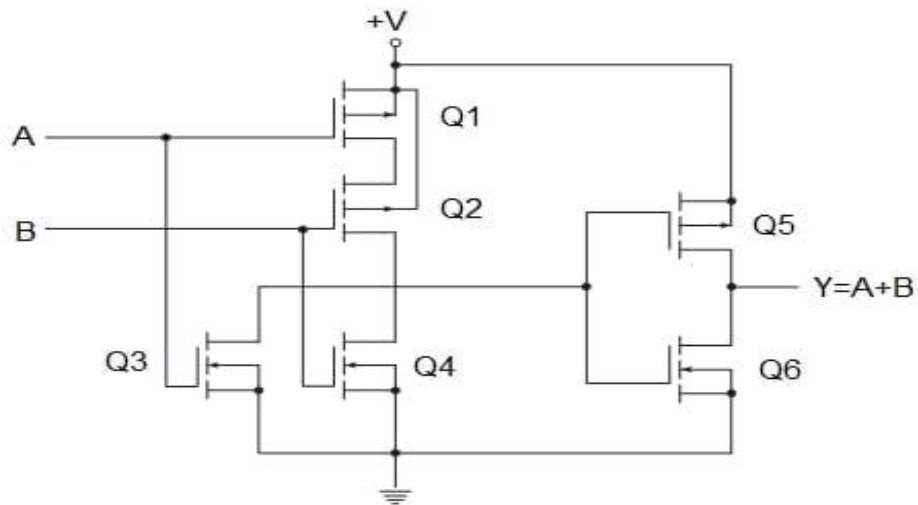
### 1.6.5 OR GATE

An OR gate is a NOT-NOT-OR or NOT-NOR. Then, it is a NOR gate followed by an inverter.

Table 1.7 shows the truth table for the OR circuit.

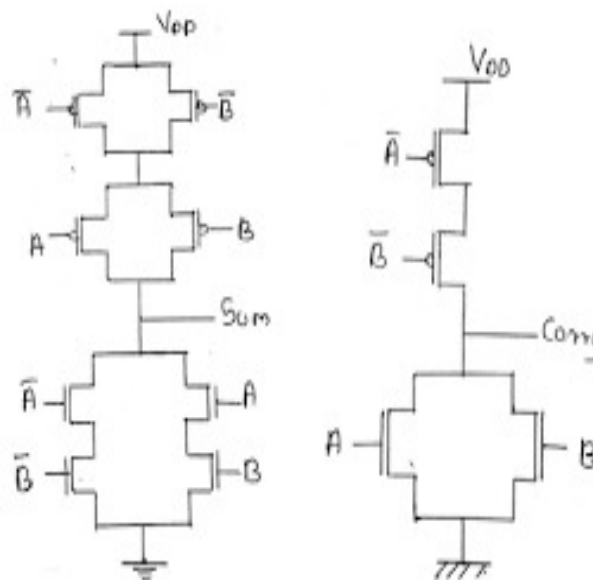
**Table:1.7: The truth table for a two-input OR circuit.**

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



**Fig:1.18: A CMOS two-input OR gate.**

#### 1.6.6 HALF ADDER



**Fig:1.19: HALF ADDER CMOS DESIGN**

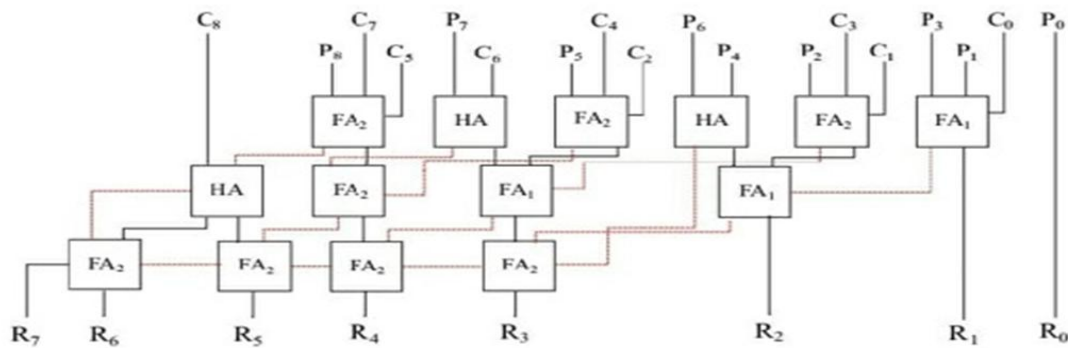
The half adder adds two single binary digits A and B. It has two outputs, sum (S) and carry (C). The carry signal represents an overflow into the next digit of a multi-digit addition. The value of the sum in decimal system is  $2C + S$ . The simplest half-adder design, pictured on the right, incorporates an XOR gate for S and an AND gate for C. The Boolean logic for the sum (in this case S) will be  $A'B + AB'$  whereas for carry (C) will be  $AB$ . With the addition of an OR gate to combine their carry outputs, two half adders can be combined to make a full adder. The half adder adds two input bits and generates a carry and sum, which are the two outputs of a half adder

#### 1.6.7 FULL ADDER

A full adder circuit can be implemented with the help of two half adder circuits. The first half adder will be used to add A and B to produce a partial Sum. The second half adder logic can be used to add  $C_{IN}$  to the Sum produced by the first half adder to get the final S output. If any of the half adder logic produces a carry, there will be an output carry. Thus,  $C_{OUT}$  will be an OR function of the half-adder Carry outputs.



## 37



Block Diagram of Ternary Multiplier

**Fig:1.21: BLOCK DIAGRAM OF TERNARY MULTIPLIER**

## 1.7 ORGANIZATION OF THE REPORT

**CHAPTER 1** discuss about VLSI, Motivation, Multiplexing, Adders, Logic gates, Ternary multiplier. It also discusses about the various multipliers and organization of the project work. It also discusses about the various multipliers and organization of the project work.

**CHAPTER 2** delivers detailed literature survey related to the project, literature summary, problem definition, objectives of the project.

**CHAPTER 3** describes the working and existing techniques of Ternary multipliers.

**CHAPTER 4** discusses the proposed Ternary multiplier, adders, simulation HSPICE.

**CHAPTER 5** provides the simulation results and performance analysis of the existing and proposed project work. It does a full comparative analysis on how the proposed work is better off than the existing design.

provides the conclusion project work. It also gives the schema of the future work that is how the project could be enhanced further. Finally, the references referred to lead the project are given in brief.

## CHAPTER 2

### LITERATURE SURVEY AND OBJECTIVES

This chapter discusses about the literature survey and summary. It also gives a detailed idea about analysis of an efficient ternary multiplier for low power applications.

#### 2.1 LITERATURE SURVEY

**P. C. A. Antoniou, “Low power dissipation mos ternary logic Balla and family,” no. 2, pp. 217–225.-1984**

To design basic ternary arithmetic and memory circuits comprised of a set of inverters, NOR gates, and NAND gates. To use and synthesize complex ternary arithmetic circuits and shift registers. To know ternary circuits like low power dissipation, and reduced propagation delay and component count. The proposed designs have been simulated in Synopsys HSPICE using the analytical model. The operating voltage for our study is 0.9 V and the channel length is 32 nm. The noise margin has been calculated using DC analysis in HSPICE. In particular, one input is varied from 0 to VDD (0.9 V) while keeping the other inputs fixed at one of the three values, namely, 0, VDD/2 or VDD. The designs in single trit and counts for larger sizes have been estimated assuming a Wallace-like structure. The performance of the proposed multiplier has also been studied for different drive strengths and plotted. Change in PDP as the drive strength increases is small. The paper concludes with an implementation of the cyclic convolution, an application in which a significant advantage can be gained through the use of ternary digital hardware.

**D. Mateo and A. Rubio, “Design and implementation of a 5x5 trits multiplier in a quasi adiabatic ternary cmos logic,” *IEEE J. Solid-State Circuits*, vol. 33, no. 7, pp. 1111–1116, 1998.**

To design low power digital IC's. To design a 5\*5 ternary digit multiplier and implement in a 0.7- $\mu$ m CMOS technology. In order to validate the functionality of the QAT logic and evaluate its performance, a 5\*5 trits multiplier has been implemented in a 0.7  $\mu$ m double-metal single-poly CMOS technology. The IC has a total area of 2.6 mm. The consumption of the multiplier is measured by using a Tek-DSA602A digitizing signal analyzer by sampling the voltage and current of each clock. out is an output of the multiplier, giving “1” as a result. The energy dissipated in the IC in this switching, and delivered by the mentioned clock, is the difference between the energy given by the clock and the energy returned to it. Doing that for all of the clocks, the total consumption of the QAT multiplier is obtained. It is one order of magnitude better than the PDP of a conventional binary CMOS 8\*8 mult , and seven times better than the PDP of a multiplier implemented in other quasi-adiabatic logic. The area saving of the QAT multiplier in front of the smallest fully adiabatic but binary multiplier is 60% in the number of devices, as well as having an intrinsic benefit in routing because of having five trits in front of 8 bits.

Results show a satisfactory power saving with respect to conventional and other quasi-adiabatic binary multipliers, and a decrease of the area needed with respect to a fully adiabatic binary one. A special feature of this logic is to be ternary, in order to diminish the area needed with respect to other adiabatic binary logics, keeping a satisfactory power saving. A 5 5 trits multiplier has been implemented using this logic. Measurements show a power–delay product one order of magnitude better than a conventional 8 8 bit CMOS multiplier and smaller than the PDP of a multiplier implemented in other quasi-adiabatic logic; and there is an important area saving with respect to fully adiabatic 8 8 bit multipliers.

**Lin, Sheng, Kim, Y. Bin, Lombardi, and Fabrizio, “CNTFET-based design of ternary logic gates and arithmetic circuits,” *IEEE Transactions on Nanotechnology*, vol. 10, no. 2, pp. 217–225, 2011.**

To design Ternary logic is a promising alternative to the conventional binary logic, since it is possible to accomplish simplicity and energy efficiency in modern digital design due to the reduced circuit overhead such as interconnects and chip area. To design a resistive-load CNTFET-based ternary logic has been proposed to implement ternary logic based on CNTFET. All simulations have been performed in HSPICE using the CNTFET mode. A complete set of ternary gates has been implemented using multidiameter CNTFET. simulation results using SPICE are reported to show that the proposed ternary logic gates consume significantly lower power and delay than the previous resistive-load CNTFET gates implementations. The utilization of the proposed ternary gates combined with binary gates results in over 90% reductions in terms of the power delay product. This paper has presented the design of the new a ternary logic family based on CNTFETs. As the threshold voltage of the CNTFET is function of the geometry of the CNTFET, a novel multidiameter CNTFET-based ternary design has been pursued, Compared with previous CNTFET-based designs, the proposed ternary gates achieve high performance, low power, and small area due to the removal of resistors and the utilization of binary gates in the design of arithmetic circuits.

**M. Moaiyeri, a. Doostaregan, and K. Navi, “Design of energy-efficient and robust ternary circuits for nanotechnology,” *IET Circuits, Devices & Systems*, vol. 5, no. 4, p. 285, 2011.**

The carbon nanotube field effect transistor (CNFET)-based circuits implements all the possible kinds of ternary logic, including negative, positive and standard ternary logics, in one structure. To demonstrate great improvements in terms of speed, power consumption and insusceptibility. This property of CNFETs makes them very suitable for the multiple-V<sub>t</sub> design method. The proposed circuits are simulated exhaustively, using

Synopsys HSPICE with 32 nm-CNFET technology in various test situations and different supply voltages. Simulation results demonstrate great improvements in terms of speed, power consumption and insusceptibility. To process variations with respect to other conventional and state of- the-art 32 nm complementary metal-oxide semiconductor and CNFET-based ternary circuits. Novel energy-efficient and robust CNFET-based ternary logic and arithmetic circuits for nanotechnology have been proposed in this paper. Each of the proposed ternary logic gates has included all the negative, positive and standard ternary logics. The circuits have been exhaustively tested and them simulation results have confirmed significant improvements in terms of performance and robustness compared to the other conventional and state-of-the-art CMOS and CNFET-based ternary circuits, existing in the literature. The simulation results also demonstrate that utilising the proposed circuits in designing ternary arithmetic circuits leads to higher operating speed and lower power consumption, compared to the previously presented designs.

**B. Srinivasu and K. Sridharan, “Low-Complexity Multiternary Digit Multiplier Design in CNTFET Technology,” *IEEE Trans. on Circuits and Systems II*, vol. 63, no. 8, 2016.**

To design a novel ternary multiplexer based on classical Wallace multiplier and includes a novel ternary multiplexer design requiring only a small number of CNTFETs. Two ternary full adder configurations are also proposed based on examination of the multiplier structure. To design a new single trit multiplier which requires 67% less CNTFETs compared to a recent design. Carbon Nanotube Field Effect Transistor (CNT-FET) technology using unary operators of multivalued logic. low-CNTFET count design of a multiternary digit (trit) multiplier based on the classical Wallace multiplier. HSPICE simulations reveal low power-delay product for the proposed designs for different choices of drive strength. Further, the designs are comparable to prior works with respect to noise margin. A low CNTFET-count multiternary digit multiplier is presented. Two custom full-adder configurations are also developed. HSPICE simulations reveal excellent performance with respect to power, delay as well as noise margin.

**S. A. ErfanShahrom, “A new low power multiplexer based ternary multiplier using CNTFETs,” *AEU - International Journal of Electronics and Communications*, pp. 191–207, 2018.**

To design a circuit using two supply voltage,  $V_{DD}$  and  $V_{DD}/2$ , the circuit is designed so as  $V_{DD}/2$  could be transmitted to output directly for logic ‘1’ to eliminate direct current from source to ground. The carry output is ‘1’ only when both inputs are ‘2’, otherwise equals ‘0’. As opposite, in a PT-NAND, output is ‘0’ only when both inputs are ‘2’, otherwise equals ‘2’, so the Carry can be implemented by a PT-NAND and a binary inverter. As it can be seen 6 CNTFETs is needed and there is no path for direct current between sources and ground. CNTFETs are used to design proposed multiplier. The number of transistor is reduced from 80 and 26 in work to 23 in this design, there is no path for direct current from sources to ground for all component. The multiplier can be extended to multi-bit multiplier using single bit multiplier, full adders and half adders according to multiplication algorithm, here, for example, two bit multiplier is implemented.

The designed circuit has been simulated using Hspice software by Stanford 32nm CNTFET library To study the operation of P-MUX and N-MUX, the simulation output of P-MUX and N-MUX are shown in different state. The proposed full adder has a lower power dissipation and PDP according to remove direct current. The output waveforms of two bit multiplier, as to fit the page, some states are randomly selected. The result of simulation, using Hspice software and Stanford 32nm CNTFET library with the voltage of 0.9v as expected, indicate much lower power dissipation and power delay product. In this paper Producing logical level ‘1’,  $V_{DD}/2$  is directly transmitted to output instead of voltage division of  $V_{DD}$ . The direct current between source and ground will be cutoff. This result in high decreasing the static power dissipation and PDP consequently.

## 2.2 LITERATURE SUMMARY

That is the multiplicand adds itself the multiplier number of times. To implement this

multiplication, it requires a large number of hardware elements and also it will operate at lower speed.

### 2.3 OBJECTIVE OF THE PROJECT

- To increase the efficiency of the Ternary Multiplier for Low Power applications.
- In order to improve Efficiency Low Power Techniques will be applied in all Circuits.
- To compare, the characteristics of MOSFET Technology in terms of Power, Delay and Power Delay Product.
- To identify the Feasibility of Ternary Multiplier

## CHAPTER 3

### PROCEDURES OF EXISTING SYSTEM

#### 3.1 TERNARY MULTIPLIER

A CMOS based ternary Wallace-Tree Multiplier has been implemented. For lower power-delay product, a design technique using both binary logic gates and ternary logic gates has been implemented. For example, a ternary single-digit Multiplier and ternary half-adder has reduced PDP with respect to their counter parts using CNFETs based ternary gates only. Ternary multi-digit multiplier implemented using 3:1 multiplexers and unary operators has been presented .

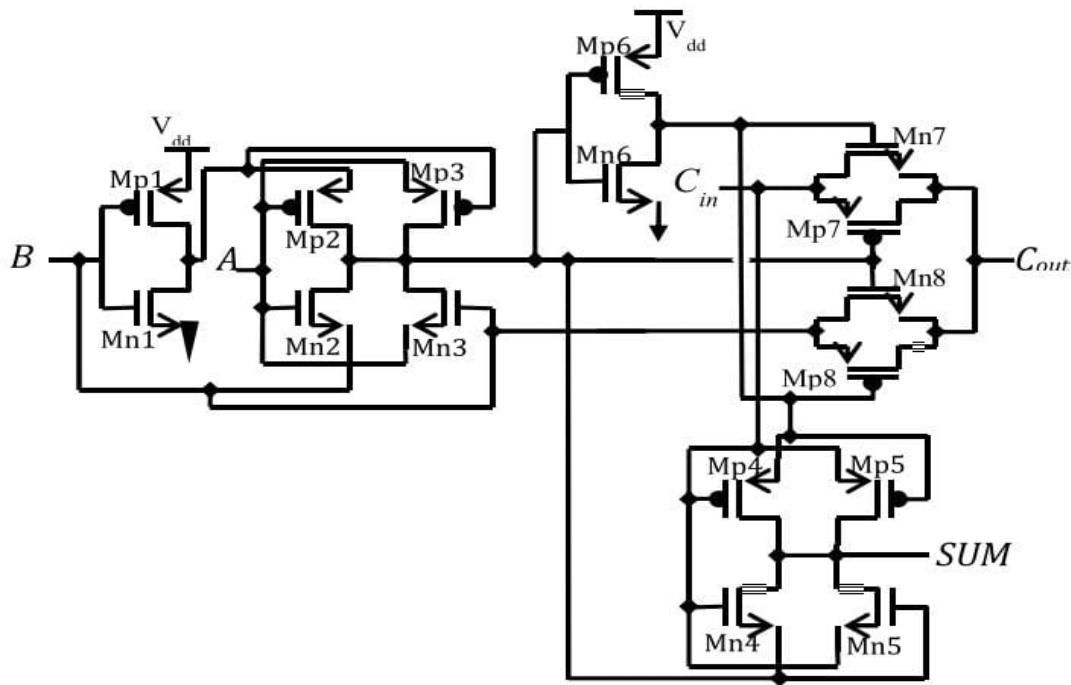
**Table:3.1: Unary operators for multiplier design**

| <i>A</i> | <i>AP</i> | <i>AN</i> | <i>A1</i> | <i>A2</i> | <i>A1</i> | <i>A2</i> | <i>AP</i> | <i>AN</i> |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0        | 2         | 2         | 1         | 2         | 1         | 0         | 0         | 0         |
| 1        | 2         | 0         | 2         | 0         | 0         | 2         | 0         | 2         |
| 2        | 0         | 0         | 0         | 1         | 2         | 1         | 2         | 2         |

The design includes a novel 3:1 ternary multiplexer which requires less number of CNFETs and is based on the classical Wallace multiplier approach. Two full adder configurations for the corresponding multi-digit multiplier can be found. It has lower delay but higher power when compared to prior works. The Single-Trit Multiplier circuit in contains *VDD* to ground paths which allows path for leakage and thereby giving rise to larger overall power consumption

### 3.2 FULL ADDER

A full adder circuit can be implemented with the help of two half adder circuits. The first half adder will be used to add A and B to produce a partial Sum. The second half adder logic can be used to add  $C_{in}$  to the Sum produced by the first half adder to get the final S output. If any of the half adder logic produces a carry, there will be an output carry. Thus,  $C_{out}$  will be an OR function of the half-adder Carry outputs.

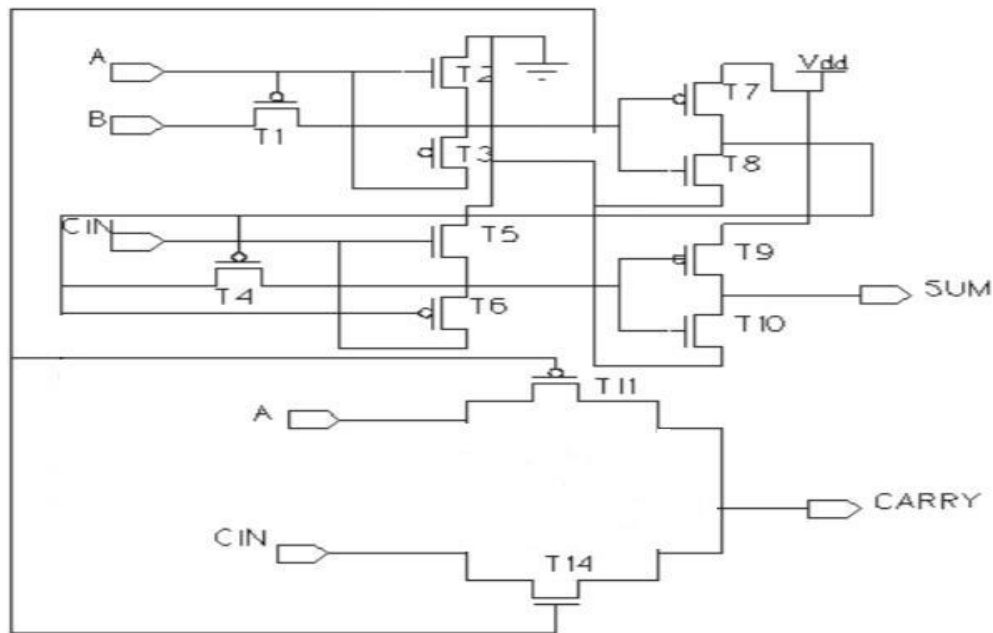


**Fig:3.1: FULL ADDER CMOS DESIGN**

## CHAPTER 4

### PROPOSED METHODOLOGY

#### 4.1 PROPOSED FULL ADDER

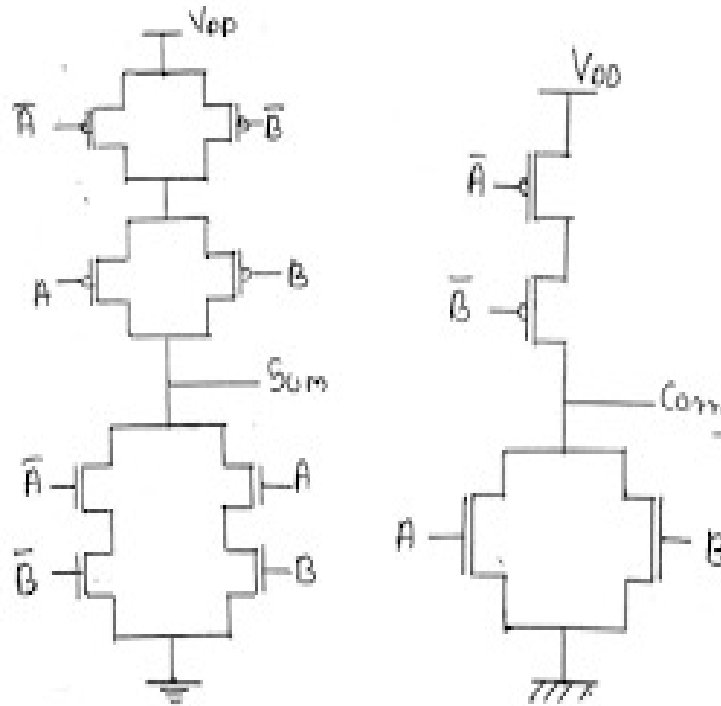


**Fig:4.1: Full Adder**

Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.

12T: It is a schematic of 12T and it contains two XOR/XNOR gate using PMOS and NMOS, four transistors transmission gate multiplexer and three vpulse is added in it and one vdc supply is connected. In this we take 1.2V voltage supply, width and rise time, fall time can be take in different ratio.

## 4.2 HALF ADDER



**Fig:4.2: HALF ADDER CMOS DESIGN**

The half adder adds two single binary digits A and B. It has two outputs, sum (S) and carry (C). The carry signal represents an overflow into the next digit of a multi-digit addition. The value of the sum in decimal system is  $2C + S$ . The simplest half-adder design, pictured on the right, incorporates an XOR gate for S and an AND gate for C. The Boolean logic for the sum (in this case S) will be  $A'B + AB'$  whereas for carry (C) will be  $AB$ . With the addition of an OR gate to combine their carry outputs, two half adders can be combined to make a full adder. The half adder adds two input bits and generates a carry and sum, which are the two outputs of a half adder.

### 4.3 TERNARY MULTIPLIER

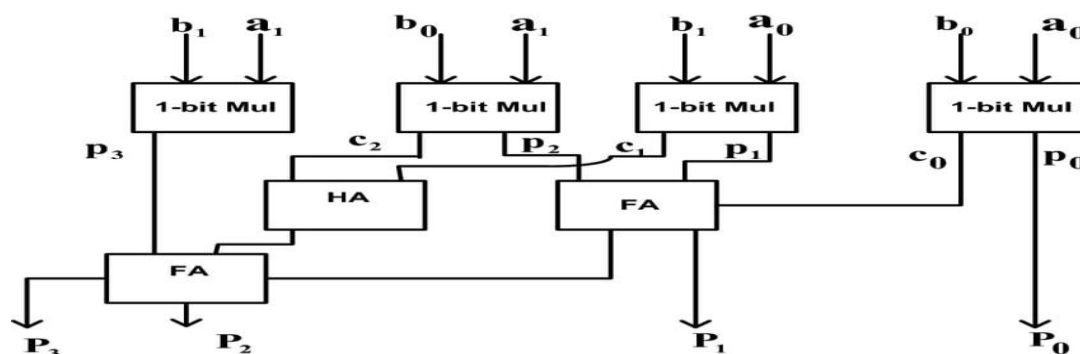
A ternary logic system is a multi valued logic (MVL) system with three logic levels which correspond to logic 0, logic 1 and logic 2. In a balanced or ternary neutral system, the three logic levels are identified as -1, 0 and 1. A ternary logic function  $f(X)$  can be defined as a function mapping  $\{0,1, 2\}$  into  $\{0, 1,2\}$  where  $X$  is given by  $\{X_1, \dots, X_n\}$  and  $X_i, X_j \in \{0,1, 2\}$ . Few basic operations [9] of ternary logic like OR and AND function can be established as below equations (1) and (2) respectively.

$$X_i + X_j = \max \{X_i, X_j\} \quad (1)$$

$$X_i.X_j = \min \{X_i, X_j\} \quad (2)$$

In this work, the three logic levels 0,1,2 correspond to the voltages 0,  $V_{dd}/2$ ,  $V_{dd}$  i.e. 0 V, 4.5 V and 9 V respectively

Inverter is another important ternary logic function. There are three ternary inverters, namely, Standard Ternary Inverter(STI), Negative Ternary Inverter(NTI) and Positive Ternary Inverter(PTI). Table I gives the logic definition of these ternary inverters, where for input  $x$ , the output of the inverters are given as  $y_1, y_2, y_3$ .



**Fig:4.3: TERNARY MULTIPLIER**

#### 4.4 HSPICE SOFTWARE

Hewlett Simulation Program with Integrated Circuit is the full name of HSPICE. Unlike board-level designs made up of individual components, integrated circuits cannot be breadboarded before being manufactured. Furthermore, due to the high cost of photolithographic masks and other manufacturing requirements, it is critical to design the circuit to be as close to perfect as possible before it is designed. Until committing to producing an integrated circuit, the industry-standard method of verifying circuit operation at the transistor level is to simulate the circuit with SPICE.

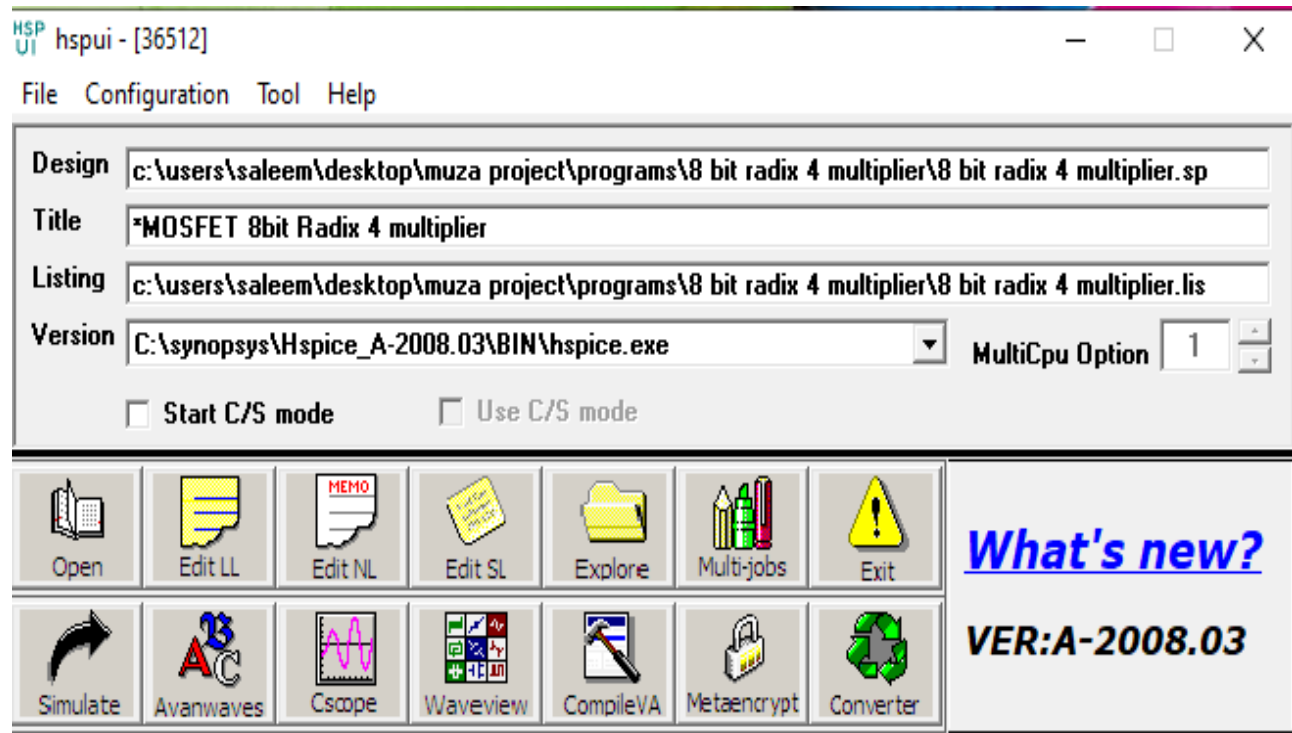
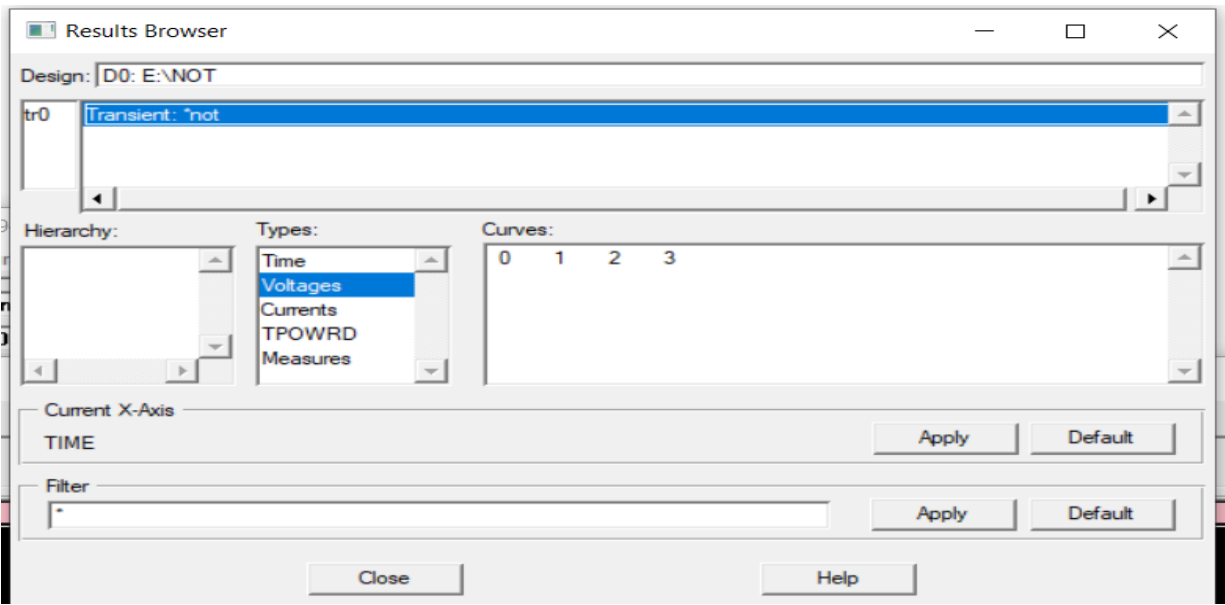
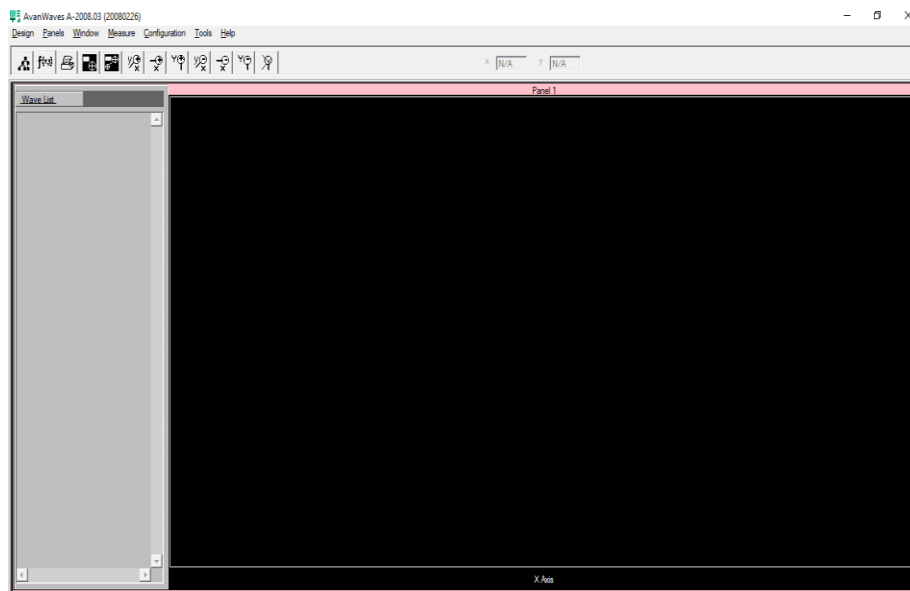


Fig:4.4: HSPICE SOFTWARE



**Fig:4.5: RESULT BROWER SCREEN IN HSPICE SOFTWARE**

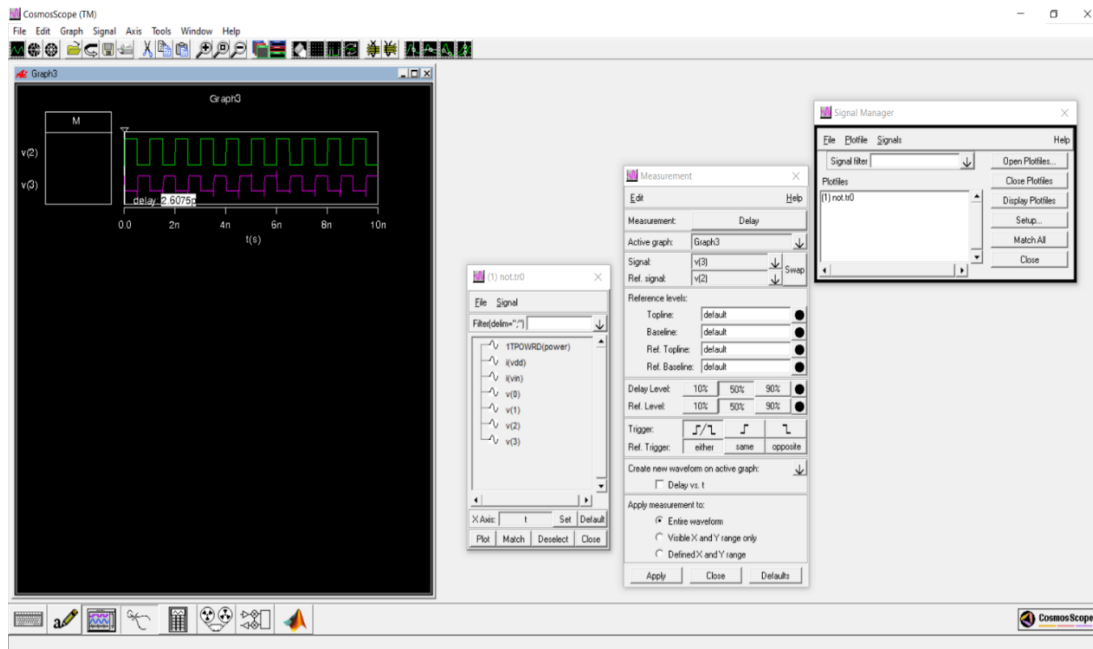
#### 4.4.1 SIMULATED SCREEN



**Fig:4.6: SIMULATED SCREEN**

## 4.5 COSMOSCOPE

Cosmoscope is a software which is used to calculate the delay value.



**Fig:4.7: COSMOSCOPE SOFTWARE**

## 4.6 POWER

Electric power is the rate at which electrical energy is transmitted by an electric circuit per unit of time. It is the pace at which work is completed. Electrical power is the rate at which electrical energy is transmitted by an electric circuit per unit time in terms of an electric circuit.

The unit of the Electric Power = Watt

**Power=V\*I(W)**

$$I=Q/t$$

Where:

W=power in watts

V=potential in volts

I=current in amps

Q=charge in coulombs

t = time in seconds

#### 4.6 DELAY

The time it takes for a signal to reach its destination is referred to as delay. The time it takes for the consequence of a shift in input to be noticeable at the output is known as the delay of a logic gate.

#### 4.7 POWER DELAY PRODUCT

Power Delay Product is the product of the power and delay.

$$\mathbf{PDP=Power * Delay}$$

## CHAPTER 5

### RESULT AND DISCUSSIONS

The existing and proposed architecture are simulated by using AvanWaves simulator. Output of the existing ternary multipliers and proposed single trit multipliers are analysed from the results and it is concluded that the proposed architecture reduces the power consumption compared with the existing system. The Synthesis is done using HSPICE.

#### 5.1 SIMULATION RESULTS OF LOGIC GATES

##### 5.1.1 SIMULATION RESULT OF NOT GATE



**Fig:5.1.1: Simulation Result of NOT GATE**

Figure 5.1.1 shows the simulation output of a HA. The inputs given were a a the output is taken by b.

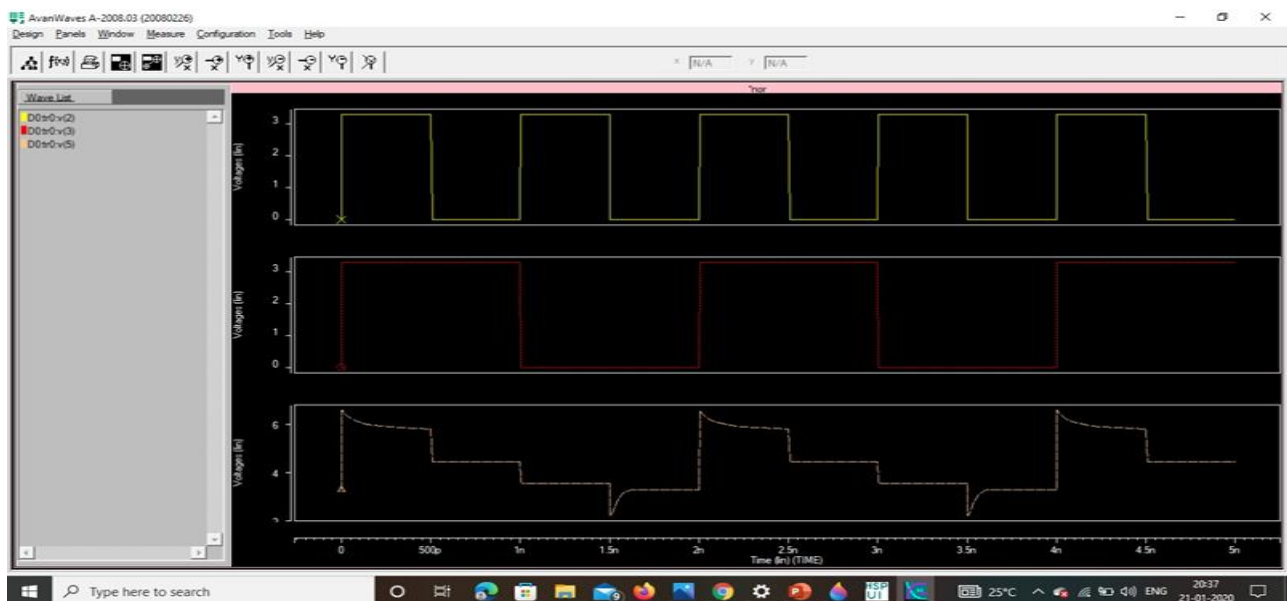
### 5.1.2 SIMULATION RESULT OF NAND GATE



**Fig:5.1.2: Simulation Result of NAND GATE**

Figure 5.1.2 shows the simulation output of a HA. The inputs given were a and b the output is taken by c.

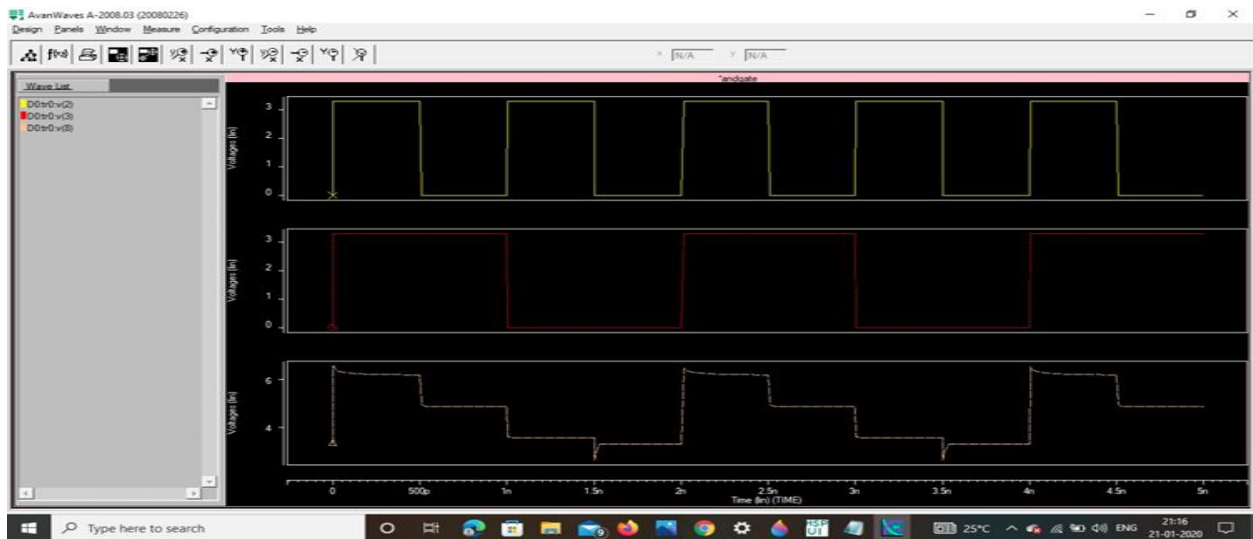
### 5.1.3 SIMULATION RESULT OF NOR GATE



**Fig:5.1.3: Simulation Result of NOR GATE**

Figure 5.1.3 shows the simulation output of a HA. The inputs given were a and b the output is taken by c.

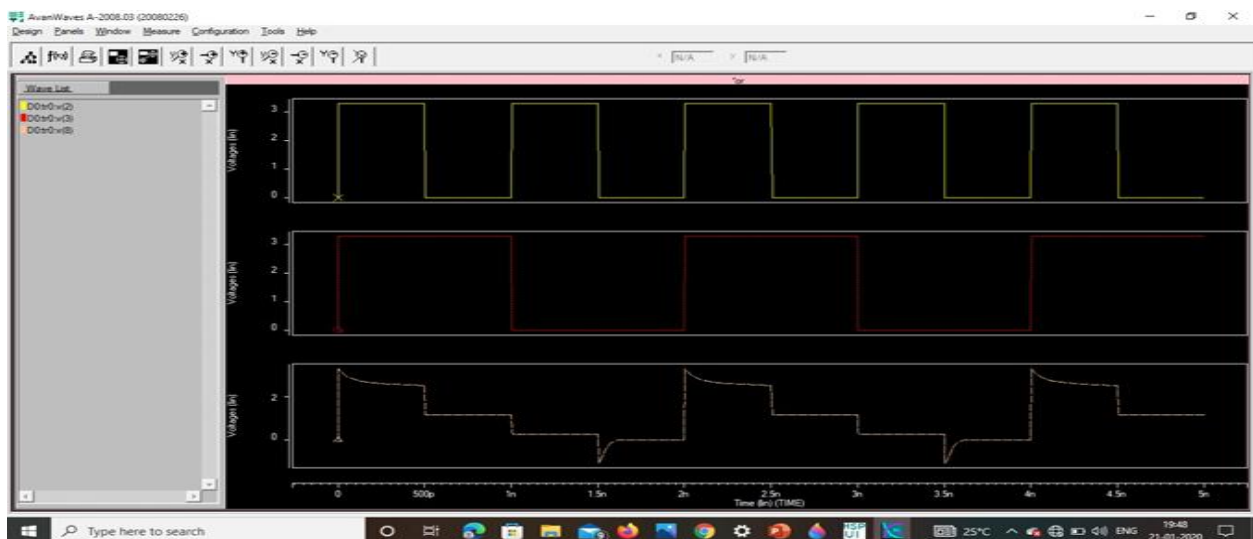
### 5.1.4 SIMULATION RESULT OF AND GATE



**Fig:5.1.4: Simulation Result of AND GATE**

Figure 5.1.4 shows the simulation output of a HA. The inputs given were a and b the output is taken by c.

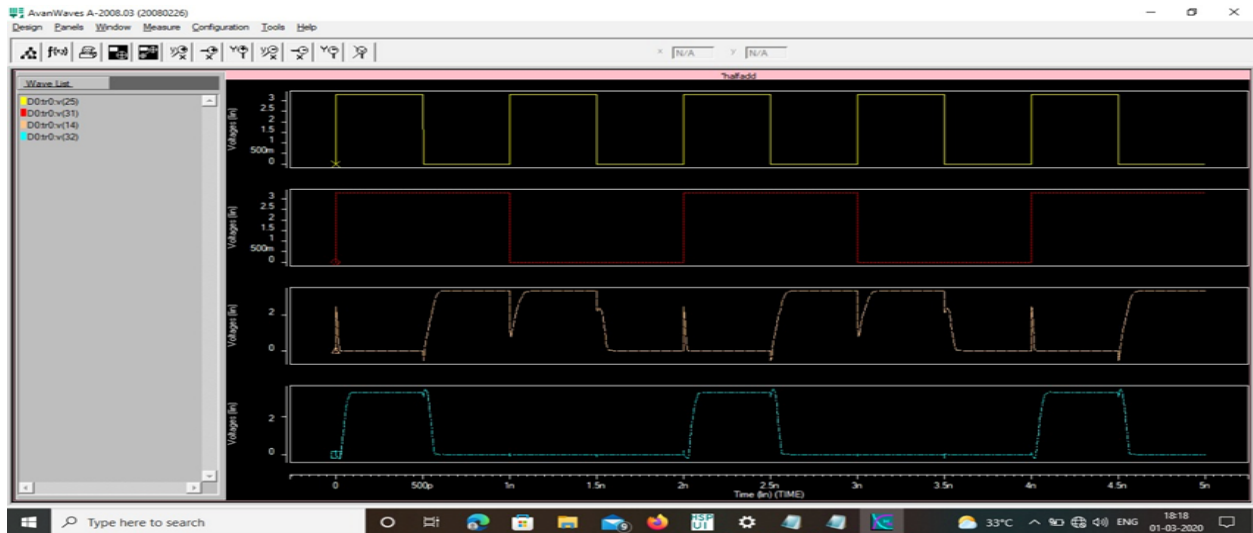
### 5.1.5 SIMULATION RESULT OF OR GATE



**Fig:5.1.5: Simulation Result of OR GATE**

Figure 5.1.5 shows the simulation output of a HA. The inputs given were a and b the output is taken by c.

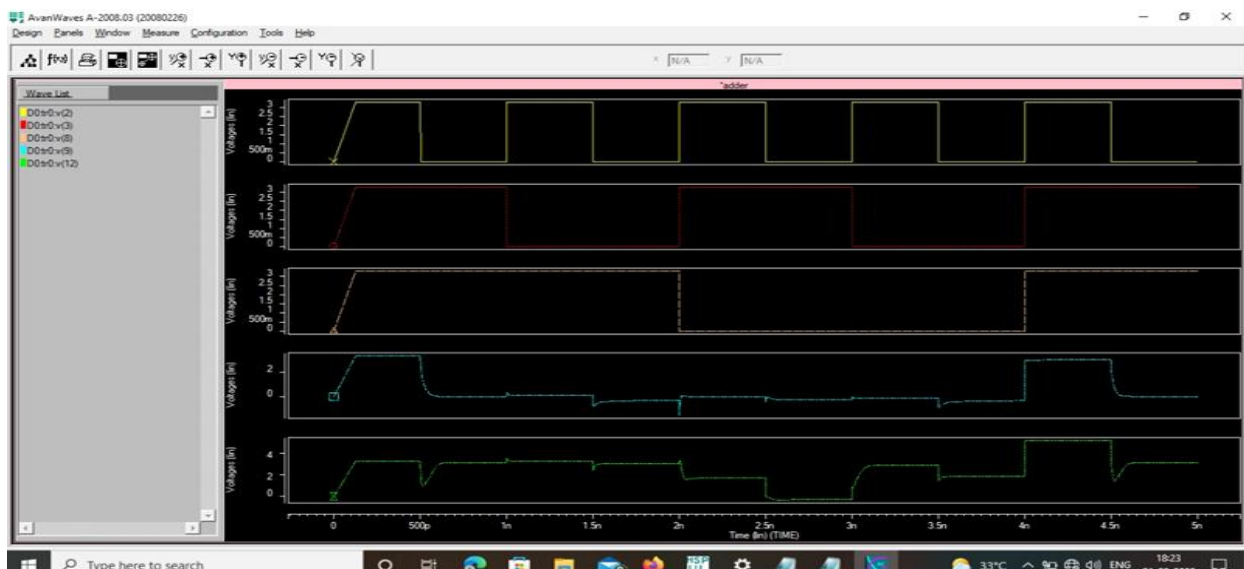
## 5.2 SIMULATION RESULT OF HALF ADDER



**Fig:5.2: Simulation Result Of Half Adder**

Figure 5.2 shows the simulation output of a HA. The inputs given were a and b the output is taken by c and d.

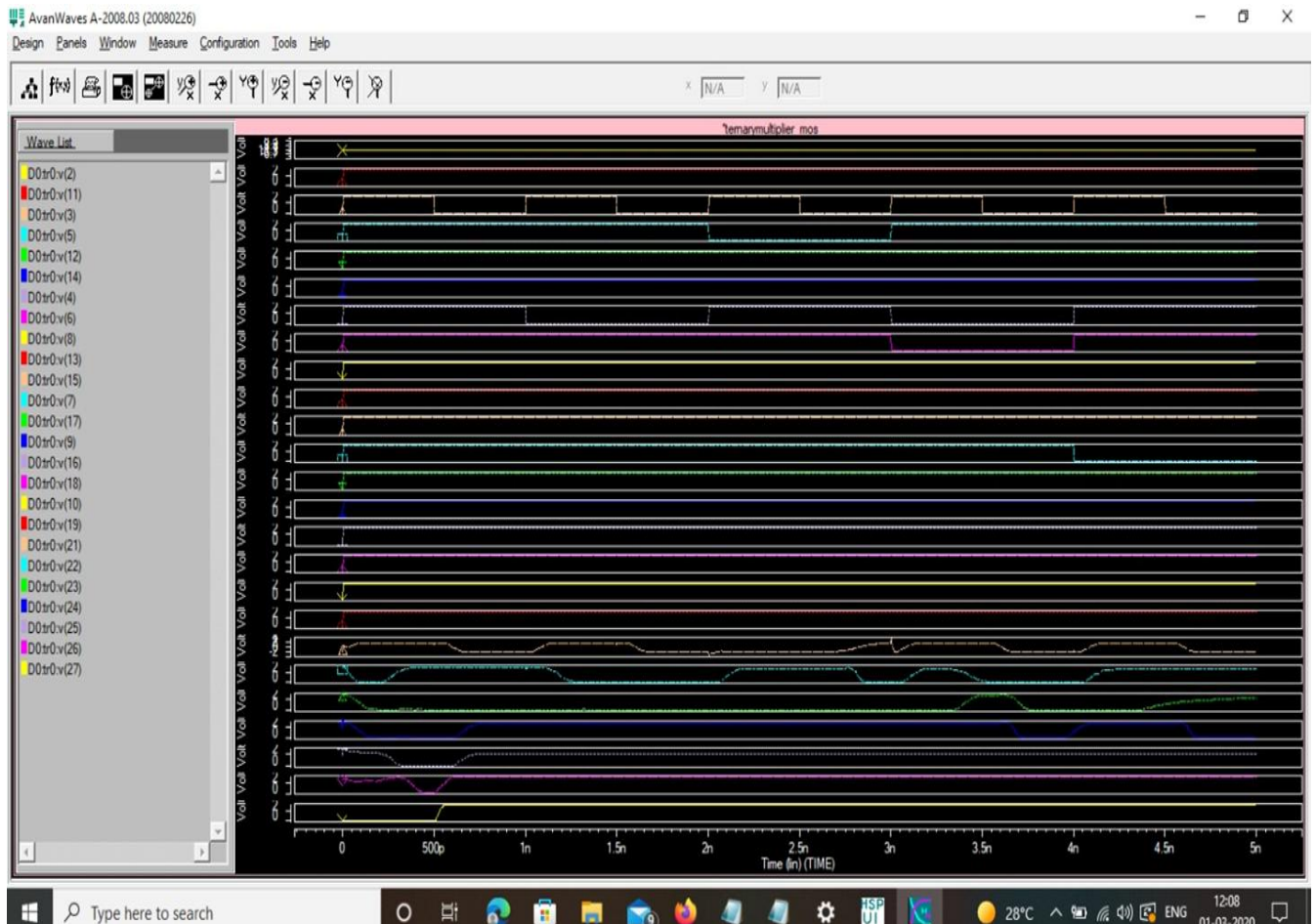
## 5.3 SIMULATION RESULT OF FULL ADDER



**Fig:5.3: Simulation results of Full Adder**

Figure 5.3 shows the simulation output of a Full Adder. The inputs given were a, b, c and the output is taken by sum and carry.

## 5.4 SIMULATION RESULTS FOR TERNARY MULTIPLIER



**Fig:5.4: Simulation results of TERNARY MULTIPLIER**

## 5.5 PERFORMANCE ANALYSIS

The performance analysis of LOGIC GATES, HALF ADDER, FULL ADDER, TERNARY MULTIPLIER in terms of power has been given below.

### 5.5.1 PERFORMANCE ANALYSIS OF LOGIC GATE

**Table:5.1: Design analysis of 130NM NOT GATE**

| <b>NOT GATE<br/>(130NM)<br/>PARAMETERS</b> | <b>VALUES</b>                 |
|--------------------------------------------|-------------------------------|
| POWER                                      | $7.878 * 10^{-3} \text{ W}$   |
| DELAY                                      | $2.8562 * 10^{-12} \text{ s}$ |
| POWER DELAY<br>PRODUCT                     | $22.501 * 10^{-15} \text{ J}$ |

**Table:5.2: Design analysis of 130NM NAND GATE**

| <b>NAND GATE(130NM)<br/>PARAMETERS</b> | <b>VALUES</b>                  |
|----------------------------------------|--------------------------------|
| POWER                                  | $1.091 * 10^{-3} \text{ W}$    |
| DELAY                                  | $493.18 * 10^{-12} \text{ s}$  |
| POWER DELAY<br>PRODUCT                 | $538.059 * 10^{-15} \text{ J}$ |

**Table:5.3: Design analysis of 130NM NOR GATE**

| <b>NOR GATE<br/>(130NM)<br/>PARAMETERS</b> | <b>VALUES</b>                       |
|--------------------------------------------|-------------------------------------|
| POWER                                      | $1.423 \times 10^{-3} \text{ W}$    |
| DELAY                                      | $105.79 \times 10^{-15} \text{ s}$  |
| POWER DELAY<br>PRODUCT                     | $150.539 \times 10^{-18} \text{ J}$ |

**Table:5.4: Design analysis of 130NM AND GATE**

| <b>AND GATE<br/>(32NM)<br/>PARAMETERS</b> | <b>VALUES</b>                       |
|-------------------------------------------|-------------------------------------|
| POWER                                     | $1.413 \times 10^{-3} \text{ W}$    |
| DELAY                                     | $501.37 \times 10^{-15}$            |
| POWER DELAY<br>PRODUCT                    | $708.435 \times 10^{-18} \text{ J}$ |

**Table:5.5: Design analysis of 130NM OR GATE**

| <b>OR GATE<br/>PARAMETERS</b> | <b>VALUES</b>                       |
|-------------------------------|-------------------------------------|
| POWER                         | $1.422 \times 10^{-3} \text{ W}$    |
| DELAY                         | $153.04 \times 10^{-15} \text{ s}$  |
| POWER DELAY<br>PRODUCT        | $217.622 \times 10^{-18} \text{ J}$ |

#### 5.5.2 PERFORMANCE ANALYSIS OF HALF ADDER

**Table:5.6: Design analysis of 130NM HALF ADDER**

| <b>PARAMETERS</b>      | <b>POWER</b>                     | <b>DELAY</b>                       | <b>POWER DELAY<br/>PRODUCT</b>      |
|------------------------|----------------------------------|------------------------------------|-------------------------------------|
| EXISTING HALF<br>ADDER | $2.941 \times 10^{-3} \text{ W}$ | $47.864 \times 10^{-12} \text{ s}$ | $140.768 \times 10^{-15} \text{ J}$ |
| PROPOSED HALF<br>ADDER | $4.074 \times 10^{-3} \text{ W}$ | $2.4372 \times 10^{-12} \text{ s}$ | $9.929 \times 10^{-15} \text{ J}$   |

### 5.5.3 PERFORMANCE ANALYSIS OF FULL ADDER

**Table:5.7: Design analysis of 130NM FULL ADDER**

| PARAMETERS             | POWER                            | DELAY                             | POWER DELAY<br>PRODUCT             |
|------------------------|----------------------------------|-----------------------------------|------------------------------------|
| EXISTING FULL<br>ADDER | $5.983 \times 10^{-3} \text{ W}$ | $8.848 \times 10^{-12} \text{ s}$ | $52.937 \times 10^{-15} \text{ J}$ |
| PROPOSED FULL<br>ADDER | $5.810 \times 10^{-3} \text{ W}$ | $8.844 \times 10^{-12} \text{ s}$ | $51.383 \times 10^{-15} \text{ J}$ |

### 5.5.2 PERFORMANCE ANALYSIS OF TERNARY MULTIPLIER

**TABLE:5.8: Design analysis of 130NM TERNARY MULTIPLIER**

| PARAMETERS                        | POWER                            | DELAY                              | POWER DELAY<br>PRODUCT              |
|-----------------------------------|----------------------------------|------------------------------------|-------------------------------------|
| EXISTING<br>TERNARY<br>MULTIPLIER | $4.719 \times 10^{-3} \text{ W}$ | $372.52 \times 10^{-15} \text{ s}$ | $1757.92 \times 10^{-18} \text{ J}$ |

|                                   |                                  |                                    |                                    |
|-----------------------------------|----------------------------------|------------------------------------|------------------------------------|
| PROPOSED<br>TERNARY<br>MULTIPLIER | $2.885 \times 10^{-3} \text{ W}$ | $250.59 \times 10^{-15} \text{ s}$ | $722.95 \times 10^{-18} \text{ J}$ |
|-----------------------------------|----------------------------------|------------------------------------|------------------------------------|

## CONCLUSION

Multiplication and addition are the important operations in DSP processors. The Half and full adders have been designed using CMOS logic styles to improve their performance in various means like less transistors, low power, minimal delay, reduced area and increased power delay product. This project explores and analyses a novel low power Single-Trit Multiplier and Three-Trit Multiplier design with two modified Full Adder configurations have been presented using a multiplexer based design approach. The simulation are carried out. The power, delay and power delay product analysis are compared. The multiplier circuits were designed and their performance were analysed using HSpice tool. In order to reduce power and delay furthermore, the proposed ternary multiplier is developed and the metrix like power, delay, PDP (Power Delay Product) are analysed and compared with the existing multiplier.

The performance of the proposed designs have tremendous savings in power consumption and the Single-Trit Multiplier design has up to 56% reduction in delay, 99% reduction in power and 99.8% reduction in powerdelayproduct (PDP) when compared to existing Single-TritMultiplier Design. Proposed Three-Trit Multiplier Design has up to 16.67% reduction in delay, 98.26% reduction in power and 98.55% reduction in power-delay product (PDP) when compared to Three-Trit Multiplier Design in existing literature. This proves that the Proposed Ternary Multiplier is more efficient and reliable which can be used for lower power, high speed real time applications.

## REFERENCES

- [1] A. Bachtold, P. Hadley, T. Nakanishi, and C. Dekker, "Logic circuits with carbon nanotube transistors," *Science*, vol. 294, pp. 1317–1320, 11 2001.
- [2] P. C. Balla and A. Antoniou, "Low power dissipation mos ternary logicfamily," no. 2, pp. 217–225.
- [3] Dhande, AP, Ingole, and VT, "Design And Implementation Of 2 Bit Ternary ALU Slice," *Proc. Int. Conf. IEEE-Sci. Electron., Technol. Inf.Telecommun*, pp. 1–11, 2005.
- [4] S. A. ErfanShahrom, "A new low power multiplexer based ternary multiplier using CNTFETs," *AEU - International Journal of Electronics and Communications*, pp. 191–207, 2018.
- [5] Harita Sirugudi, Sharvani Gadgil\*, Chetan Vudadha, "A Novel Low Power Ternary Multiplier Design using CNFETs", 2020.
- [6] A. Heung and H. T. Mouftah, "Depletion/enhancement cmos for a lower power family of three-valued logic circuits," no. 2, pp. 609–616.
- [7] Lin, Sheng, Kim, Y. Bin, Lombardi, and Fabrizio, "CNTFET-based design of ternary logic gates and arithmetic circuits," *IEEE Transactions on Nanotechnology*, vol. 10, no. 2, pp. 217–225, 2011.
- [8] D. Mateo and A. Rubio, "Design and implementation of a 5x5 trits multiplier in a quasi adiabatic ternary cmos logic," *IEEE J. Solid-State Circuits*, vol. 33, no. 7, pp. 1111–1116, 1998.

[9] M. Moaiyeri, a. Doostaregan, and K. Navi, "Design of energy-efficient and robust ternary circuits for nanotechnology," IET Circuits, Devices & Systems, vol. 5, no. 4, p. 285, 2011.

[10] Mudassir. R and Abid. Z, "New parallel multipliers based on low power adders,"2005.

[11] PygastiJuveria and K.Ragini - "Low Power and High Speed Full Adder using new XOR and Xnor Gates" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN : 2278-3075, Volume-8 Issue-8 June-2019.

[12] Radhakrishnan. D., "Low-voltage low-power CMOS full adder," IEEE Proc. Circuits Devices Syst., vol. 148, no. 1, pp. 19–24, Feb. 2001.

[13] D. A. Rich, "A survey of multivalued memories," IEEE Trans. On Comput., vol. 35, no. 2, pp. 99–106, 1986.

[14] Sathiyabama G., "A Survey Of Low Power High Speed Full Adder ",2012.

[15] Sathiyabama G, Malarkkan S., "Dynamic Power Reduction in Carry Save Multipliers using Multi VDD Technique with Single Supply Level Converter",2012.

[16] Senthilpari .C, "A Low Power and High Performance Radix 4 Multiplier Design Using A Modified Pass Transistor Logic Technique "., 2012.

[17] B. Srinivasu and K. Sridharan, "Low-Complexity Multiternary Digit Multiplier Design in CNTFET Technology," IEEE Trans. on Circuits and Systems II, vol. 63, no. 8, 2016.

[18] Supriya S.Saste and Prof.Anil G.Sawant "Design and Implementation of Radix 4 Based Multiplication on FPGA" International Journal of Engineering Research &

[19] S. Tabrizchi, H. Sharifi, F. Sharifi, and K. Navi, "A Novel Design Approach for Ternary Compressor Cells Based on CNTFETs," Circuits, Systems, and Signal Processing, 2015.

[20] Wen. M C, Wang. S J and Lin. Y N," Low power parallel multiplier with column bypassing,"2001.