DESIGN AND ANALYSIS OF HIGH SPEED POWER EFFICIENT MOD-GDI LOGIC BASED 4-BIT VEDIC MULTIPLIER USING NIKHILAM SUTRA

A thesis submitted by:

SAMPA SADHU

Examination Roll No: M6VLS22005

University Registration No: 150133 of 2019-2020

In partial fulfillment of the requirements for the degree
of

MASTER OF VLSI DESIGN AND MICROELECTRONICS TECHNOLOGY

Under The Guidance of

Prof. Sudipta Chattopadhyay

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
JADAVPUR UNIVERSITY
KOLKATA, INDIA
JUNE, 2022

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY
CERTIFICATE

This is to certify that the dissertation entitled **“Design and Analysis of High Speed Power Efficient MOD-GDI Logic Based 4-Bit Vedic Multiplier using Nikhilam Sutra”** submitted by **Sampa Sadhu** (Examination Roll No: **M6VLS22005**; University Registration No: **150133 of 2019-2020**) of Jadavpur University, Kolkata, is a record of bonafide research work under my supervision and be accepted in partial fulfillment of the requirement for the degree of MASTER OF VLSI DESIGN AND MICROELECTRONICS TECHNOLOGY of the institute. The research results represented in this thesis are not included in any other paper submitted for the award of any degree to any other university or institute.

…………………………………………………….

Prof. Sudipta Chattopadhyay
Supervisor
Professor
Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata 700032

…………………………………………. …

Prof. Ananda Shankar Chowdhury
Head of the Department
Department of Electronics &
Telecommunication Engineering
Jadavpur University
Kolkata 700032

..……..…….…………………………………

Prof. Chandan Mazumdar
Dean
Faculty Council of Engineering &
Technology
Jadavpur University
Kolkata 700032

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL*

The foregoing thesis is hereby approved as a creditable study of an engineering subject and presented in a manner satisfactory to warrant acceptance as pre-requisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion or conclusion drawn there in but approve the thesis only for which it is submitted.

Committee on Final Examination
for the Evaluation of the Thesis

…………………………………
Signature of the Examiner

…………………………………
Signature of the Supervisor

*Only in the case the thesis is approved

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC THESIS

I hereby declare that this thesis titled **“Design and Analysis of High Speed Power Efficient MOD-GDI Logic Based 4-Bit Vedic Multiplier using Nikhilam Sutra”** is an original research work done by me under the guidance of my supervisor. This work has not been submitted previously to any other institute.

All the information have been obtained and presented in accordance with Academic rules and Ethical Code of Conduct of the institute.

I also declare that, as required by the rules and conduct, I have cited and referenced the materials that are not original to this work.

Sampa Sadhu

Examination Roll No: M6VLS22005

University Registration No: 150133 of 2019-2020

Thesis Title: Design and Analysis of High Speed Power Efficient MOD-GDI Logic Based 4-Bit Vedic Multiplier using Nikhilam Sutra

Date: …..……………………………………..

Sampa Sadhu

ACKNOWLEDGEMENT

This thesis is the end of my journey in obtaining my master degree. This thesis has been kept on track and been seen through to completion with the support and encouragement of numerous people including my well-wishers, my friends and teachers. At the end of my thesis, I would like to thank all those people who made this thesis possible and an unforgettable experience for me.

First and foremost, I offer my sincerest gratitude to my supervisor, Prof. Sudipta Chattopadhyay, who has supported me throughout my thesis with patience and knowledge. I attribute the level of my master degree to her encouragement and effort and without her, this thesis, too, would not have been completed.

I would also like to thank Prof. Ananda Shankar Chowdhury, Head of the Department, Electronics and Telecommunication Engineering, Jadavpur University for her generous help and suggestions.

I want to thank all the teachers and the staff members of Electronics and Telecommunication Engineering Department of Jadavpur University for their unending support which helped me a great extent for the research work.

Last, but, not the least, I am forever indebted to my husband, my parents and my in- laws for their understandings and encouragements when it was most required.

Place: Kolkata ……………………………………………
Date: Sampa Sadhu

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

- **UT : Urdhva Tiryagbhyam**
- **ALU : Arithmetic and Logic Unit**
- **MOD-GDI : Modified Gate Diffusion Input**
- **MAC : Multiply and Accumulate**
- **FFT : Fast Fourier Transform**
- **VLSI : Very Large Scale Integration**
- **FPGAs : Field Programmable Gate Arrays**
- **LUT : Look Up Table**
- **CPU : Central Processing Unit**
- **DSP : Digital Signal Processors**
- **HDL : Hardware Description Language**
- **VHDL : VHSIC Hardware Description Language**
- **MATLAB : Matrix Laboratory.**

ABSTRACT

With the expansion of digital signal processing applications, the need for high-speed processing is increasing day by day. To achieve this goal, high-speed adders and multipliers (which form the fundamental module) are needed. Moreover, various types of applications require various performance features such as high speed, scalability, low power consumption, reconfiguration, less area, layout regularity, and one or more combinations of these parameters.

The Vedic method of the ancient Indian mathematics based on sixteen Vedic formulae or aphorisms have attracted the attention of researchers in this regard which can offer faster multiplication operations. Urdhva Tiryagbhyam (UT) and Nikhilam are two important Sutras which can be used successfully to design high speed multipliers. Recognizing the correct Vedic formula and executing multiplication operations based on the selected formula can significantly improve the speed of the multiplier.

The main purpose of this research work is to propose a multiplier that produces improved performance in terms of power consumption, delay, and area. Since the performance of the multiplier operations can be significantly influenced by the selected logic technique, the Modified-Gate-Diffusion-Input logic (MOD-GDI logic) is considered here to design various types of 4-bit multipliers. It is basically a low-power design logic technique, which can be used to perform any function, including low transistor count. The Nikhilam Sutra is suitable for larger number of multiplication operations. This sutra can be used for the multiplication very effectively to multiply numbers, around to the power of 10 bases (like 10, 100, 1000, etc). Hence this proposed work focuses on the design based on the realization of a high-speed as well as power-efficient 4-bit Vedic multiplier using Nikhilam Sutra based on the MOD-GDI logic. The design is carried out using T-Spice design tool and its performance has been analyzed in terms of power consumption, speed and area requirement as compared to UT multiplier and conventional Array multiplier. Moreover, the performance of Nikhilam multiplier is also analyzed using MATLAB platform considering the closeness of the multiplicand and the multiplier to the various bases.

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Vedic mathematics is basically a gift from the ancient-sages of India. Vedic mathematics is based on some Sutras and Upa-Sutras and describes the natural way to solve a whole range of mathematical problems. This is a strong field that has already been explored in the recent digitization era. Different techniques have emerged based on sixteen Sutras or formulae that are related to mathematics in terms of arithmetic, geometry, algebra, etc. Vedic mathematics converts difficult calculations into uncomplicated calculations. Vedic mathematics is a familiar technique for low power consumption, area, and delay. Hence, it can be applied in various branches of engineering such as digital signal processing and computing.

1.1.1 ORIGIN OF ANCIENT VEDIC MATHEMATICS

The word "Vedic" is derived from the word "Veda", which means the warehouse of all knowledge. The word “Veda” also refers to the holy ancient Hindu-literature which is separated into four parts. The Vedas are considered to be the oldest form of human written record. The Vedas were initially approved orally from the previous generation to the next. Later they were copied into Sanskrit language. A study of all the Vedic scripts found in different parts of India did not find the least difference between them. The Vedas contain information on many subjects from mathematics, architecture, religion, astronomy, medicine, etc. “Jagadguru Shankaracharya Bharati Krishna Tirthaji Maharaja” [1986], an Indian scholar, performed a detailed investigation has demonstrated 16 Sutras (formulae) and 13 Upa-sutras (sub-formulae) for performing basic and complex mathematical operations.

1.1.2 REQUIREMENT OF FASTER AND POWER EFFICIENT PROCESSOR

Although the growth of the electronics market has pushed the VLSI sector towards extremely high integration densities and system-on-chip (SOC), there have been critical concerns about the rapid increase power-consumption and area. High power usage raises the chip's temperature profile, affecting the system's overall performance. Furthermore, the rapid expansion of laptops and portable personal communication systems necessitates requires long battery life in moderate performance levels. Moreover, technical advancement in the digitalization era demands the development of better and faster processors, as well as processors that are power efficient. The widespread use of arithmetic units has resulted from recent technological advancements in the field of signal processing. Multipliers are the main fundamental block in high-speed arithmetic logic units, digital signal processing units, and multiplier & accumulator units. Multiplication is the most significant operation in computer arithmetic. As multiplication is crucial for mathematical processes such as convolution, correlation, image processing, frequency analysis, etc, therefore the process time for multiplication operations must be reduced as far as possible.

1.2 MOTIVATION

The above background demanded for intensive research into integrated circuit design with high speed, low power and low area. The number of operations processed per second increases when speed is improved. With the increased circuit complexity, the density of components on a chip also increases, resulting in increased power dissipation. As a result, in recent times with the successful growth of high-speed processors, the prime concern of the researchers is focused on reducing the power consumption while simultaneously increasing the speed of the design.

This has motivated to implement a high speed as well as power-efficient multiplier for different real life applications. Since the applications of “Vedic mathematics” in various mathematical operations have already been studied in various research works, one of the famous Sutras called Nikhilam Sutra has been explored in this research work for the purpose of the design of the faster multiplier. The concept of MOD-GDI technique as an effective way to significantly reduce power consumption and has been an important research topic till now. Various research papers have been proposed recently using several domains of MOD-GDI logic in Vedic multiplication algorithm. This research focuses on the creation of 4-bit multipliers based on “Modified-Gate-Diffusion-Input” (MOD-GDI) logic “Nikhilam” sutra of Vedic multiplication.

1.3 OBJECTIVE

Multiplication acts as the heart of the processing of today's real-time signal-processing algorithms. This is the most essential arithmetic unit the DSP processor. Multiplier is the primary determinant of power-consumption, processor speed, as well as processor packaging. Latency and degree of efficiency are two significant characteristics associated with the multiplication of processor applications. The actual delay in calculating the function is defined as Latency. The degree of efficiency is a measurement of how many calculations can be completed over a period of time. Hence all the processors require high-speed multipliers.

Low-power, high-speed multiplier circuits are required for the construction of power-efficient, high-speed processors. High-speed of the multiplier can be obtained by minimizing the architecture's complexity. Various multiplication algorithms, such as Array, Wallace Tree, and Booth, etc. are always being researched in order to get the best possible results in this field. Due to its simplicity and speed, Vedic multiplication has been determined to be the most appropriate process. The performance of the multiplier circuit can be further optimized in terms of power efficiency by using MOD-GDI logic. This has been verified in several research works.

The main objective of this research work is to develop faster and more power-efficient multiplier (4×4) based on Nikhilam sutra from Vedic mathematics using MOD-GDI logic. The logical elements and layout space are reduced while implementing this Vedic multiplier. As a result, power consumption is reduced. The detailed performance analysis of the designed multiplier with respect to the conventional Array multiplier, and UT multiplier has been another objective of this research.

1.4 METHODOLOGY

The ancient "Vedic Mathematics" system is consisting of a set of 16 formulas (major Sutras) and their 13 Upa-sutras (Sub-sutras), which can be implemented for solving mathematical problems. Only two Sutras are used in this work for the multiplication operation, out of several alternatives. These two Sutras, which are used in this research work, are Urdhva Tiryagbhyam (UT) sutra and Nikhilam Navatascaramam Dasatah (Nikhilam) Sutra.

"Urdhva Tiryagbhyam" (UT) is one of the Vedic Sutras, which literally means vertically and crosswise, and is used to perform a multiplication. Unlike traditional multiplication schemes, the partial products are gathered at each step in this multiplication procedure. As a result, the partial product accumulation delay can be minimized and subsequently, the multiplier's speed can be improved.

The Nikhilam algorithm signifies "All from 9 and last from 10". It's also known as the basic multiplication method. This algorithm can be used for any multiplication operation. The working principle of Nikhilam sutra for multiplication operation is based on the remainder. The calculation of the remainder is based on the difference between the nearest base value and the given number. The selection of the nearest base value is considered in terms of the powers of 10 for decimal numbers. Nikhilam formula is suitable for larger number of multiplication operations. This Nikhilam sutra has been attempted on the proposed 4-bit multiplier.

In this research work, the Modified-Gate-Diffusion-Input logic (MOD-GDI logic) style is considered to design various types of 4-bit multipliers. It is basically a low-power design logic technique, which can be used to perform any function, including low transistor count. With this technology, total power consumption, propagation delay, required transistors for design are much more reduced as compared with CMOS and GDI techniques.

The proposed architectures are implemented using TSPICE software (S-Edit Win64 16.30.20150626.04:37:24). In addition, the performance analysis of Nikhilam Sutra based multiplier is carried out using MATLAB R2015a (8.5.0.197613) 64-bit, on Windows 7 64-bit OS, 4 GB RAM, Intel Core i5-3317U CPU, processor running at 1.70 GHz.

1.5 CONTRIBUTION

The design of a high-speed and power-efficient 4-bit multiplier circuit is proposed in this thesis. The following are the major contributions to this work:

- A 4-bit high-speed and low-power multiplier has been designed by combining the "Nikhilam sutra" of Vedic multiplication and a novel MOD-GDI logic style and the design is implemented on T-SPICE.

- The basic gates based on MOD-GDI logic required for implementation of aforementioned algorithm is implemented using TSPICE platform. After that, all of these fundamental gates are combined to form the desired structure.
- To compare the performance of above proposed 4-bit Nikhilam multiplier, 4-bit UT multiplier and 4-bit array multiplier have been developed with MOD-GDI logic using T-SPICE.
- The efficiency of the implemented design has been analyzed by comparing its performance with various multipliers like 4-bit UT multiplier and 4-bit array multiplier, in terms of power consumptions, and speed.
- The performance of Nikhilam multiplier is also analyzed w.r.t different nearest base by changing the value of multiplier, multiplicand and nearest base through MATLAB.

1.6 OUTLINE OF THE THESIS

The organization of this thesis is as follows:

- Chapter 2 presents the detailed literature survey of related designs already proposed previously.
- Chapter 3 explains the concept of "Vedic sutra" and its basic algorithms for multiplication. This chapter also describes MOD-GDI logic techniques in detail.
- Chapter 4 discusses the proposed design of a 4-bit Nikhilam multiplier of Vedic mathematics based on MOD-GDI logic. This is followed by the design of MOD-GDI logic based 4-bit Array multiplier and 4-bit UT multiplier.
- Chapter 5 summarizes the simulation results and their critical analysis.
- Finally, the thesis is concluded in Chapter 6. Some of the related works that can be performed in the future is also considered.

CHAPTER 2

LITERATURE REVIEW

2.1 OVERVIEW

This chapter provides a thorough review of the literature based on the performance of various multipliers in the domain of multiplication. Creating high-performance VLSI structures necessitates the development of efficient arithmetic units that optimize decision-making parameters like power consumption and speed.

Multipliers perform a crucial role in DSP processors and microprocessors. Area is another important parameter to consider while designing various circuits. There are three essential parameters in VLSI design are speed, power, and area. With the advance of technology, numerous researchers have experimented with multipliers that can achieve any of the above-mentioned purposes.

2.2 REVIEW OF THE RELATED WORK

With the advancement of technology, low power consumption has become a major concern for researchers. Other design characteristics such as speed, area, power, and Power Delay Product are properly considered in addition to this criterion to improve a device's overall performance. A multiplier is a functional block that is widely employed in "digital signal processors" in various units such as "Arithmetic and Logic Units”, "Multiply and Accumulate", and various operations such as "Fast Fourier Transform", "Discrete Cosine Transform", and so on. In order to achieve the following goals, multiplier design is always being improved. To fulfill the demand for high-speed, power-efficient multipliers, a variety of multiplication algorithms were investigated.

Vedic Mathematics based on Sixteen Simple Mathematical sutra from the “Vedas” was offered by Jagadguru Swami “Sri Bharti Krishna Tirthaji Maharaja” [1965]. His disciples had been waiting for it for a long period of time. It is the product of the intuitional representation of fundamental mathematical truths, rather than being pragmatically conceptualized and worked out as is the case with other scientific works. This ancient Indian Vedic knowledge should be complete and perfect in every way, able to release the most light on any topic that an aspirant seeker of knowledge could wish to be educated on.

The design of multipliers based on Vedic mathematics sutra such as Urdhava Tiryabhyam (i.e. vertically and cross wise) and Nikhilam sutras were proposed by “Moumita Ghosh” [2007], “Saurabh Sunil” [2011] and “Abhishek Gupta” et al. [2012], “Kokila” et al. [2012], and “Pohokar” et al. [2015]. These sutras were widely utilized in ancient India to make decimal numerical issues easier to solve. The binary multiplication is proposed based on these sutras. “Jin Hyukkim” et al. developed an FPGA implementation of these sutras (2000).

“Morgenshtein” et al. [2002] proposed GDI logic as an alternative to CMOS logic. It is a low-power design technique that allows logic functions to be realized with fewer transistors. Several arithmetic and logic circuits have been designed using this logic. GDI logic based adders were implemented by “Lee” [2007]; “Moradi” et al. [2009]; “Dan Wang” et al. [2009]; “Uma” et al. [2012]; “Shrivas” et al. [2012]; “Dhar” [2014]; “Archana” et al. [2014]; “Morgenshtein” et al. [2014]; “Shinde” et al. [2014] “Foroutan” et al [2014]; and “Soundharya”et al.[2015]. GDI logic based subtractors were implemented by “Dhar” et al [2014] and “Singh” [2014]), multiplier (implemented by “Gupta” et al [2013] and “Reddy” et al [2014]. Moreover, GDI logic based divider was implemented by “Saberkari” et al [2009], comparators were implemented by “Khurana” et al [2013]; “Sharma” et al. [2014] and “Shekhawat” et al [2014], Arithmetic and Logic Unit was implemented by “Dubey” et al. [2014], flip flops were implemented by “Morgenshtein” et al [2004]; “Fisher” et al [2009]; “Swami” et al [2011]; “Abiri” et al [2014] “Dhar” [2014]. Moreover, memory unit was implemented by “Magesh Kannan” et al [2015]), clock generators were implemented by “Hari” et al. [2011].

“Gurumurthy” et al. [2010] proposed the "Array of Array" multiplier that is basically an imitative of "Braun-Array Multiplier". Braun-array multiplier is much more suitable for VLSI design due to its less space difficulty although it shows more time complexity. Since time complexity of tree multipliers is O (log n), are less suitable for VLSI realization. Moreover, they are less regular, they require more routing length, which leads to degradation of performance as well as complexity of higher space. Hence this configuration is not suitable for VLSI applications. The inherent ability of this proposed multiplier is mainly used to reduce the complexity of both time and space, including intermediate relative performance. This is the main advantage of this proposed multiplier. In this proposed designed a 16×16 unsigned 'Array of Array' multiplier circuit was implemented using hierarchical structure, and optimized using Vedic multiplication formula based on "Urdhva Triyagbhyam" and “Karatsuba-Ofman” algorithm. The proposed algorithm is suitable for mathematical coprocessors in the computer field. The Algorithm is applied to SPARTAN-3E-FPGA (Field-Programmable-Gate-Array). The proposed multiplier performance shows a significant reduction in average power consumption and time-delay as compared with encoded radix-4 Booth-multiplier.

“Vaidya” et al. [2010] suggested that a simple processor central processing unit (CPU) spends enough processing time to perform mathematical operations, especially multiplication. Multiplication is a basic mathematical operation and requires a considerable amount of more hardware resources and processing time instead of addition and subtraction. In this work, a comparative study of various multipliers for low power requirements and high speeds has been conducted. This paper provides information on the “Urdhva Tiryakbhyam” algorithm of ancient Vedic mathematics used for multiplication to improve the speed as well as the area of the multiplier. Vedic mathematics suggests another formula for multiplication of larger numbers named as the "Nikhilam Sutra". By using this Nikhilam sutra, the speed of the multiplier can be increased by reducing the number of repetitions.

“Sumit Vaidhya” et al. [2010] conducted a comparative analysis of various multipliers for low power consumption and high speed. This paper describes the "Urdhva Tiryakbhyam" algorithm, which is used for multiplication to increase the speed and area parameters of multipliers in Ancient Indian Vedic Mathematics. Vedic Mathematics proposed another formula for large-number multiplication, named as, 'Nikhilam Sutra,' which can speed up the multiplier by minimizing the number of iterations. The time delays of the Nikhilam Sutra Vedic multiplier were compared to those of conventional multipliers in this article. “Chitralekha Mehera” [2012] investigated the conceptions of the Vedic Mathematics Sutra "Nikhilam Navatascaramam Dasatah" in computing mathematical problems such as addition, subtraction, multiplication, division, 10’s complement, and multiplication tables, using numerous examples.

“Pradhan” et al. [2011] proposed an architecture for a 16×16 Vedic multiplier module based on the “Urdhva Tiryagbhyam” Sutra and the "Nikhilam Sutra" approach, which uses a Carry save adder. The Urdhva Tiryagbhyam Sutra 16×16 Vedic multiplier module consists of one 16-bit carry-save adder, four 8-bit Vedic multiplier modules, and two 17-bit complete adder sections. In the multiplier design, the carry-save adder speeds up the addition of partial products. Using Xilinx ISE 10.1 software, the 16×16 Vedic multiplier was programmed in VHDL, synthesized, and simulated. The Spartan 2 FPGA device XC2S30-5pq208 was used to implement this multiplier. When the magnitude of both operands is more than half of their maximum values, the 16x16 Vedic multiplier using "Nikhilam Sutra" was found to be faster than the 16x16 Vedic multiplier using "Urdhva Tiryakbhyam" Sutra in terms of speed. This method works effectively for multiplying numbers that have more than 16 bits. Delay, number of slices, number of 4 input LUTs, and number of bonded inputs and outputs are the parameters used to evaluate performance.

Based on the Urdhava -Triyakbhayam sutra of Vedic mathematics, “Jaina” et al. [2011] proposed a multiply and accumulate (MAC) unit with a multiplier. VHDL was used to code, and XILINX: VERTEX2P: XC2VP2:-7 and XILINX: SPARTAN3:XC3S50:-4 were used to synthesize the code. In terms of speed, the proposed MAC unit based on Vedic mathematics is very good. It is structured in a regular and parallel manner. As a result, implementing it on a silicon chip may be straightforward. A carry-save adder was used in the multiplier, which results in high-speed operation and a low transistor count. When compared to the proposed multiplier in the literature, the combinational delay outperformed the multiplier. In addition, the area was compared in terms of the number of slices and the number of LUTs with four inputs.

“Charishma” et al. [2012] implemented the design of a high-speed multiplier based on ancient Vedic mathematics techniques, which have been modified to get better performance. Vedic mathematics is basically an ancient method of mathematics which has various exclusive techniques for mathematical calculation based on the sixteen sutras or formulae. This work has demonstrated the effectiveness of Urdhva-Triyakbhyam, Vedic method for multiplication is a various process of multiplication. It enables parallel-generation of intermediary products, eliminates unnecessary quality steps including zero, and scales to higher-bit levels using Karatsuba-algorithms compatible with different data types. Urdhva -Tiryakbhyam Sutra is very efficient formula, providing less time delay for multiplication operation of all types of smaller and larger numbers. Further, the coding of Urdhva Tiryakbhyam Sutra was done on Verilog HDL for (32x32) bits multiplication and implemented on FPGA with Xilinx-Synthesis tool. The output has been shown on LCD display of Spartan-3E kit. The synthesis results show that the computation-time for multiplication operation (32-bits) is 31.5260 ns.

According to Ramachandran et al. [2012], the multipliers are the essential and central components of all “Digital Signal Processors” (DSP), and the speed of the DSPs is significantly controlled by the speed of their multipliers. With significant arithmetic computations, multiplication is the most fundamental process. Latency and throughput are two essential characteristics linked with multiplication algorithms used in DSP applications. The "actual delay of computing a function" is referred to as latency. The term "throughput" refers to the number of computations that may be completed in a given amount of time. Most DSP algorithms' execution times are determined by their multipliers, necessitating the use of a high-speed multiplier. For small inputs, Urdhva Tiryakbhyam Sutra is faster, but Nikhilam Sutra is faster for larger inputs. A revolutionary Integrated Vedic multiplier structure was proposed here, which automatically selects the suitable multiplication sutra based on the inputs. As a result, the suggested integrated Vedic multiplier architecture selects whichever sutra is faster based on inputs. The simulation results showed that Urdhva performs better for small inputs, but Nikhilam performs faster for large inputs (with 64-bit multiplicands, more than twice as much).

“Sree Nivas” et.al [2012] developed the 2's complement block and encoding techniques used in the design to implement the Nikhilam sutra multiplier. By dropping zeros from the least significant and conducting bit reduction, “Vedavathi” el al. [2012] introduced an iterative multiplication technique based on the Nikhilam sutra. In terms of delay and power, “Harish Kumar” [2013] compared the Vedic multipliers based on the Urdhava and Nikhilam sutra to the Array multiplier. Floating-point multipliers were developed by “Vishal Singla” [2013] utilizing the divide and conquer approach and pipelining.

“Somani” et al. [2012] introduced the multiplication function in mathematical operations. A Central Processing Unit (CPU) spends enough processing time to perform mathematical operations. The multiplication operation requires considerably more processing time and hardware resources instead of addition and subtraction. Digital-signal-processor (DSP) is a technology that is ubiquitous in engineering. High-speed multiplication in DSP is very important for Fourier transform, convolution, digital filter etc. This research paper presented a comparative study of Vedic mathematics based on Urdhva-Tiryakbhyam sutra with the conventional multiplier as well as hierarchical array-of-array multiplier in terms of various parameters such as power, area, speed, delay etc. This paper provides information on the "Urdhva-Tiryakbhyam" algorithm of Vedic multiplier used for multiplication to improve the power, area, speed, and delay of the multipliers.

“Virendra Magar” [2013] proposed the basic block of MAC based on ancient Vedic mathematics on high-speed and low time delay multiplier. This type of multiplier enables parallel creation of partial-products and eliminates unnecessary steps of the multiplication. Multiplier structure was based on creating all the partial-products and their sum in one step. Chipscope VIO was used by the user to provide random input of the desired values, on the basis of which the proposed Vedic multiplication was performed. The proposed algorithm was simulated using VHDL, which is a hardware description language with very high speed integrated circuit. The propagation time was found to be quite less for this proposed architecture. Xilinx Chipscope VIO-Generator was permitted to provide runtime input. The Xilinx Chipscope tool was used to test FPGA inside results during logic in FPGA. This circuit used the “Xilinx Spartan 3 Family FPGA” development board.

“Premananda” et al. [2013] presented a multiplier (8-bit) using a Vedic mathematics based on Urdhva Tiryagbhyam sutra to create partial products. The addition of partial products was realized using Vedic multiplier based on the carry-skip technique. The proposed multiplier was realized using a 4-bit multiplier and modified ripple-carry-adder. In this proposed design, the logic levels count was reduced, thus the logic delay was also reduced.

“Goyal” et al. [2013] introduced that multiplication operation is a key factor for mathematical operations, it is performed in a number of DSP applications. Since the execution time of multipliers are more, so a fast multiplier is needed to save time for implementation. In this paper the multiplication operation was described using the ancient Vedic mathematics technique, which has a unique method of multiplication calculation based on the 16 sutra. In this paper the Nikhilam, Urdhva Tiryakbhyam, and Karatsuba sutra based techniques were described and obtained the performance analysis of these techniques. The simulation was done using Modelsim tool and the obtained results are compared based on the delay of multiplication operations. This paper also described the speed of different multiplier techniques by comparing them with each other.

“Magar” [2013] introduced that the speed of the multiplier is very essential for any digital signal processor (DSP). “Vedic Mathematics” is the ancient technique of Indian mathematics which has an exclusive technique of mathematical calculation using the 16 sutra. This proposed work represented the efficiency of the Vedanta method of the Urdhva -Triyakbhyam sutra for multiplication operation, which distinguished the real process of multiplication operation. The high-speed and low integrated delay co-efficient was distinguished which is the basic MAC block based on ancient “Vedic Mathematics”. It allowed parallel production of partial-products and removed all unnecessary multiplication steps. Multiplier structure was based on all the partial-products and their aggregates in a single step.

The performance of any processor, according to “Chaudhary” [2013], is determined by its power and delay. To get an effective processor, the power and delay should be reduced. A multiplier is the most common architecture in processors. An effective processor can be generated if the multiplier's power and delay are minimized. The most common multiplier architectures are Array and Vedic multipliers. There are two different kinds of multiplication techniques utilized in Vedic multipliers, based on Urdhva Triyagbhyam and Nikhilam sutras. The purpose of this study was to compare various architectures in order to determine which one is superior for multiplication in terms of power and delay. The Verilog language was used to develop the architectures, and the simulation tool is Xilinx 10.1i ISE.

“Pradhan” et al. [2013] described a multiplier design based on Nikhilam Sutra of Vedic mathematics. By extracting the complement of the given integer from the closer base, this technique executed a multiplication action. The larger operations were broken down into smaller ones. The multiplier's speed was improved by using a carry-save adder. With Xilinx ISE 10.1 software, the designed circuit was synthesized and simulated for the Spartan 2 FPGA device XC2S30-5pq208. The synthesis findings were used to determine the parameters in the output, such as delay and device usage. The speed and device consumption were tested using an antique multiplier architecture that provides speed increase based on the evaluation results.

“Itawadiya” et al. [2013] demonstrated that DSP based on Vedic mathematics takes less time to process than MATLAB operations. The authors used the Vedic Urdhava-Triyagbhayam

Multiplication Sutra in MATLAB to implement high-speed DSP actions on two finite-length sequences. The Urdhaya-Triyagbhyam technique and the Conventional technique were used to compare response time for Convolution, Auto-Correlation, Circular-Convolution, and Cross-Correlation. According to the results, DSP based on Vedic mathematics operations save processing time by 40-60% when compared to MATLAB's built-in function.

"Kayal" et al. [2013] demonstrated a high-speed, low-power digital multiplier by combining Vedic multiplication algorithms with McCMOS technology, a very efficient leakage control approach. An 8 bit Vedic multiplier was constructed utilizing multiple channel CMOS (McCMOS) technology with node technologies of 130 nm; 90 nm; 65 nm; and 45 nm. Comparative simulation results revealing the circuit's performance were also shown. "Khaldoon" et al. [2014] proposed a new, effective reduction strategy for implementing the tree multipliers on "Field Programmable Gate Arrays" (FPGAs) in a technique that is more suited to the lookup tables (i.e. LUTs) structure in FPGAs. The method relies on a library of m: n counters. The goal of this strategy is to reduce the count of reduction steps as much as possible in order to achieve a maximum reduction ratio, which minimizes area and time.

The need for high-speed multipliers grew with the increased demand for high-speed processors. In most of the rapid processing systems, a multiplier is one of the hardware components. This is a significant source of power dissipation as well as a high delay block. Multiplication, rather than addition and subtraction, demands significantly more hardware resources and processing time in a normal CPU. In this research, the authors created a high-speed multiplier inspired by Indian ancient Vedic mathematics "Banupriya" et al. [2014] based on Urdhva and Nikhilam Sutras. In comparison to other existing multiplication algorithms, this approach required fewer calculations as the number of multiplier bits rises. The Vedic multiplier was compared to existing methods such as the booth and array multiplier. This technique was able to reduce the overall time delay of the multiplier.

"Panwit Tuwanuti" et al. [2014] proposed the mathematic parallel process idea by using Vedic Multiplier techniques. To implement the Multi core processing with MPICH2 (MPI-protocol), the "Urdhva Tiryakbhyam Sutra" and "Nikhilam Sutra" were used. "Urdhva Tiyakbhyam" sutra of Vedic mathematics was used to break long digits to sub-block for distribution to another core. "Nikhilam Sutra" was used to decrease value to improve some of calculation efficiently.

"Jain" et al. [2014] used Vedic Mathematics based on Nikhilam sutra and Parvarty Sutra to construct optimal binary division architecture, as well as introduced their use for the calculation of deconvolution. The proposed division algorithm was written in Verilog and then synthesized and simulated with the Xilinx ISE design suite 14.2. According to the researchers, simulated results for their suggested Vedic divider circuit demonstrated a 19% reduction in delay over the conventional technique.

The layout for the sutra "Ekanyunena Purvena" was proposed by "Angshnuman Khan" et al. [2015]. This sutra literally means "One less than the previous" or "One less than the one before". It is a Upa-Sutra of the "Nikhilam Navatashcaramam Dashatah" from Vedic mathematics. This Sutra is primarily used to multiply a number by 9, 99, 999, and so on. Because one number should have an array of nine, this architecture can be considered as a generalized approach.

“Tadas” [2015] proposed the performance of the Vedic mathematics based on Nikhilam, Dhwajank, and paravartya sutras for 64-bit binary number division and devised a division algorithm. The study concluded that these algorithms needed the minimum procedures. As a result, their method found to be cost-effective and efficient, and simple to design using VHDL.

“Ram” et al. [2016] proposed a VLSI structure for a high-speed 32-bit multiplier based on the Nikhilam Sutra and the Urdhva-Tiryagbhyam Sutra from Vedic mathematics. For implementation and synthesis, Xilinx Software was utilized. The multipliers were compared in terms of delay, power, the number of LUTs, and the number of flip flops. The study concluded that the Nikhilam Sutra of Vedic multiplier had a shorter delay and consumed less memory than Urdhava-Triyakbhyam multiplier. They also demonstrated that by combining the Urdhva-tiryagbhyam Vedic multiplier with BEC-1 adders, latency could be minimized as the number of bits grows.

"Pinninti Kishore" et al. [2017] used the MOD-GDI technique to propose a high-speed optimized 4-bit array multiplier. The operation of MOD-GDI and its basic operations were described first, followed by the design of fundamental logic blocks that would be used in the suggested multiplier design. Mentor Graphics Pyxis schematic tools were used for the simulation scientific analysis of the 4 x 4 bit Array multiplier. As a result, it was found that the MOD- GDI array multiplier used extremely little power. In addition, the number of transistors required was minimized, and the propagation delay was reduced. The performance of the proposed multiplier by using MOD-GDI technique was found to be better than array multiplier based on CMOS and GDI technique in terms of power dissipation, propagation delay and transistor count.

“Bilanchi” [2020] introduced a Vedic multiplier based on the Urdhava Triyakbhayam Sutra that utilized 4:2 compressor blocks and compared it to other 4:2 compressor blocks architecture Vedic multipliers found in the literature in terms of Look Up Tables and Propagation Delay. The suggested multiplier was designed in Simulink and then automatically programmed in VHDL using MATLAB's HDL coder, before being synthesized and implemented on a Xilinx Artix 7 FPGA using the Xilinx VIVADO tool package. According to the author, the proposed multiplier performs well and is a better solution for model-based designs.

2.3 OUTCOME OF THE REVIEW

Power consumption and delays have been identified as important components of design in VLSI technology based on the preceding review. The "Vedic mathematics," which serves as the foundation for a variety of mathematical operations, has a wide range of applications in a variety of domains. Because of its simplicity, speed, and compact size, the architecture has gained its popularity to a great extent. These benefits have been well demonstrated in the preceding review. Furthermore, by using MOD-GDI logic instead of CMOS technology, power consumption can be lowered significantly. In the aforementioned studies, this concept has been experimentally proven. In context to Vedic multiplier, Urdhava Tiryakbhyam Sutra based design is suitable for any kind of input. In several applications of digital signal processing, this method is commonly utilized. However, the above literature review shows that only a few articles are based on the Nikhilam sutra. Because of its limits on the use of inputs, this sutra is not that effective. As a result, the Nikhilam sutra-based based multiplier has not been used widely. Vedic multipliers are usually used to minimize computing complexity. It is found that the the application of various Vedic sutras has already been carried out. A comparison of conventional binary multipliers with Vedic multipliers has been done in some of the researches. The Urdhava multiplier's design process delay increases as the number of bits increases. As a result, this method works similar to an array multiplier in the limiting case. The multiplication of the complement of the numbers has also been performed in the Nikhilam sutra. The resulting value is found to be less if the number is nearer to the base value, and the number of bits will be reduced. If it is not closed, then, the complement value will be greater and the bits will not be reduced. Because of its input range limitations, the Nikhilam Sutra is not utilized in many of the practical applications. The main advantage of this method is that it is efficient if the numbers are nearer to the base value. The remainders are obtained by determining the 2's complement of multiplier and multiplicand. The multiplier is utilized to multiply the multiplicand's and multiplier's 2's complement. Hence it is evident that there is an immense scope of research in the area of design of power efficient Vedic multiplier based on Nikhilam Sutra which is the main motivation of this research.

CHAPTER 3

THEORETICAL BACKGROUND OF VEDIC MULTIPLIER AND MOD-GDI LOGIC

3.1 OVERVIEW

Speed and power are essential requirements for the design of compact and ultra-fast electronic devices. Multiplication is used in digital signal processing, arithmetic units in microprocessors, image processing applications, and all of which contribute significantly to overall power consumption. The three most significant parameters in VLSI design are power, speed, and area. With technological advancements, many researchers have attempted multipliers that provide either of the above- mentioned targets. This chapter provides detailed theoretical concepts on the two primary components of this thesis work, namely the "Vedic multiplier" technique and the "MOD-GDI" Logic. Because of its simplicity, power consumption, area, and speed, multipliers based on "Vedic mathematics sutras" (“Somani” et al. [2012]) have found major applications. The "Nikhilam sutra" of "Vedic mathematics" serves as a framework for this thesis work. This study demonstrated the algorithm of this sutra. Furthermore, the MOD-GDI logic technique and its functions are introduced in (“Pinninti Kishore” et al. [2017]).

3.2 VEDIC MULTIPLIER

Vedic mathematics is basically a Vedic method of the ancient Indian mathematics, derived from Vedic formulas, recently rediscovered. The system of this operation is based on sixteen Vedic formulas or aphorisms, which describe the natural way to solve a whole range of mathematical problems. Recognizing the correct Vedic formula and executing multiplication operations based on the selected formula can significantly improve the speed of the multiplier (“Charishma” et al. [2012]). This proposed work focuses on the design based on the realization of a high-speed as well as power-efficient Vedic multiplier using Nikhilum formula based on the MOD-GDI logic technique.

3.2.1 HISTORY OF ANCIENT INDIAN “VEDIC MATHEMATICS”

The derivation of the word "Vedic" is from the word "Veda", which means the warehouse of all knowledge. The word “Veda” also refers to the holy ancient Hindu-literature which is separated into four parts. The Vedas are considered to be the oldest form of human written record. The Vedas were initially approved orally from the previous generation to the next. Later they were copied into Sanskrit language. A study of all the Vedic scripts found in different parts of India did not find the least difference between them. The Vedas contain information on many subjects from mathematics, architecture, religion, astronomy, medicine, etc. “Jagadguru Shankaracharya Bharati Krishna Tirthaji Maharaja” [1986], an Indian scholar, performed a detailed investigation has demonstrated 16 sutras (formulae) and 13 up-sutras (sub-formulae) for performing basic and complex mathematical operations. Vedic-mathematics is not only a mathematical marvel, but also it is logical. It explains several mathematical terms including arithmetic, geometry, and factorization of quadratic equations, integration, trigonometry, and even calculus. Because of its unique features, "Vedic mathematics" (“Somani” et al. [2012]) has reached the top of the list of research topics around the world. This study provides a brief description of the 16 sutras and 13 sub-sutras of Vedic mathematics, (“Manikadan S.K” [2017]) which is described below.

3.2.1.1 THE LIST OF SUTRAS AND THEIR UPA-SUTRAS OF VEDIC MATHEMATICS

There are 16 Sutras and 13 Upa-Sutras of ancient Indian Vedic Mathematics [“Manikadan S.K” (2017)] has been listed below.

- “Ekadhikina purvena” : The Upa-Sutra of this Sutra is “Anurupyena”
- “Ekanyunea purvena” : No Upa-Sutra for this Sutra
- “Anurupye shunyamanyat” : The Upa-Sutra of this Sutra is “Yavadunam Tavadunam”
- “Gunakasamuchyah” : No Upa-Sutra for this Sutra
- “Gunitasamuchya” : No Upa-Sutra for this Sutra
- “Paraavartya yojayet” : The Upa-Sutra of this Sutra is “Kevalaih Saptakam Gunyat”
- “Nikhilam” : The Upa-Sutra of this Sutra is “Sisyate Sesasamjnah”
- “Puranapuranabhyam” : The Upa-Sutra of this Sutra is “Antyayordasake pi”
- “Shesanyankena charamena” : The Upa-Sutra of this Sutra is “Vilokanam”
- “Sankalana vyavakalanabhyam” : The Upa-Sutra of this Sutra is “Yavadunam Tavadunikryta”
- “Urdhva Tiryagbhyam” : The Upa-Sutra of this Sutra is “Adyamadyenantyamantyena”
- “Shunyam saamyasamuccaye” : The Upa-Sutra of this Sutra is “Vestanam”
- “Yaavadunam” : The Upa-Sutra of this Sutra is “Samuccayagunitha”
- “Vyashtisamanstih” : The Upa-Sutra of this Sutra is “Lopanasthapanabhyam”

- “Soopantyadvayamantyam” : The Upa-Sutra of this Sutra is “Gunitasamuccayah Samuccayagunitah”
- “Chalana kalanabhyam” : The Upa-Sutra of this Sutra is “Antyayoreva”

3.2.1.2 THE DESCRIPTION OF 16 SUTRAS OF VEDIC MATHEMATICS

The description of the 16 Sutras of Vedic Mathematics (“Manikadan S.K” [2017]) are given as follows:

- “Ekadhikina purvena”: The operation of this sutra is defined as, by ‘1’ more than previous.
- “Ekanyunea purvena”: The operation of this sutra is defined as, By ‘1’ lesser than the previous.
- “Anurupye shunyamanyat”: The operation of this sutra is defined as, If one in ratio, other will be zero.
- “Gunakasamuchyah”: The operation of this sutra is defined as, Factors of the sum = sum of all factors.
- “Gunitasamuchya”: The operation of this sutra is defined as, Product of the Sum (POS) = Sum of Product (SOP).
- “Paraavartya yojayet”: The operation of this sutra is defined as, transpose cum adjust.
- “Nikhilam”: The operation of this sutra is defined as, all from ‘9’ and last from ‘10’.
- “Puranapuranabhyam”: The operation of this sutra is defined as, By non-completion or completion.
- “Shesanyankena charamena”: The operation of this sutra is defined as, Remainders by last digit
- “Sankalana vyavakalanabhyam”: The operation of this sutra is defined as, By sum and subtraction
- “Urdhva Tiryagbhyam”: The operation of this sutra is defined as, vertically-crosswise
- “Shunyam saamyasamuccaye”: The operation of this sutra is defined as, if sum is same then that the sum is ‘0’
- “Yaavadunam”: The operation of this sutra is defined as, whatever be the extension of the deficiency

- “Vyashtisamanstih”: The operation of this sutra is defined as, partial and overall
- “Soopantyadvayamantyam”: The operation of this sutra is defined as, ultimate & double the penultimate
- “Chalana kalanabhyam”: The operation of this sutra is defined as, differences & similarities

3.2.2 VEDIC MULTIPLICATION

Among the four basic mathematical operations (addition; subtraction; multiplication; and division), multiplication is operation is considered the most difficult operation. Researchers are working to improve the power efficiency and speed of multipliers, which are fundamental building blocks and significant among other basic mathematical operational blocks. The use of "Vedic-Principles" in multiplication allows this complex operation to be performed in a simplified and quick manner. In order to achieve high performance, "Vedic multiplication" significantly reduces area as well as power when compared to “Conventional multiplication” methods."Vedic multiplication" is commonly used for decimal number multiplications. However, it is now also used in binary multiplication systems to achieve compatibility with the available hardware design. Among all the above- mentioned sutras, multiplication based on the "Urdhva Triyagbham” and “Nikhilam” sutra is a simple and effective method (“Banupriya” et al. [2014]).

Urdhava- Tiryakbhyam is a standard Vedic multiplication technique. It is appropriate for all input ranges. The Nikhilam Sutra is a special case of Vedic multiplication. It is primarily used to solve complex calculations using simple techniques that can be performed mentally. Typically, the sutras are described in terms of decimal numbers. The same logic for binary numbers based on digital hardware is used for binary implementation. In this thesis the proposed multiplier has been designed based on Nikhilam sutra of Vedic mathematics.

3.2.3 “NIKHILAM NAVATASHCARAMAM DASHATAH” SUTRA

The Nikhilam algorithm signifies “All from 9 and last from 10”. It's also known as the basic multiplication method. This algorithm can be used for any multiplication operation. Usually, it works with operands of a larger bit size. It converts multiplication with large operands to multiplication and addition operations with smaller operands (“Sumit Vaidhya” et al. [2010]).

The working principle of Nikhilam-sutra is based on the remainder. The remainder is calculated by subtracting the given number from the nearest base value. For decimal numbers, the base value is chosen in powers of ‘10’. Multiplication based on the Nikhilam-sutra is suitable for multiplying larger number of features. The larger the actual number, the lesser the difficulty of the multiplication. This sutra can be used very effectively to multiply numbers, around to the power of ‘10’ bases (like 10, 100, 1000, etc).

The procedure for selecting the nearest radix or base and its impact on the overall complexity of the multiplication process are discussed. In the proposed algorithm, the process of selecting the nearest base is dependent on the number of digits in both multiplicand and multiplier. The base is usually taken as a power of ‘10’ in the decimal algorithm.

Figure 3.1 depicts the principle of Vedic signed multiplier based on the Nikhilam Sutra. The architecture is made up of steps such as multiplicand complement, multiplier complement, multiplication, addition, and concatenation. A 4-bit radix is selected for both the 4-bit multiplier (n) and the 4-bit multiplicand (m) in 4-bit signed Vedic multiplication operation. The block of 2’s complement determines the 4-bit multiplier’s complement (n’) by calculating the difference between the 4-bit base and the 4-bit multiplier (n). Similarly, by calculating the difference between the 4-bit base and the 4-bit multiplicand (m), the 2’s complement block determines the 4-bit complement of multiplicand (m’).

The product is split into two parts: the left-hand side part of the result (LHS) and the right-hand side part of the result (RHS). The 4-bit multiplier block determines the product of the 4-bit complement of multiplicand (m’) and the 4-bit complement of multiplier (n’) to obtain the 8-bit product. The RHS part of the 4-bit multiplier output is indicates the least significant 4-bit of this product. The carry-save adder (4-bit) block combines the most significant four bits of multiplier output with four bits of multiplicand and four bits of multiplier. The output of carry save adder determines the LHS of the product. Finally, the concatenation process has been done for LHS part and RHS part to obtain the final 8-bits product.

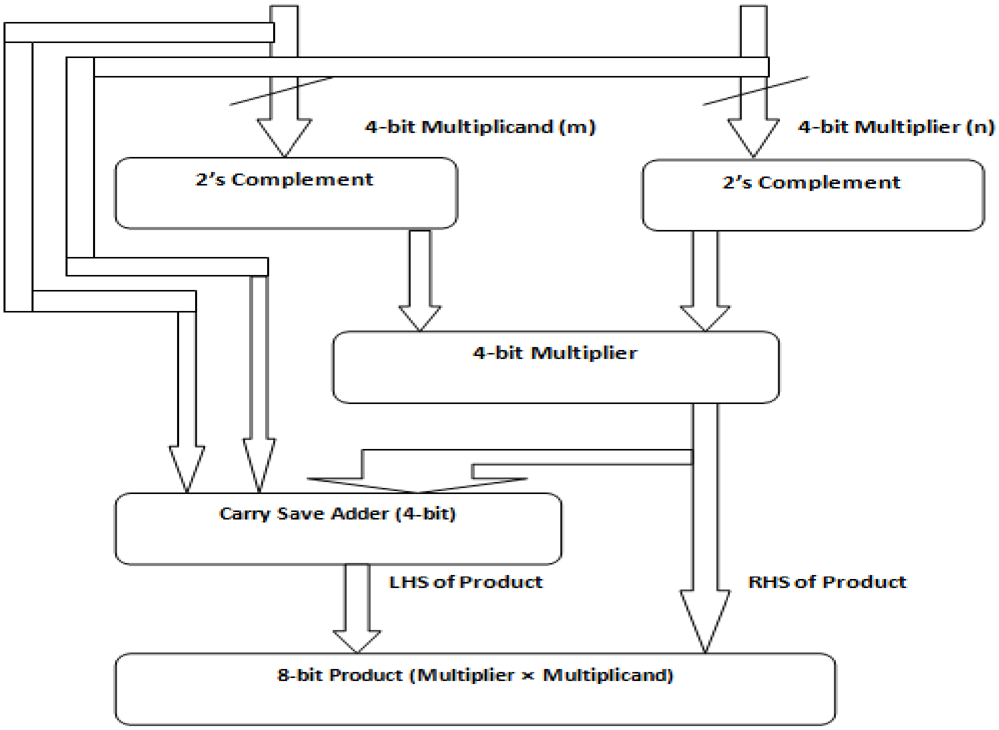


Figure: 3.1 Architecture of 4-bit Vedic multiplier based on Nikhilam sutra

3.2.4 Algorithm of Vedic Multiplier Based On Nikhilam Sutra

The algorithm of the Nikhilam sutra has been discussed in the following tables. There are three algorithms for Nikhilam sutra explained in details in tabular form with proper example for better understanding.

Table 3.1 Both Numbers are Greater Than Base

Steps of Algorithm	**Both Numbers are Greater Than Base**	
Step 1.	X= Multiplier, Y= Multiplicand	X=104, Y=102
Step 2.	Select Nearest base value 'B' (i.e. Power of '10')	B=100
Step 3.	Difference (M)= Multiplier(X)-Nearest base(B)	M=104-100=4
Step 4.	Difference (N)= Multiplicand(Y)-Nearest base(B)	N=102-100=2
Step 5.	Product of two differences (C) = M*N	C=M*N=4*2=8
Step 6.	D= X+N= Y+M	D= X+N=104+2=106
Step 7.	Result= B*D+C	Result= B*D+C=100*106+8 =10608

Table 3.2 Both Numbers are Lesser than Base

Steps of Algorithm	**Both Numbers are Less Than Base**	
Step 1.	X= Multiplier, Y= Multiplicand	X=98, Y=96
Step 2.	Select Nearest base value 'B' (i.e. Power of '10')	B=100
Step 3.	Difference (M)= Nearest base(B)- Multiplier(X)	M=100-98=2
Step 4.	Difference (N)= Nearest base(B)- Multiplicand(Y)	N=100-96=4
Step 5.	Product of two differences (C) = M*N	C=M*N=2*4=8
Step 6.	D= X-N= Y-M	D= X-N=98-4=94
Step 7.	Result= B*D+C	Result= B*D+C=100*94+8 =9408

Table 3.3 Multiplier is Greater Than Base and Multiplicand is Lesser than Base

Steps of Algorithm	Multiplier is Greater Than Base and Multiplicand is Less Than Base	
Step 1.	X= Multiplier, Y= Multiplicand	X=103, Y=99
Step 2.	Select Nearest base value ‘B’ (i.e. Power of ‘10’)	B=100
Step 3.	Difference (M)= Multiplier(X)-Nearest base(B)	M=103-100=3
Step 4.	Difference (N)= Nearest base(B)- Multiplicand(Y)	N=100-99=1
Step 5.	Product of two differences (C) = B-(M*N)	C=B-(M*N)=100-(3*1)=97
Step 6.	D= (X-N)-1= (Y+M)-1	D= (X-N)-1=(103-1)-1=101
Step 7.	Result= B*D+C	Result= B*D+C=100*101+97 =10197

3.3 “MODIFIED-GATE-DIFFUSION-INPUT” (MOD-GDI) LOGIC

In high-performance applications, power dissipation has become the most important factor to be considered. Optimization techniques for fundamental logic gates are essential for improving the performance of a wide range of low-power and high-performance devices. Modified-gate-diffusion-input (Mod-GDI) logic ("Pinninti Kishore" et al. [2017]) can be used to overcome these limitations. This method reduces the power consumption, area, and delay of digital systems. Figure 3.2 depicts the fundamental Mod-GDI logic technique. However, unlike the basic GDI cell, the Modified-GDI (Mod-GDI)cell has a low-voltage terminal S_p that is connected to a high constant voltage (i.e. power supply) and a high-voltage terminal S_n that is connected to a low constant voltage (i.e. GND). The utilization of these terminals ensures that the Mod-GDI cell can be constructed with all current CMOS process technologies.

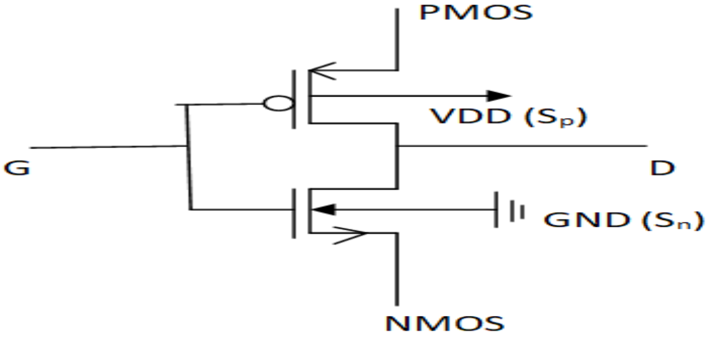


Figure 3.2 Basic Structure of Modified-Gate-Diffusion-Input (MOD-GDI) Logic

The bulk node of the PMOS transistor is connected to a high fixed voltage known as the supply voltage (VDD), while the bulk of the NMOS transistor is connected to a low voltage level known as GND in the Mod-GDI cell. As a result, the suggested Mod-GDI cell is perfectly compatible for implementation in a standard CMOS fabrication process, with the majority of PMOS transistors connected to VDD and the majority of NMOS transistors connected to GND. Mod-GDI is used in Figure 3.2 to improve logic level swings and static power properties while also allowing a simple top-down design method with a tiny cell library. The Mod-GDI cell is constructed with standard four-terminal NMOS and PMOS transistors and can be easily implemented in any standard CMOS technology. The Mod-GDI cell can also be used in non-standard technologies such as twin-well CMOS technology, Silicon on Sapphire (SOS) technology, and Silicon on Insulator (SOI) technology.

3.3.1 CHARACTERISTICS OF MOD-GDI LOGIC

The characteristics of MOD-GDI Technique ("Pankaj Verma" el al. [2013]) are described as follows:

- The Mod-GDI logic is a small area and low-power replacement for existing techniques that can be implemented in any current CMOS transistor fabrication technology. Mod-GDI is suitable for the design of high-speed, low-power circuits design with fewer transistors, static power characteristics, and even improving swing degradation and also permitting top-down design with a small cell library. Mod-GDI is suitable for the implementation of a wide variety of logic circuits using various transistor technologies.

- The implementation of logic functions in Silicon-on-Sapphire (SOS) or Silicon-on-Insulator (SOI) fabrication technologies reduces the number of wires used for interconnecting significantly; in these methodologies, floating bulk transistors are widely utilized for logical-circuit implementations for GDI logic technique and other existing logic designs. Floating bulk transistors do not require the VDD and GND wires to be connected to the bulks of the transistor. As a result, when implementing a logic function with Mod-GDI cells and SOS or SOI transistors, VDD and GND interconnect wires are not needed since the Mod-GDI cell only requires VDD and GND to supply the bulks. In contrast, the majority of previous design techniques required VDD and GND to provide the power supply to the circuits.

3.3.2 ADVANTAGES OF MOD-GDI LOGIC

The main advantages of MOD-GDI Logic ("Pankaj Verma" el al. [2013]) are as follows:

- Because of this simple arrangement, the fabrication of GDI cells is suitable for the standard (nano-scale) CMOS fabrication process. While many leakage components can be identified in advanced sub-micron technologies, the two most important leakage currents are sub-threshold leakage and gate leakage.

- When compared to a static CMOS gate, this exceptional Mod-GDI cell arrangement significantly reduces both gate leakage and sub-threshold. These methods can be used as a framework for implementing efficient synthesizers, such as short-circuit current reduction, repeaters, and other area-saving designs.

3.4 DISCUSSION

In this chapter, the theoretical background of Vedic mathematics has been discussed based on different sutras and up-sutras. Specially, Nikhilam sutra, which has been utilized in the proposed multiplier, has been discussed briefly. The algorithm of Nikhilam multiplier for various multiplication operations also explained in this section. Moreover, the MOD-GDI Logic, which is used successfully in this work, had also been explained in this chapter. The working principle, characteristics and advantages of MOD-GDI Logic are summarized in brief.

CHAPTER 4

PROPOSED DESIGN OF 4-BIT NIKHILAM VEDIC MULTIPLIER USING MODIFIED-GATE-DIFFUSION INPUT (MOD-GDI) LOGIC

4.1OVERVIEW

The necessity for high speed, low power multiplier combined with minimal layout architecture can be considered as one of the emerging areas of research of recent past. By combining the Vedic multiplication algorithm with a low-power design technique known as MOD-GDI logic, this thesis aims for providing a high-speed and power-efficient digital multiplier. Ancient Indian mathematics derived from Vedic sutras is known as Vedic mathematics and two popular Sutras in Vedic mathematics are "Urdhva-Tiryakbyham" and "Nikhilam Navatascaramam Dasatah."

From the fundamental GDI technology, the MOD-GDI technique is derived. This is a brand-new technique for designing low-power digital systems. The MOD-GDI technique is primarily used to decrease total propagation delay, power consumption, as well as transistor count. All of these criteria can be attained using the GDI technology, but there are some drawbacks to it, such as the need for additional circuitry to provide inputs to the GDI circuits and the difficulty of the fabrication process.

All designs shown in this chapter (existing and proposed) are developed based on aforementioned Vedic Sutras using Modified-Gate-Diffusion Input (MOD-GDI) approach, by exploiting a CMOS level design tool TANNER SPICE (S-Edit Win64 16.30.20150626. 04:37:24).

4.2 ARCHITECTURE OF THE EXISTING 4-BIT ARRAY MULTIPLIER

The CMOS level design of a low-power, high-speed 4×4 bit array multiplier is described below.

4.2.1 THE FUNDAMENTAL BLOCKS OF ARRAY MULTIPLIER

The fundamental blocks required for the design of 4-bit Array multiplier have been developed based on MOD-GDI logic using design tool T-SPICE are illustrated below:

4.2.1.1 AND Gate with MOD-GDI logic

The AND gate is designed with MOD-GDI logic using T-SPICE has been shown below:

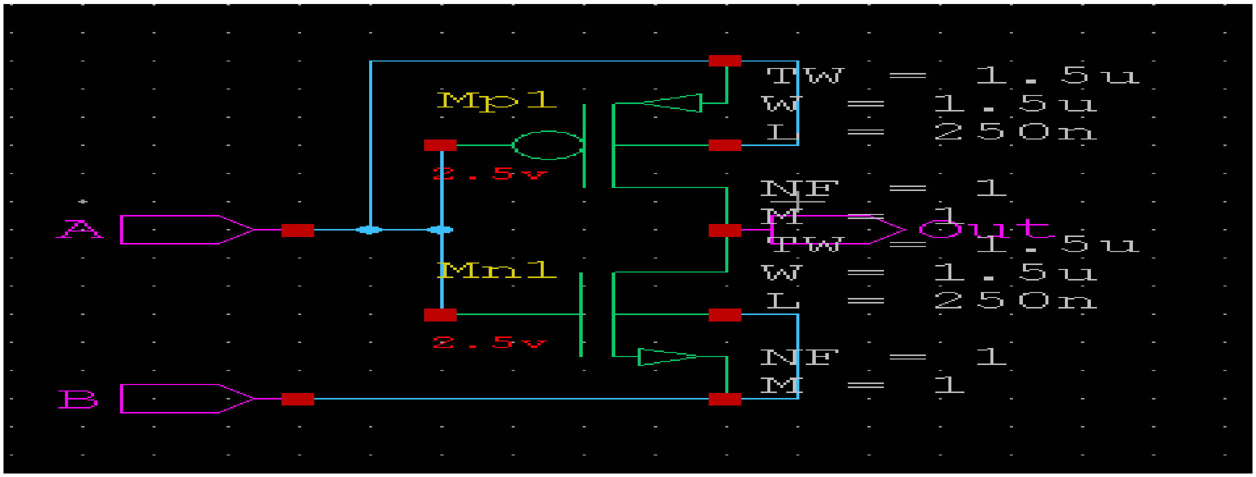


Figure 4.1 Schematic Diagram of AND Gate with MOD-GDI logic

4.2.1.2 XOR Gate with MOD-GDI logic

The XOR gate is designed with MOD-GDI logic using T-SPICE has been shown below:

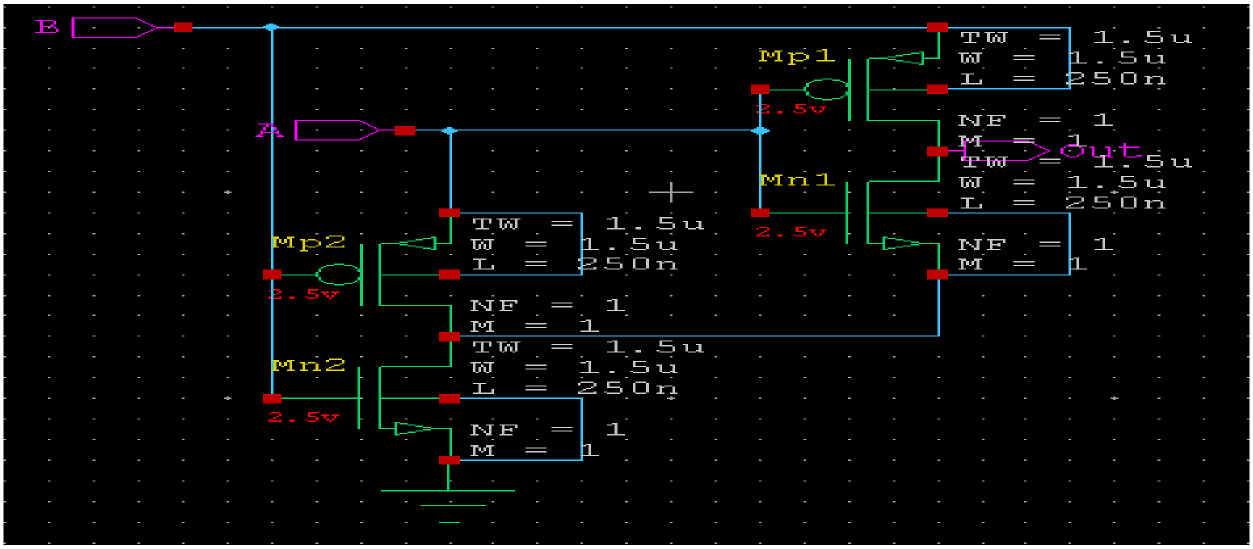


Figure 4.2 Schematic Diagram of XOR Gate with MOD-GDI logic

4.2.1.3 Two-Input MUX with MOD-GDI logic

The 2-input MUX is designed with MOD-GDI logic using T-SPICE has been shown below:

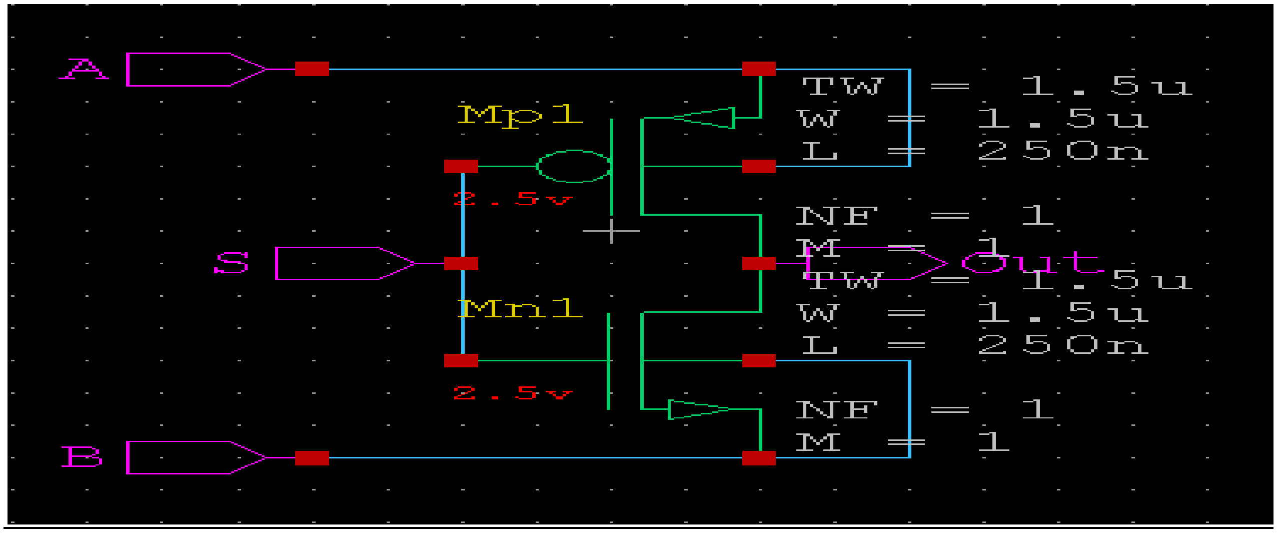


Figure 4.3 Schematic Diagram of 2-input MUX with MOD-GDI logic

4.2.1.4 Half Adder Circuit with MOD-GDI logic

The Half Adder Circuit is designed with MOD-GDI logic by combining AND gate and XOR gate using T-SPICE has been shown below:

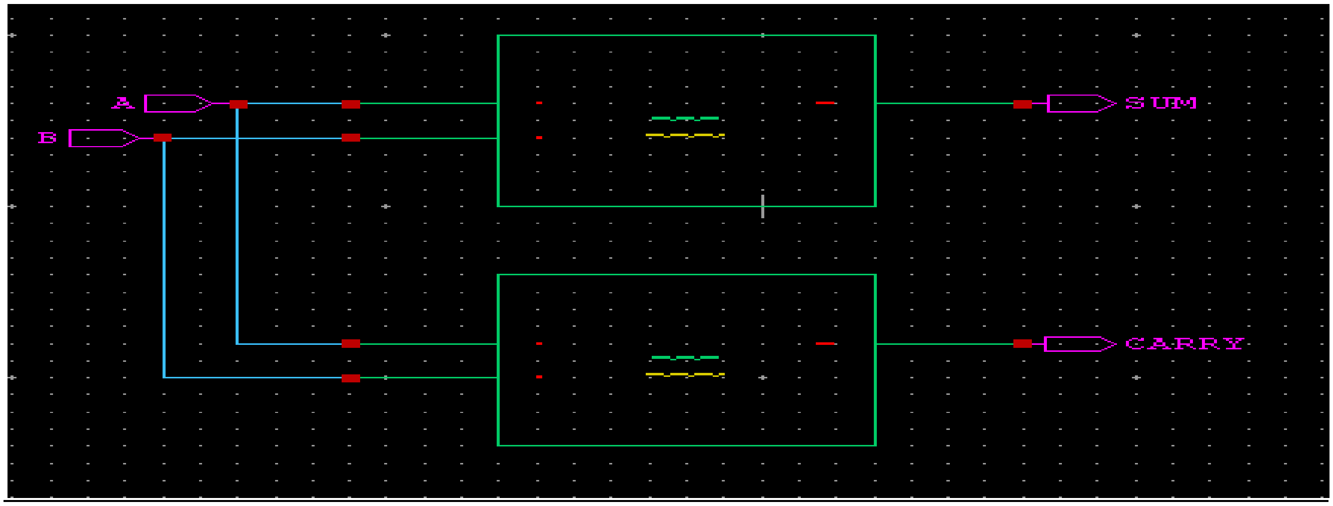


Figure 4.4 Schematic Diagram of Half Adder with MOD-GDI logic

4.2.1.5 Full Adder Circuit with MOD-GDI logic

The Full Adder Circuit is designed with MOD-GDI logic by combining 2-input MUX and XOR gate using T-SPICE has been shown below:

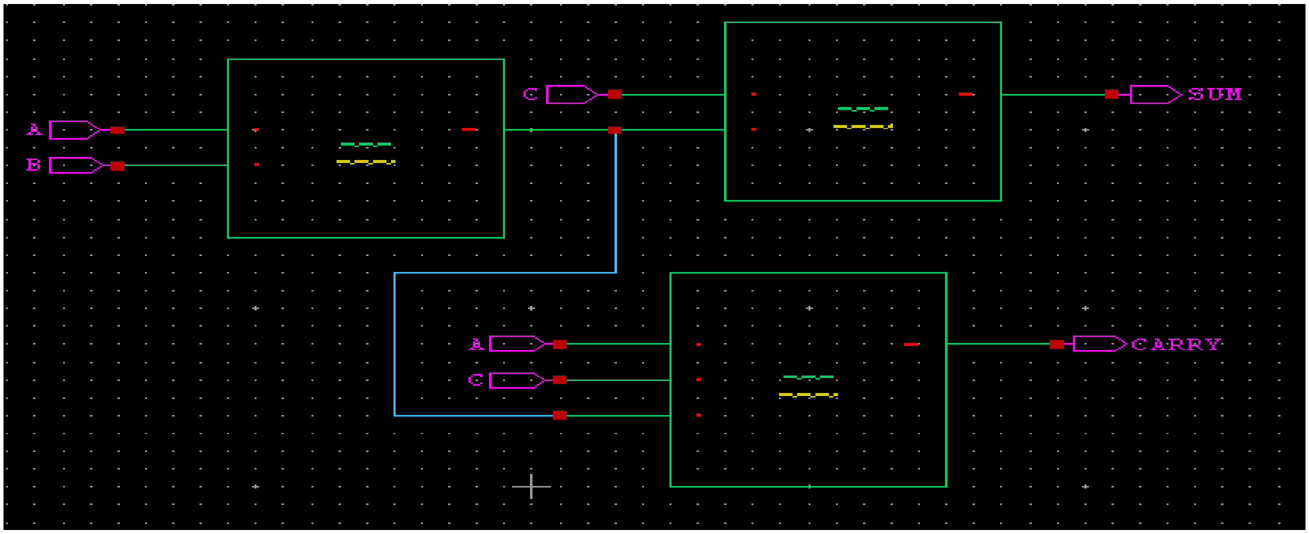


Figure 4.5 Schematic Diagram of Full Adder with MOD-GDI logic

4.2.2 DESIGN OF EXISTING 4-BIT ARRAY MULTIPLIER USING MOD-GDI LOGIC

The most significant aspect of array multiplier architecture is its regular structure. The Array multiplier has the advantage of being simple to design because the 1-bit adders are linked in an array. The AND gates of an N X N Array multiplier compute the partial products Xi Yi terms at the same time, where Xi and Yi represent the bits of the multiplicand and multiplier respectively. These terms are combined using *n* half adders and *n (n-2)* full adders. AND gates, half adders, and full adders are the main components of the Array multiplier. Figure 4.6 shows the schematic design of the array multipliers utilizing the MOD –GDI logic.

Initially, fundamental blocks like two-input AND gate, XOR gate, two-input MUX. By using these half adder, and full adder are constructed first, and then a 4-bit array multiplier is designed by utilizing these blocks. This 4×4 bit array multiplier design is implemented on T-SPICE. This design consisting of totals 16 numbers of AND gate, 4 numbers of half adders and 8 numbers of full adders as shown in Figure 4.6.

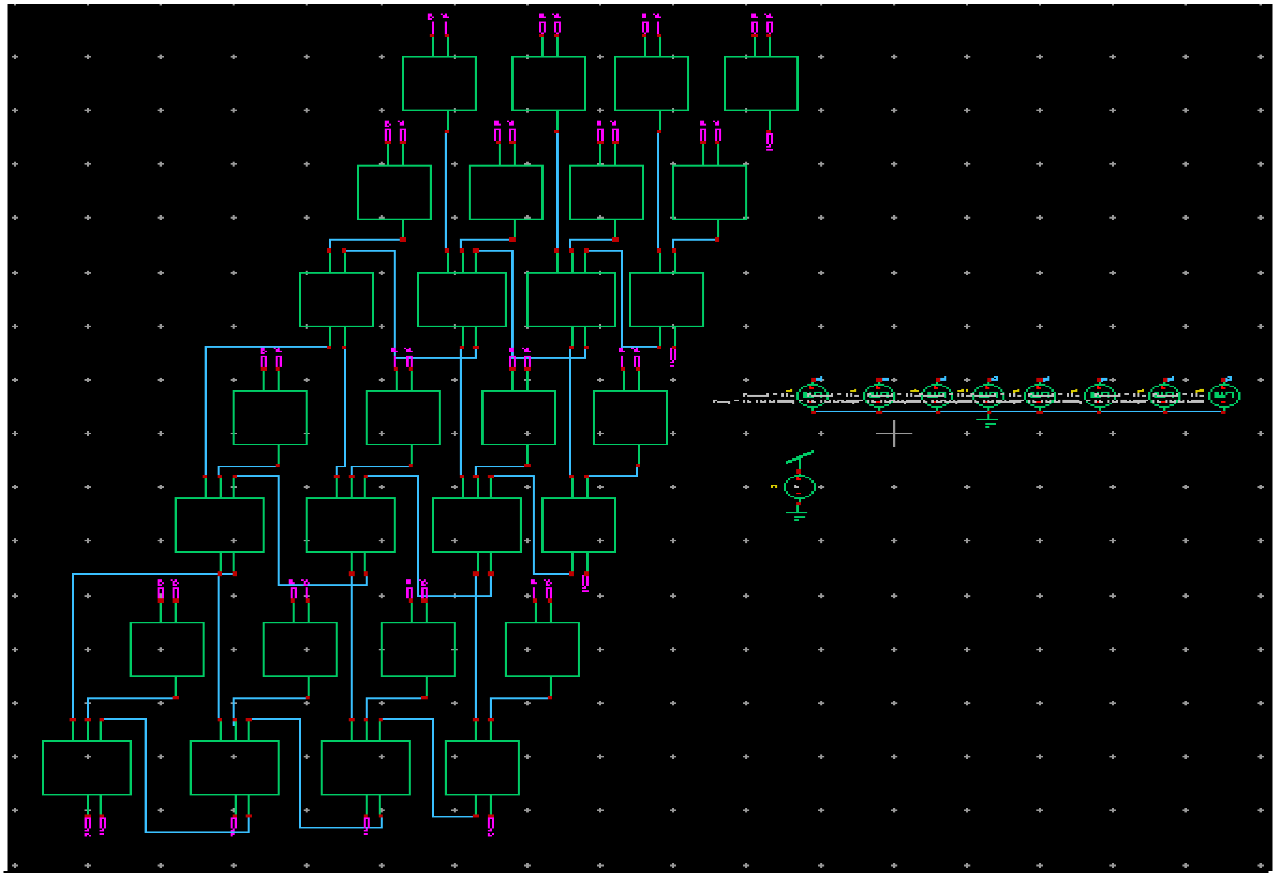


Figure 4.6 Schematic Diagram of Existing 4-Bit Array Multiplier

4.3 ARCHITECTURE OF MODIFIED 4-BIT "URDHVA TIRYAKBYHAM" (UT) MULTIPLIER USING MOD-GDI LOGIC

The "Urdhva-Tiryakbhyam" Sutra is already been used in many cases for the design of efficient multipliers. Urdhava-Tiryakbhyam is used to achieve parallelism in the creation of partial products and their summation. Recent development in the area of VLSI design introduced a new technique called, Modified Gate Diffusion Input (MOD-GDI) approach. Though many researched have already been carried out in the domain of UT multiplier, modified design of UT multiplier using MOD-GDI logic has not yet been addressed by the researchers. This area has been taken up in this research for the design of UT multiplier.

The procedure for multiplication utilizing the Urdhva Tiryagbhyam sutra is as follows: The first step is to multiply the multiplicand's leftmost digit by the multiplier's first digit. In the result, the product appears as the first digit. In the second step, add the product by multiplying the second digit of the multiplicand with the first digit of the multiplier and the first digit of the multiplicand with the second digit of the multiplier. In the solution, the result is expressed as the second digit. The third step is to multiply the multiplicand's second digit by the multiplier's second digit, and then write the product as the third digit in the answer.

4.3.1 THE FUNDAMENTAL BLOCKS OF MODIFIED UT MULTIPLIER

The fundamental blocks required for the design of 4-bit UT multiplier have been developed using MOD-GDI logic and simulated with the help of T-SPICE as illustrated below:

4.3.1.1 AND Gate with MOD-GDI logic

The AND gate is designed with MOD-GDI logic using T-SPICE has been shown below:

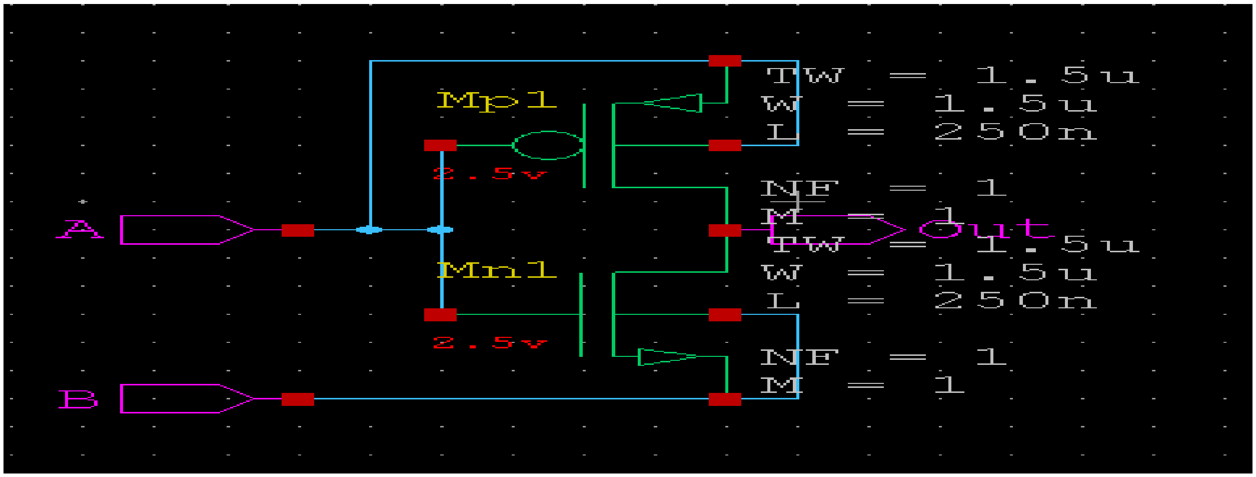


Figure 4.7 Schematic Diagram of AND Gate with MOD-GDI logic

4.3.1.2 XOR Gate with MOD-GDI logic

The XOR gate is designed with MOD-GDI logic using T-SPICE has been shown below:

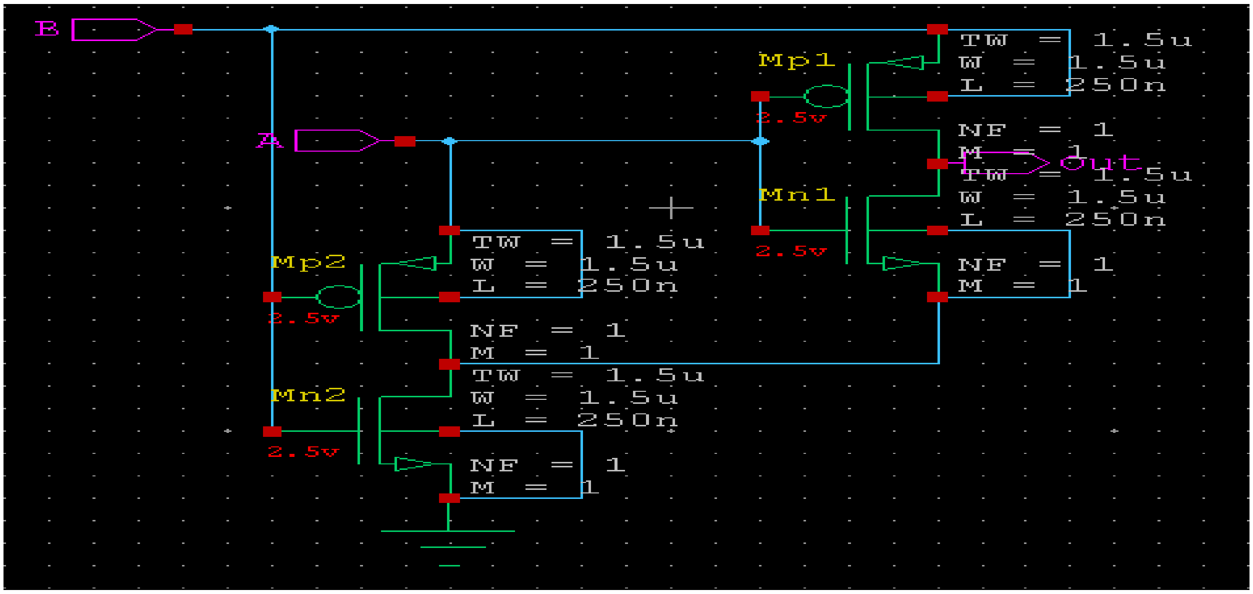


Figure 4.8 Schematic Diagram of XOR Gate with MOD-GDI logic

4.3.1.3 Two-Input MUX with MOD-GDI logic

The 2-input MUX is designed with MOD-GDI logic using T-SPICE has been shown below:

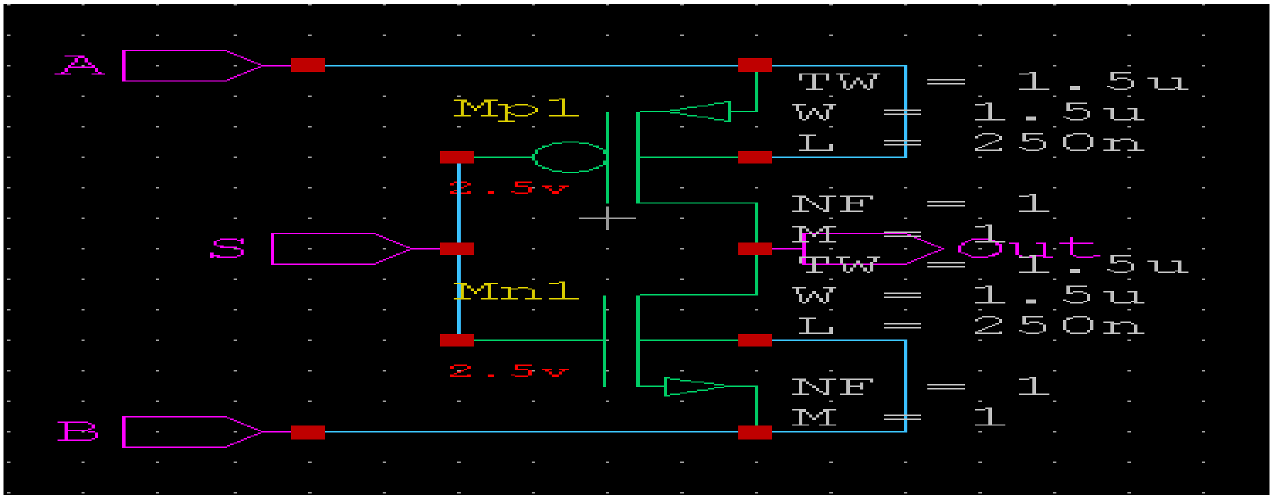


Figure 4.9 Schematic Diagram of 2-input MUX with MOD-GDI logic

4.3.1.4 Half Adder Circuit with MOD-GDI logic

The Half Adder Circuit is designed with MOD-GDI logic by combining AND gate and XOR gate using T-SPICE has been shown below:

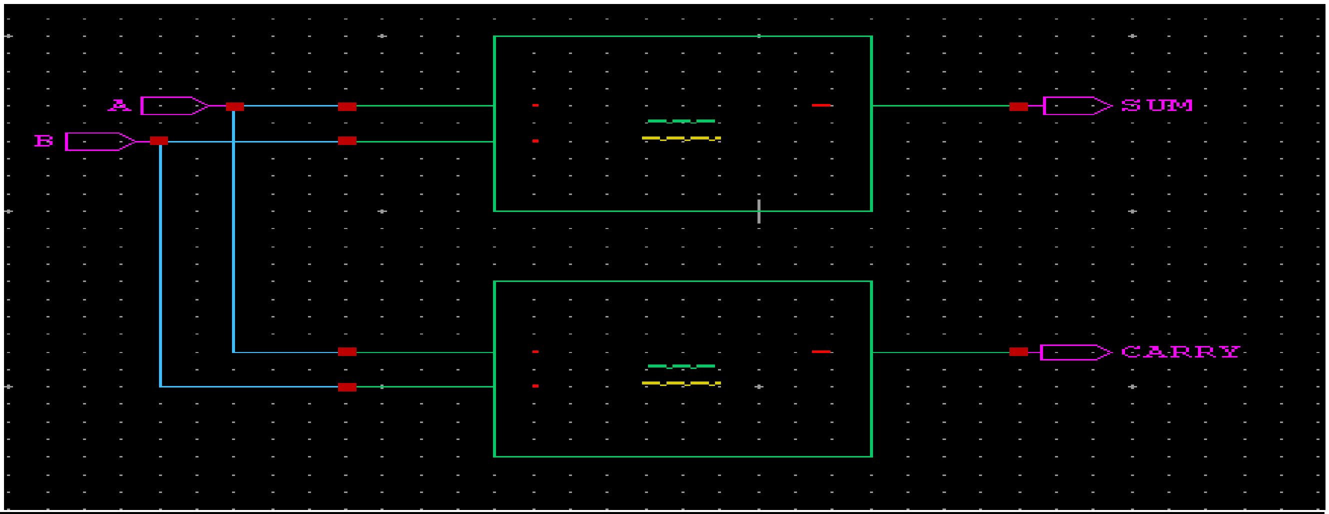


Figure 4.10 Schematic Diagram of Half Adder with MOD-GDI logic

4.3.1.5 Full Adder Circuit with MOD-GDI logic

The Full Adder Circuit is designed with MOD-GDI logic by combining 2-input MUX and XOR gate using T-SPICE has been shown below:

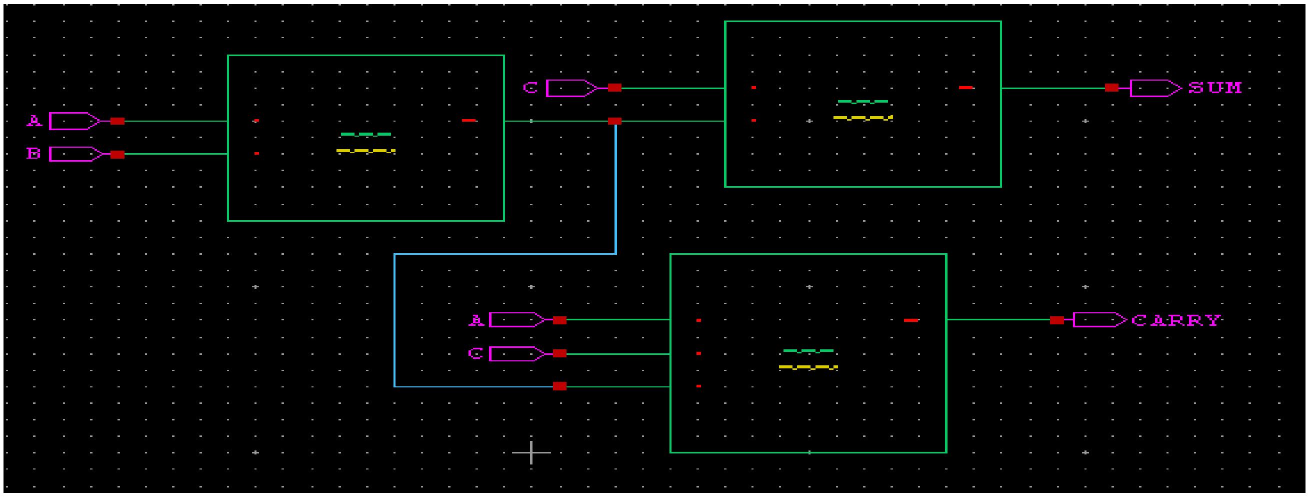


Figure 4.11 Schematic Diagram of Full Adder with MOD-GDI logic

4.3.1.6 Four-bit Ripple Carry Adder Using Full Adder with MOD-GDI logic

The Ripple Carry Adder Circuit is designed with MOD-GDI logic by combining four full adder circuits using T-SPICE has been shown below:

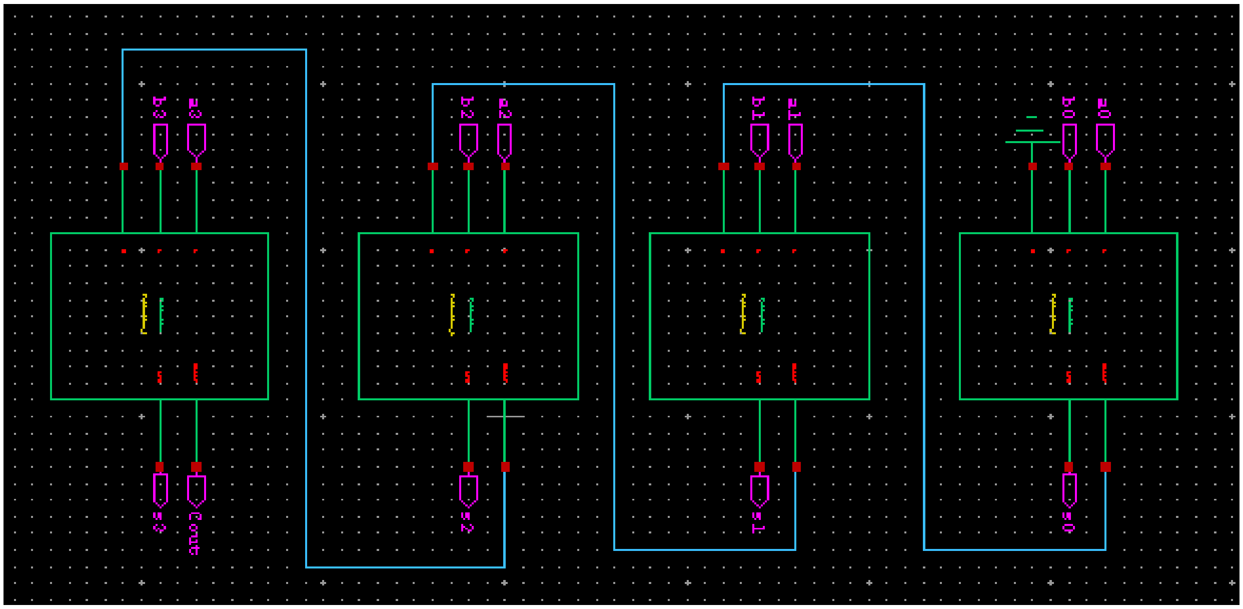


Figure 4.12 Schematic Diagram of 4-bit Ripple Carry Adder with MOD-GDI logic

4.3.1.7 Four-bit Carry Save Adder Using Full Adder and Ripple Carry Adder with MOD-GDI logic

The Carry Save Adder Circuit is designed with MOD-GDI logic by combining four full adder circuits and a Ripple Carry Adder using T-SPICE has been shown below:

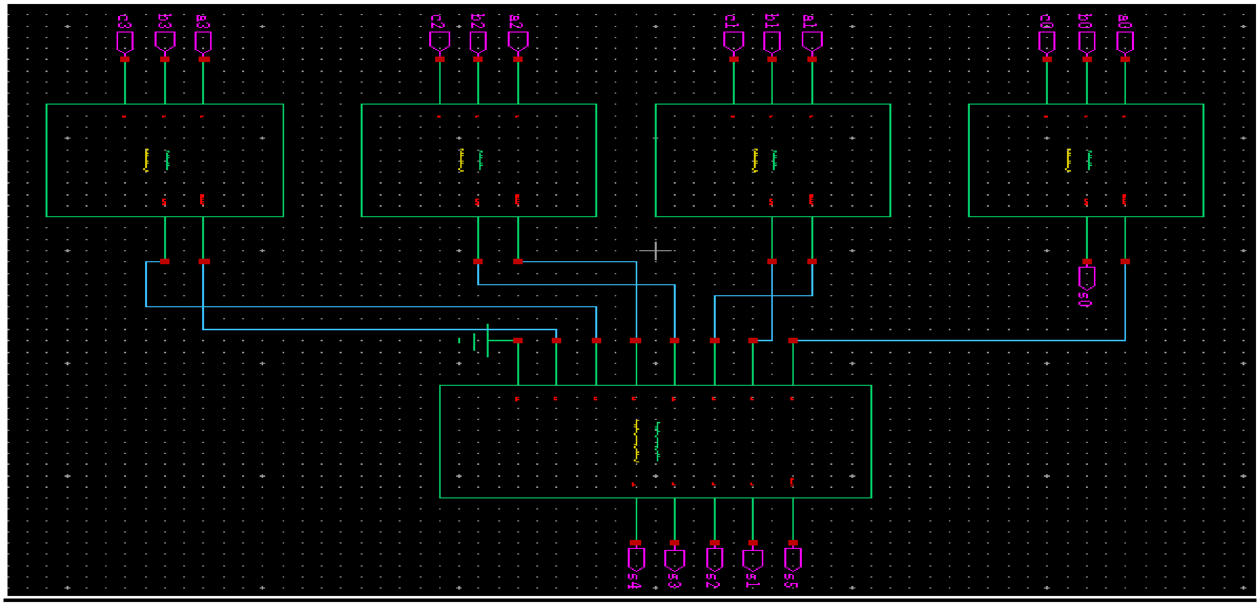


Figure 4.13 Schematic Diagram of 4-bit Carry Save Adder with MOD-GDI logic

4.3.1.8 Two-bit Adder Using Full Adder and Half Adder with MOD-GDI logic

The 2-bit Adder Circuit is designed with MOD-GDI logic by combining a half adder circuits and a full adder circuit using T-SPICE has been shown below:

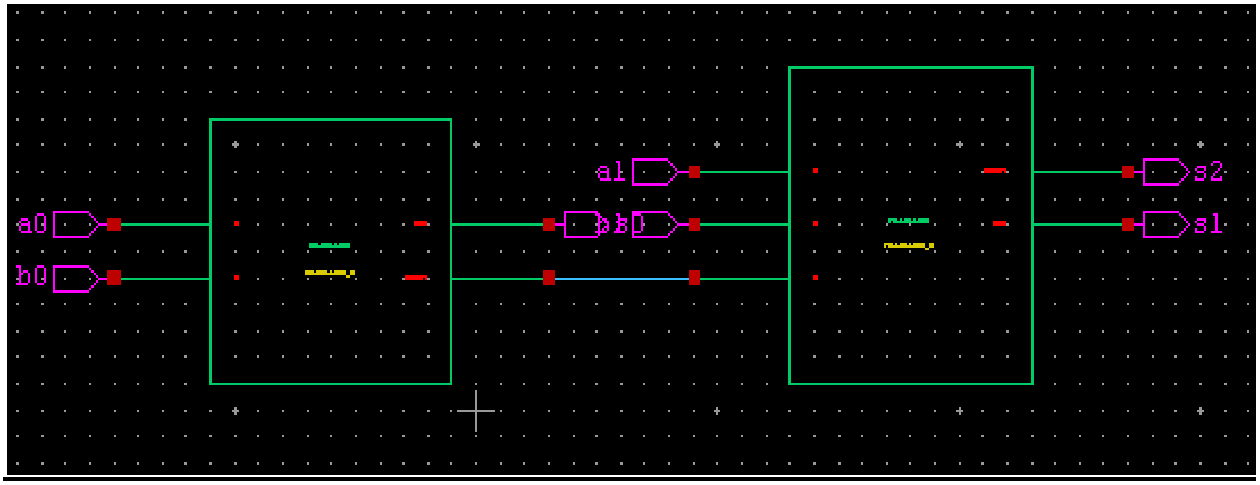


Figure 4.14 Schematic Diagram of 2-bit Adder with MOD-GDI logic

4.3.1.9 Two-bit UT Multiplier Circuit Using AND Gate and Half Adder with MOD-GDI logic

The 2-bit UT multiplier Circuit is designed with MOD-GDI logic by combining four number of AND gates and two half adder circuits using T-SPICE has been shown below:

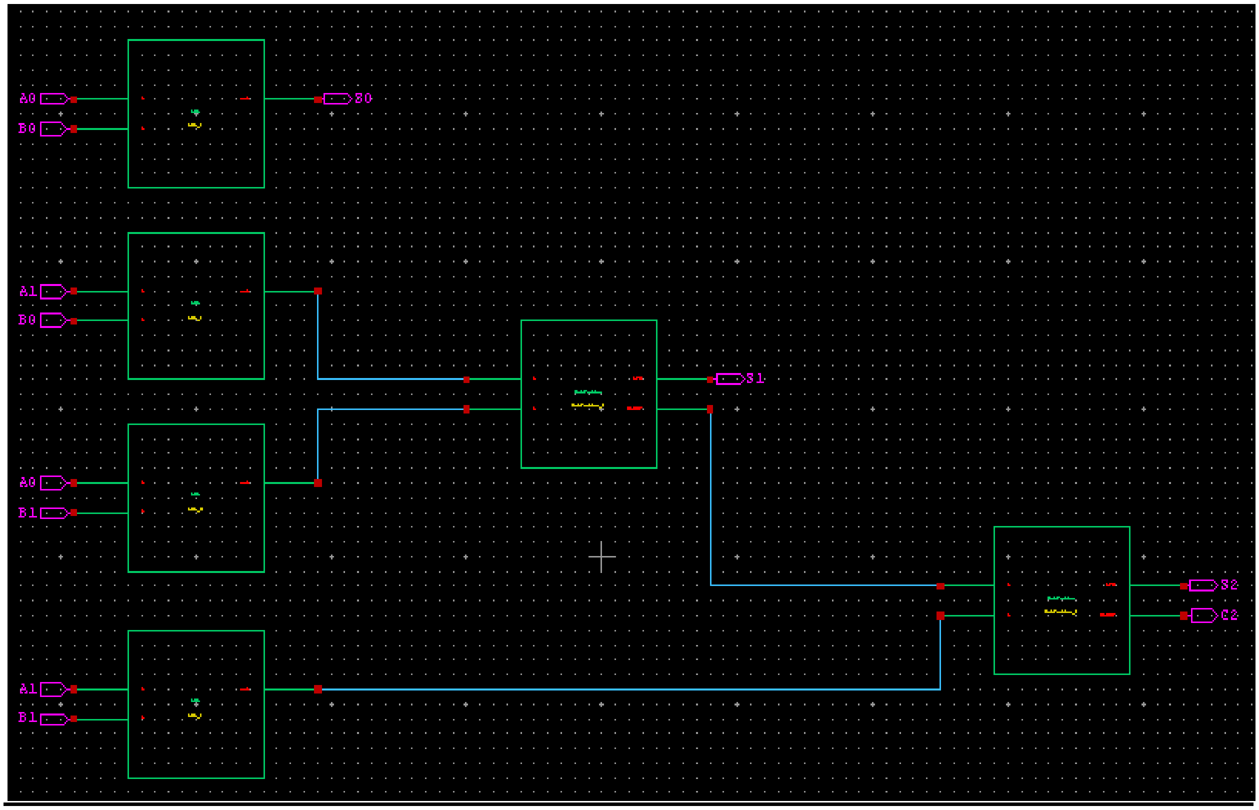


Figure 4.15 Schematic Diagram of 2-bit UT Multiplier with MOD-GDI logic

4.3.2 DESIGN OF MODIFIED 4-BIT UT MULTIPLIER

The modified 4-bit Urdhva multiplier is modified using MOD-GDI logic and created in a hierarchical form. 2-bit multipliers, which are made up of AND gates and half adder blocks, are the basic building blocks of the developed multiplier. This 2-bit multiplier is used to create the hierarchical form of this 4-bit UT multiplier. The partial-product produced by each multiplier block is given to the adder block. Four 2-bit multipliers, a 4-bit carry-save adder, a half adder, and a 2-bit adder are required for this 4-bit UT multiplier architecture.

Initially, fundamental blocks like two-input AND gate, XOR gate, two-input MUX are designed. Next, by using these fundamental blocks, other blocks like, half adder, full adder, ripple carry adder, 4-bit carry save adder, 2-bit adder and 2-bit UT multipliers are constructed. Then a 4-bit UT multiplier is designed by utilizing these blocks. The design of 4-bit UT multiplier, consisting of four 2-bit UT multipliers, a 4-bit carry save adder, a half adder, and a 2-bit adder, has been shown in Figure 4.16.

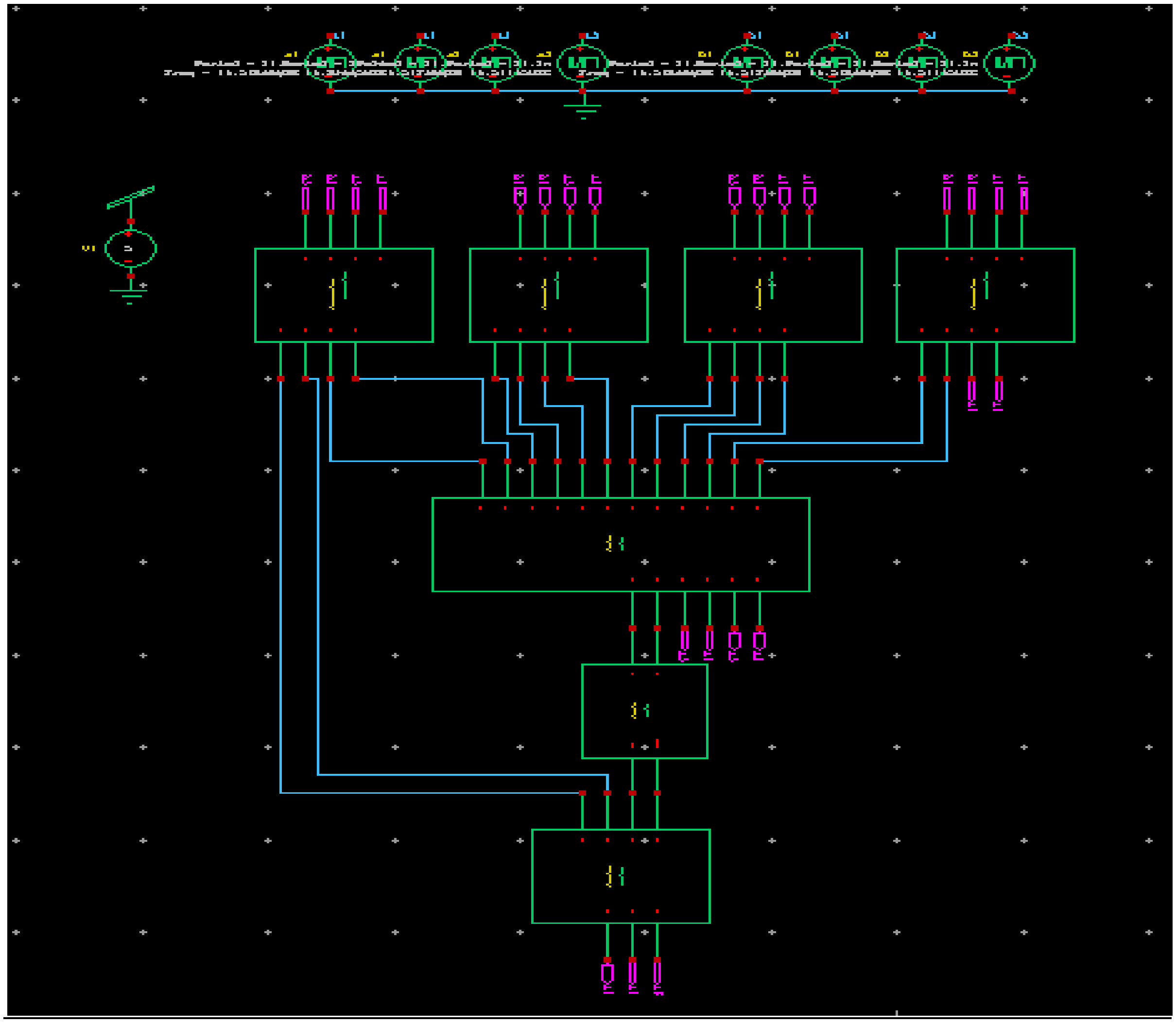


Figure 4.16 Schematic Diagram of Existing 4-Bit UT Multiplier

4.4 ARCHITECTURE OF THE PROPOSED 4-BIT "NIKHILAM NAVATASHCARAMAM DASHATAH" (NIKHILAM) MULTIPLIER USING MOD-GDI LOGIC

The Nikhilam Sutra based proposed design of 4x4 multiplier using MOD-GDI logic has been discussed here. For this design, the same design tool T-SPICE has been used.

4.4.1 THE FUNDAMENTAL BLOCKS OF NIKHILAM MULTIPLIER

The fundamental blocks required for the design of 4-bit Nikhilam multiplier have been developed using MOD-GDI logic and simulated with the help of T-SPICE as illustrated below:

4.4.1.1 Inverter (NOT Gate) with MOD-GDI logic

The Inverter is designed with MOD-GDI logic using T-SPICE has been shown below:

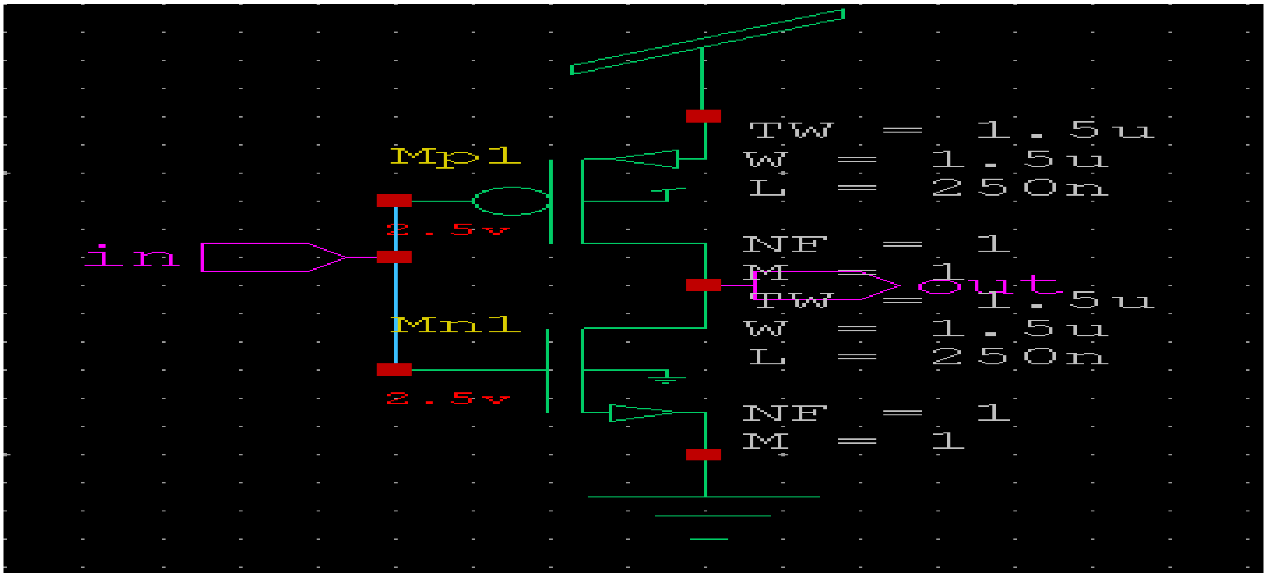


Figure 4.17 Schematic Diagram of Inverter with MOD-GDI logic

4.4.1.2 AND Gate with MOD-GDI logic

The AND gate is designed with MOD-GDI logic using T-SPICE has been shown below:

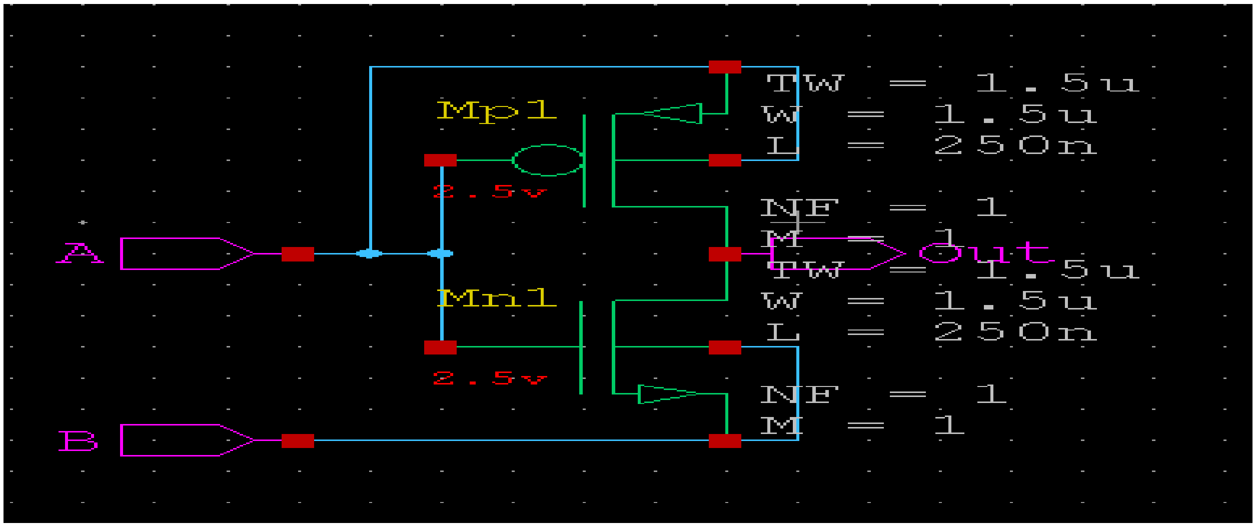


Figure 4.18 Schematic Diagram of AND Gate with MOD-GDI logic

4.4.1.3 XOR Gate with MOD-GDI logic

The XOR gate is designed with MOD-GDI logic using T-SPICE has been shown below:

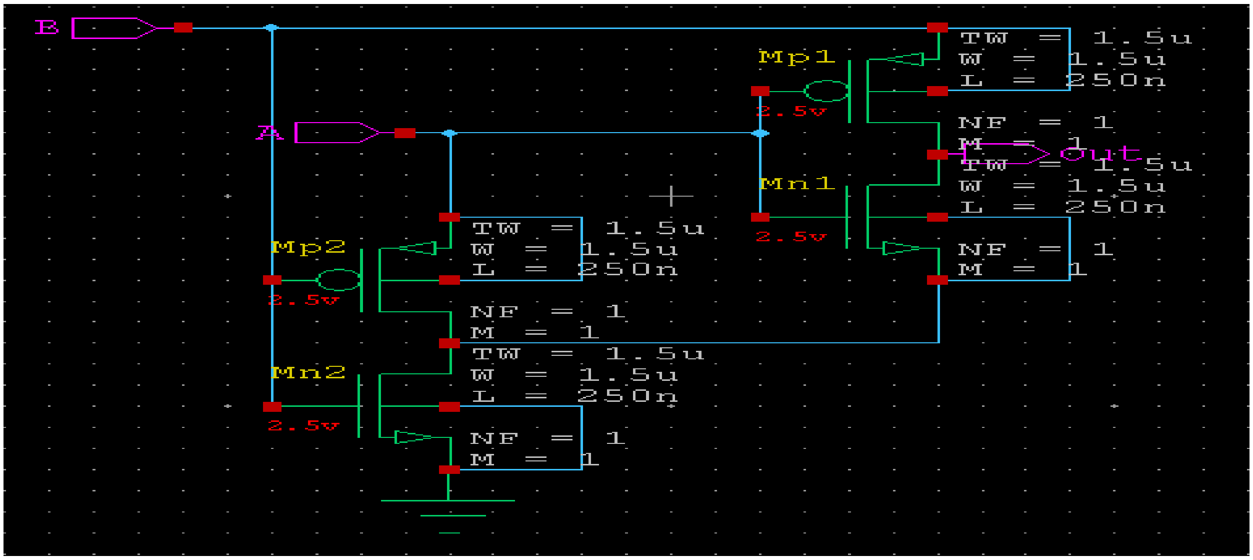


Figure 4.19 Schematic Diagram of XOR Gate with MOD-GDI logic

4.4.1.4 Two-Input MUX with MOD-GDI logic

The 2-input MUX is designed with MOD-GDI logic using T-SPICE has been shown below:

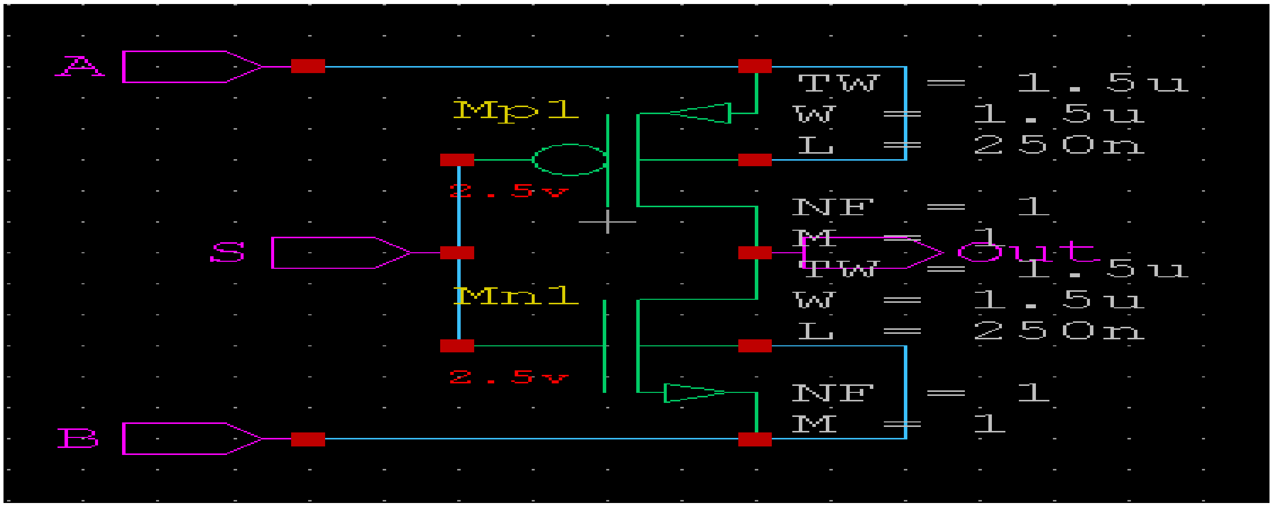


Figure 4.20 Schematic Diagram of 2-input MUX with MOD-GDI logic

4.4.1.5 Half Adder Circuit with MOD-GDI logic

The Half Adder Circuit is designed with MOD-GDI logic by combining AND gate and XOR gate using T-SPICE has been shown below:

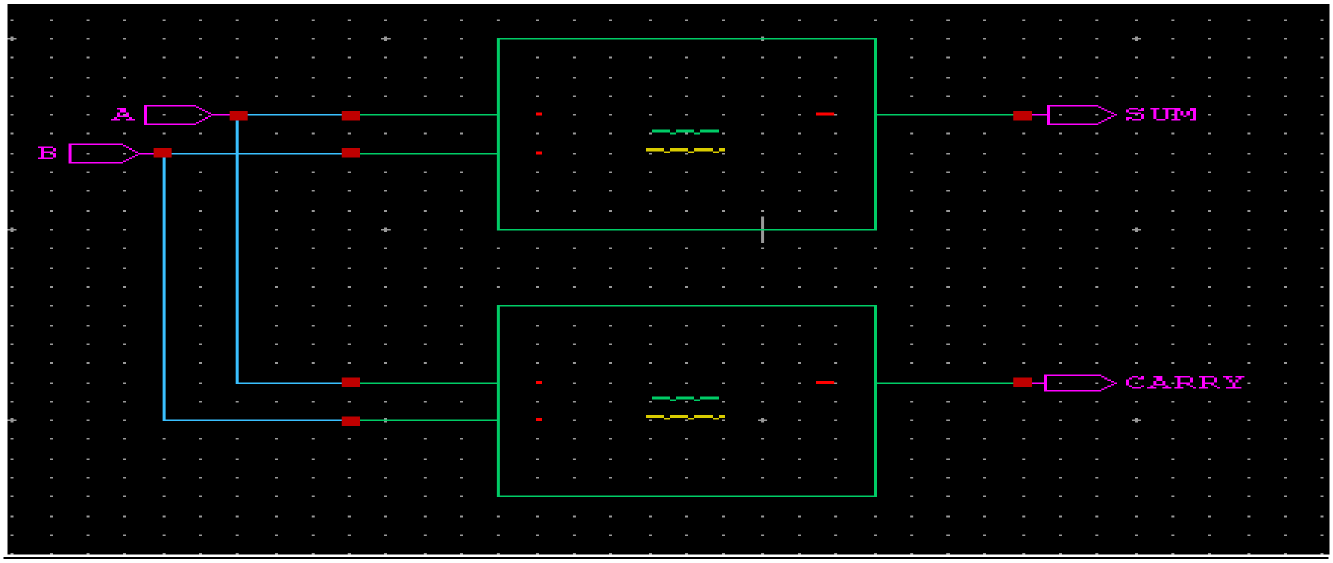


Figure 4.21 Schematic Diagram of Half Adder with MOD-GDI logic

4.4.1.6 Full Adder Circuit with MOD-GDI logic

The Full Adder Circuit is designed with MOD-GDI logic by combining 2-input MUX and XOR gate using T-SPICE has been shown below:

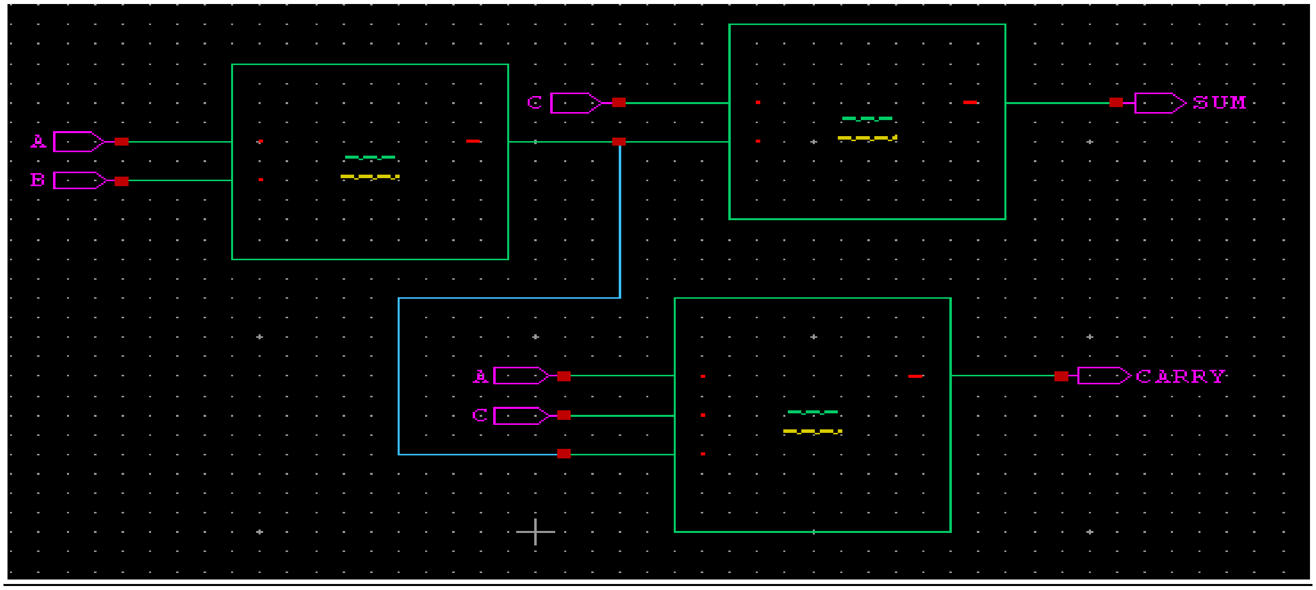


Figure 4.22 Schematic Diagram of Full Adder with MOD-GDI logic

4.4.1.7 Four-bit Ripple Carry Adder Using Full Adder with MOD-GDI logic

The Ripple Carry Adder Circuit is designed with MOD-GDI logic by combining four full adder circuits using T-SPICE has been shown below:

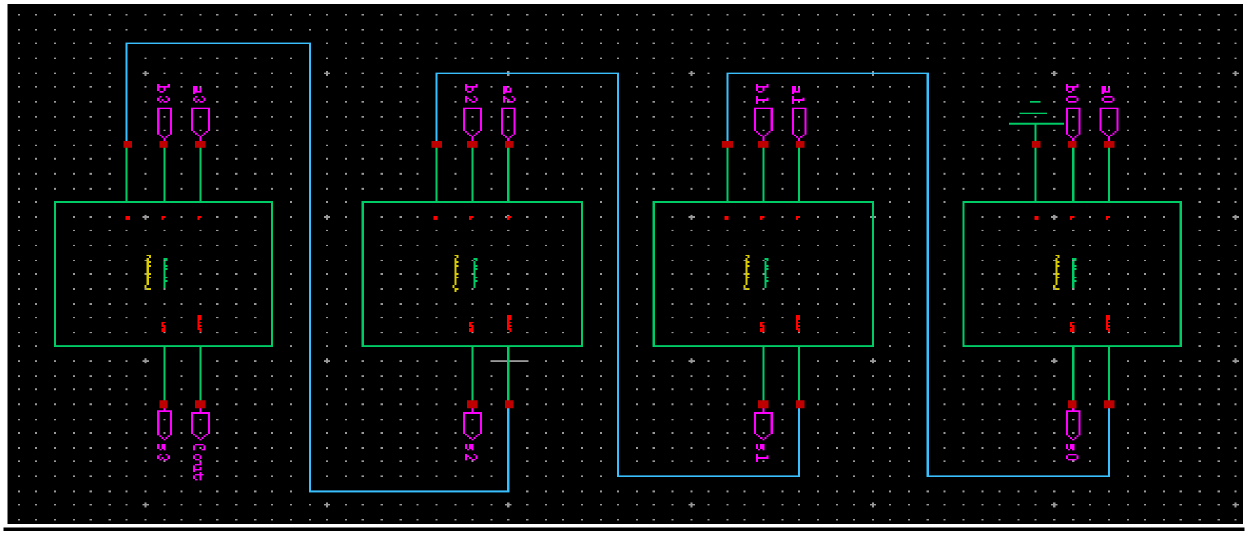


Figure 4.23 Schematic Diagram of 4-bit Ripple Carry Adder with MOD-GDI logic

4.4.1.8 Four-bit Carry Save Adder Using Full Adder and Ripple Carry Adder with MOD-GDI logic

The Carry Save Adder Circuit is designed with MOD-GDI logic by combining four full adder circuits and a Ripple Carry Adder using T-SPICE has been shown below:

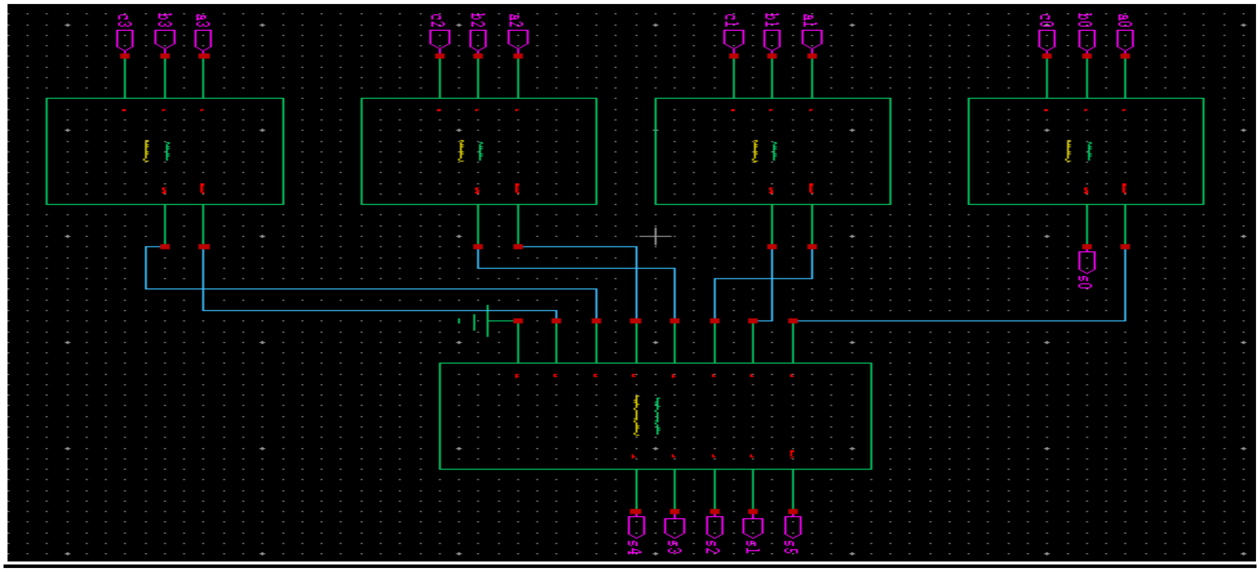


Figure 4.24 Schematic Diagram of 4-bit Carry Save Adder with MOD-GDI logic

4.4.1.9 2's Complement Circuit Using Inverter and Half Adder with MOD-GDI logic

The 2's complement Circuit is designed with MOD-GDI logic by combining four inverters and four half adder circuits using T-SPICE has been shown below:

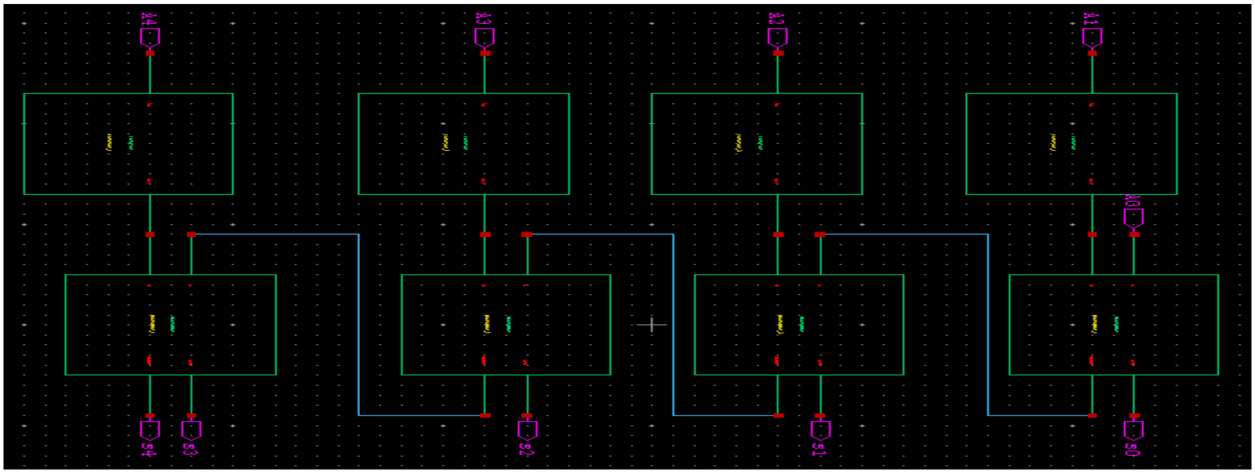


Figure 4.25 Schematic Diagram of 2’s Complement with MOD-GDI logic

4.4.1.10 Four-bit Multiplier Circuit Using AND Gate, Half Adder, and Full Adder with MOD-GDI logic

The 4-bit multiplier Circuit is designed with MOD-GDI logic by combining AND gates half adders and full adder circuits using T-SPICE has been shown below:

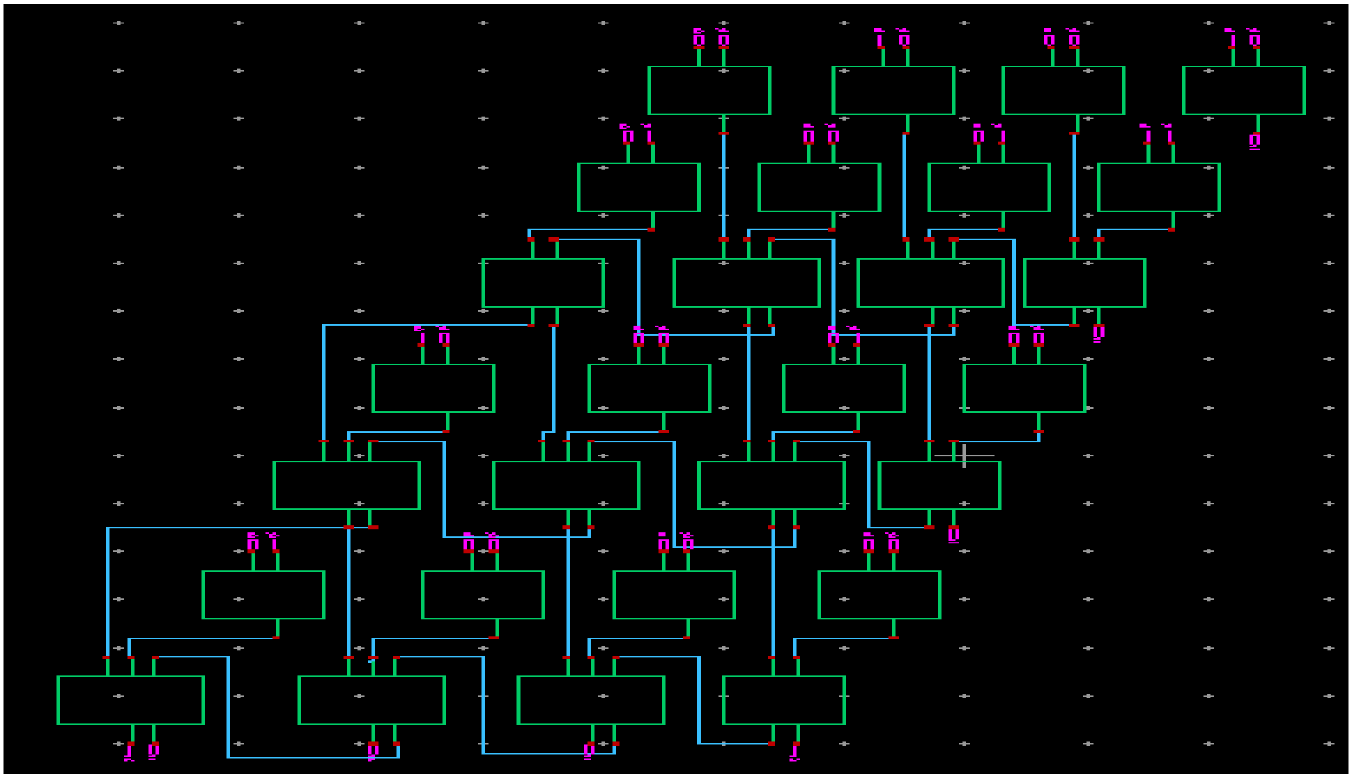


Figure 4.26 Schematic Diagram of 4-bit multiplier with MOD-GDI logic

4.4.2 DESIGN OF PROPOSED 4-BIT NIKHILAM MULTIPLIER

The architecture of the Nikhilam multiplier is depicted in the diagram below. The two inputs of the multiplier *a* (*a4 a3 a2 a1*) and multiplicand b (*b4 b3 b2 b1*) are complemented first, and then multiplied. In this case, the utilized multiplier has a significant impact on the calculation of delay. The output of the multiplier is then combined with the two inputs *a* and *b*. The R.H.S. of the original product is the right-hand side result of the multiplier, and the L.H.S. of the original product is the left-hand side result of the adder.

Initially, fundamental blocks like two-input AND gate, XOR gate, two-input MUX are designed. Next, by using these fundamental blocks, other blocks like, full adder, ripple carry adder, 4-bit carry save adder, and 2's complement are constructed. Then a 4-bit UT multiplier is designed by utilizing these blocks. The design of 4-bit Nikhilam multiplier, consisting of two 2's complements, one 4-bit multiplier, and one 4-bit carry-save adder has been shown in Figure 4.27.

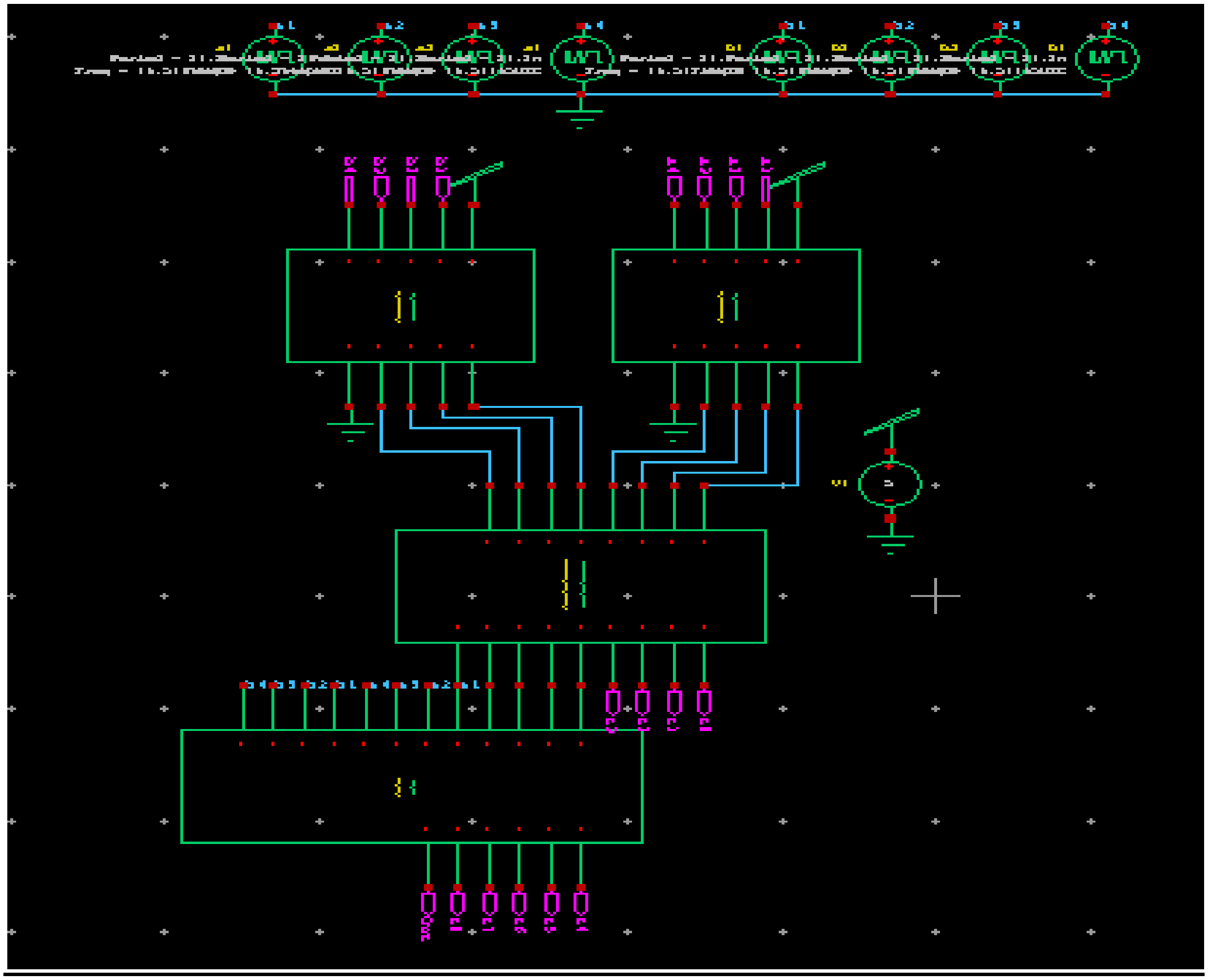


Figure 4.27 Schematic Diagram of Proposed 4-Bit Nikhilam Multiplier

4.5 PROPOSED ALGORITHM FOR PERFORMANCE ANALYSIS OF NIKHILAM MULTIPLIER IN TERMS OF DIFFERENT NEAREST BASE

4.5.1 Algorithm for Multiplication when Both Numbers are Lesser than Base

The algorithm of multiplication when both Multiplier and Multiplicand are lesser than nearest base has been shown as follows:

- **Step 1:** Select the value of the Multiplier 'X' and Multiplicand 'Y' for multiplication.
- **Step 2:** Select the nearest base value 'B' (i.e. Power of '10') w.r.t X and Y
- **Step 3:** Find out the difference (M) = Nearest base (B) – Multiplier (X)
- **Step 4:** Find out the difference (N) = Nearest base (B) – Multiplicand (Y)
- **Step 5:** Find out the product of two differences (C) = M*N
- **Step 6:** D = X - N= Y - M
- **Step 6:** Result = B*D + C

4.5.2 Algorithm for Multiplication when Both Numbers are Greater than Base

The algorithm of multiplication when both Multiplier and Multiplicand are greater than nearest base has been shown as follows:

- **Step 1:** Select the value of the Multiplier 'X' and Multiplicand 'Y' for multiplication.
- **Step 2:** Select the nearest base value 'B' (i.e. Power of '10') w.r.t X and Y
- **Step 3:** Find out the difference (M) = Multiplier (X) - Nearest base (B)
- **Step 4:** Find out the difference (N) = Multiplicand (Y) - Nearest base (B)
- **Step 5:** Find out the product of two differences (C) = M*N
- **Step 6:** D = X + N= Y + M
- **Step 6:** Result = B*D + C

4.5.3 Algorithm for Multiplication when Multiplier is Greater than Base and Multiplicand is Lesser than Base

- **Step 1:** Select the value of the Multiplier 'X' and Multiplicand 'Y' for multiplication.
- **Step 2:** Select the nearest base value 'B' (i.e. Power of '10') w.r.t X and Y
- **Step 3:** Find out the difference (M) = Multiplier (X) - Nearest base (B)
- **Step 4:** Find out the difference (N) = Nearest base (B) – Multiplicand (Y)
- **Step 5:** Find out the product of two differences (C) = B - (M*N)
- **Step 6:** D = (X - N) - 1= (Y + M) - 1
- **Step 6:** Result = B*D + C

4.6 DISCUSSIONS

In this chapter, the design of existing 4x4 Array multiplier based on MOD-GDI Logic has been shown in detail describing the basic building blocks and accumulation of these blocks for the final design. Model has been developed using T-SPICE. Next, the design of the existing 4x4 UT multiplier has been modified by incorporating the MOD-GDI logic into it during the design process. While designing, initially, the basic building blocks have been developed and then finally the multiplier has been modeled using T-SPICE. Finally, a 4-bit Nikhilam multiplier based on MOD-GDI technique has been proposed and implemented on T-SPICE. All the aforementioned required building blocks are developed and finally these blocks are accumulated properly to have the final design. Apart from the proposed 4x4 Nikhilam multiplier, other designs such as, existing 4x4 Array multiplier based on MOD-GDI Logic and the modified 4x4 UT multiplier based on MOD-GDI Logic are also elaborated here to compare the performance of the proposed design of the multiplier with the other contrast works. Finally, the performance analysis of the proposed Nikhilam multiplier has also been carried out corresponding to the closeness of different numbers to the chosen base values.

CHAPTER 5

RESULT AND ANALYSIS

5.1 OVERVIEW

In this chapter, the simulation results of the proposed design of a 4-bit Nikhilam multiplier based on MOD-GDI are presented in detail. Other than the proposed design, the simulation results of a 4-bit Array multiplier and UT multiplier based on MOD-GDI Logic are also elaborated here for the purpose of comparison. The comparison is made in terms of the transistor count, propagation latency, as well as total power dissipation.

The proposed architectures are implemented using TSPICE software (S-Edit Win64 16.30.20150626.04:37:24). There are the specifications of the computer used for testing - Windows 7 64-bit OS, 4 GB RAM, Intel Core i5-3317U CPU, processor running at 1.70 GHz.

5.2 SIMULATION ENVIRONMENT

The simulation platform utilized in all the designs is TANNER SPICE (S-Edit Win64 16.30.20150626. 04:37:24), which gives a CMOS level design of the proposed structure as well as information about power consumption and time delay. The settings of various parameters for the design of the proposed model as well as for the entire simulations have been summarized below:

- All of the design structures are modeled in 250 nm technology.
- The architectures are operated at a frequency of 20 MHz
- As per the necessity of drastic power saving in MOD-GDI circuits, slow transition clock power supply, i.e. trapezoidal clock is applied at the supply. In this thesis, both the rise time and fall time of the clock are considered to be 4 ns and pulse width to be 20 ns.
- In all simulations, different bit patterns with a pulse width of 100 ns and rise and fall times of 1ns are used as inputs.
- All the simulations are performed at a voltage of 5 V.

5.3 SIMULATION RESULTS OF 4-BIT EXISTING ARRAY MULTIPLIER

5.3.1 Output Waveforms of Existing 4-bit Array Multiplier

The multiplication operation is performed utilizing existing 4-bit Array multiplier. Both Multiplicand and Multiplier are used as input of this multiplier. In this section, the multiplication operation is done with the following data as shown below:

- ❖ The value of the Multiplicand → 0010
- ❖ The value of the Multiplier →0100
- ❖ Output of the Multiplication → 00001000

The output waveforms of the 4-bit Array multiplier corresponding to the above inputs are shown below:

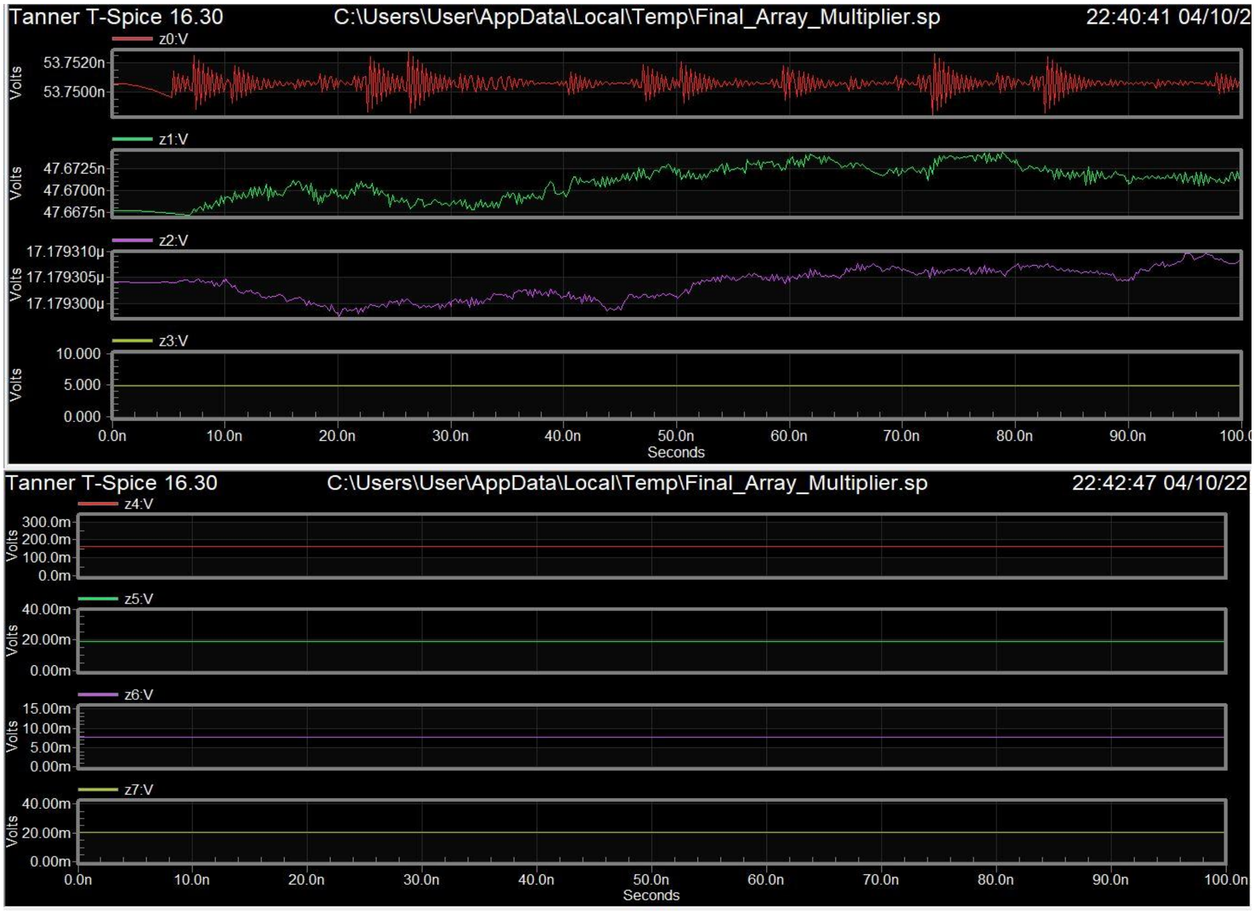


Figure 5.1 Output Waveform of Existing 4-bit Array Multiplier

The verification of the results of multiplication has been summarized in Table 5.1

Table 5.1 Multiplication result of Existing 4-bit Array Multiplier:

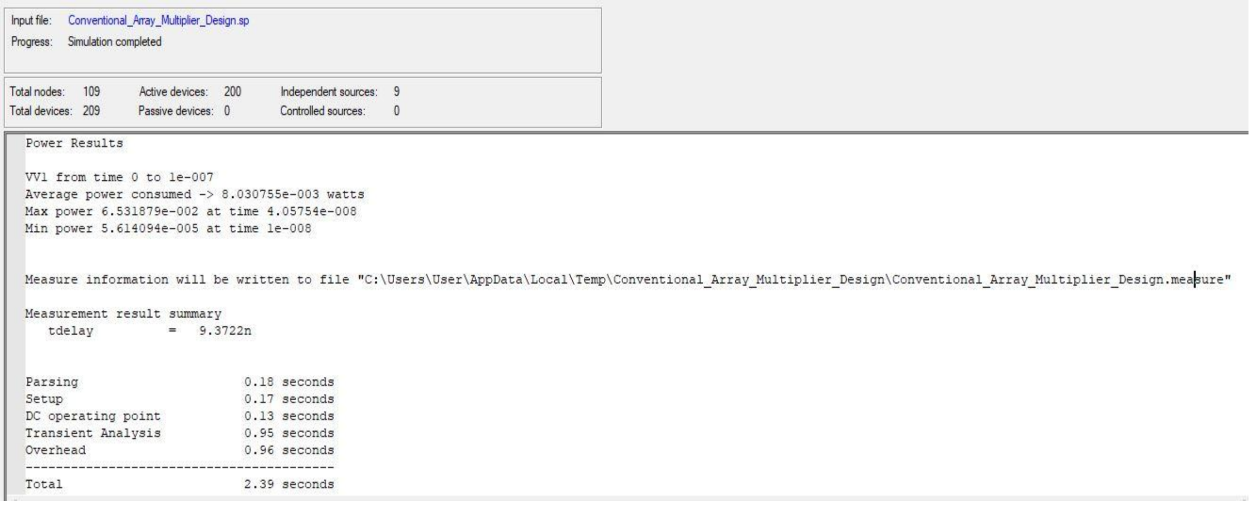
Multiplicand				Multiplier				Multiplication Result							
X0	**X1**	**X2**	**X3**	**Y0**	**Y1**	**Y2**	**Y3**	**z0**	**z1**	**z2**	**z3**	**z4**	**z5**	**z6**	**z7**
‘0’	‘1’	‘0’	‘0’	‘0’	‘0’	‘1’	‘0’	‘0’	‘0’	‘0’	‘1’	‘0’	‘0’	‘0’	‘0’

5.3.2 Power and Time Delay of Existing 4-bit Array Multiplier

The power and time delay of this multiplier simulated on T-SPICE has been derived as follows:

- Power consumption = 8.030755 mw
- Time Delay= 9.3722 ns

The screenshot of the above result is presented below:



```
Input file:     Conventional_Array_Multiplier_Design.sp
Progress:       Simulation completed

Total nodes:    109     Active devices:   200     Independent sources:   9
Total devices:  209     Passive devices:  0       Controlled sources:    0
```

```
Power Results

VV1 from time 0 to 1e-007
Average power consumed -> 8.030755e-003 watts
Max power 6.531879e-002 at time 4.05754e-008
Min power 5.614094e-005 at time 1e-008


Measure information will be written to file "C:\Users\User\AppData\Local\Temp\Conventional_Array_Multiplier_Design\Conventional_Array_Multiplier_Design.measure"

Measurement result summary
   tdelay         =   9.3722n


Parsing                     0.18 seconds
Setup                       0.17 seconds
DC operating point          0.13 seconds
Transient Analysis          0.95 seconds
Overhead                    0.96 seconds
-----------------------------------------
Total                       2.39 seconds
```

Figure 5.2 Power and Time Delay of Array Multiplier on TSPICE

5.4 SIMULATION RESULTS OF MODIFIED 4-BIT UT MULTIPLIER

5.4.1 Output Waveforms of Modified 4-bit UT Multiplier

The existing UT multiplier is modified using MOD-GDI Logic and results are presented here. Both Multiplicand and Multiplier are used as input of this multiplier. In this section, the multiplication operation is done with the following data as shown below:

- The value of the Multiplicand → 0010
- The value of the Multiplier →0100
- Output of the Multiplication → 00001000

The output waveforms of the 4-bit UT multiplier corresponding to the above inputs are shown below:

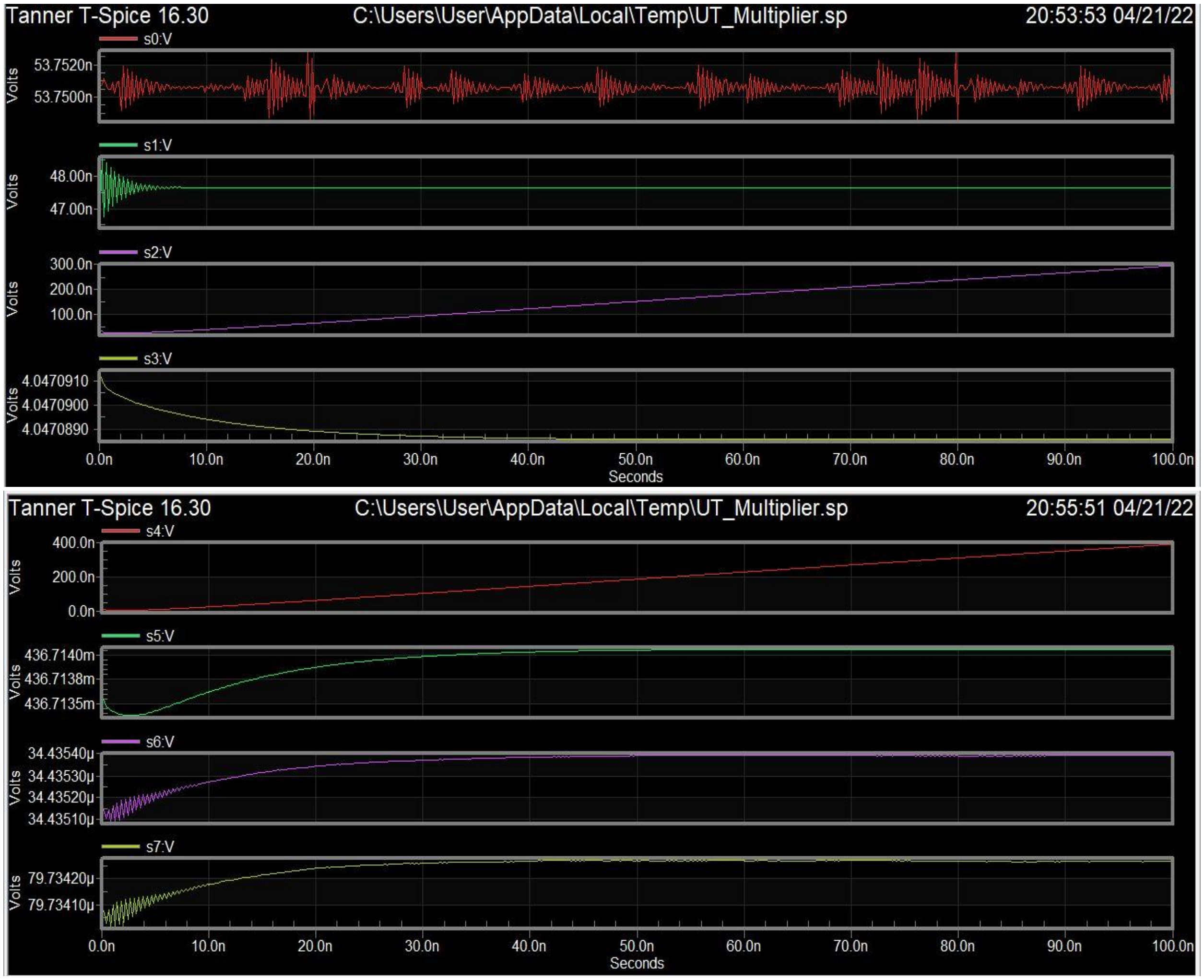


Figure 5.3 Output Waveform of Modified 4-bit UT Multiplier

The verification of the results of multiplication has been summarized in Table 5.2

Table 5.2 Multiplication result of Existing 4-bit UT Multiplier:

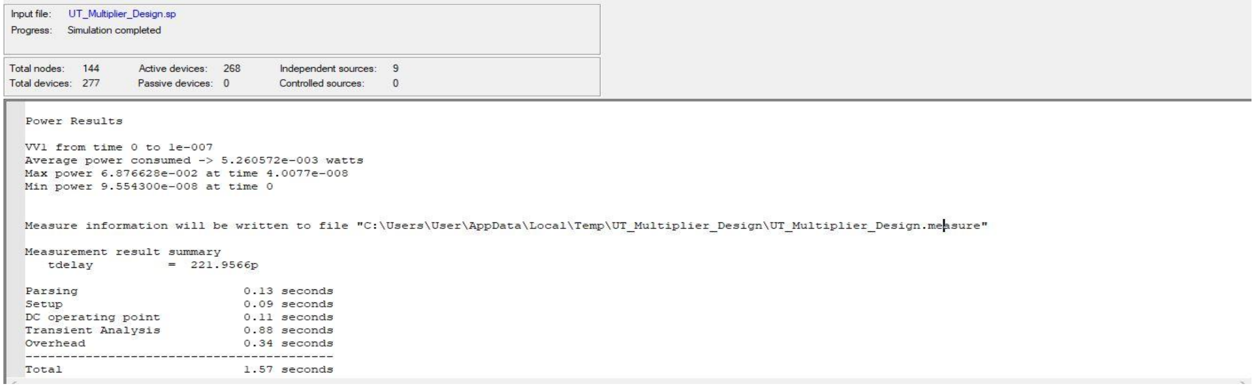
Multiplicand				**Multiplier**				**Multiplication Result**							
a0	**a1**	**a2**	**a3**	**b0**	**b1**	**b2**	**b3**	**S0**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
‘0’	‘1’	‘0’	‘0’	‘0’	‘0’	‘1’	‘0’	‘0’	‘0’	‘0’	‘1’	‘0’	‘0’	‘0’	‘0’

5.4.2 Power and Time Delay of Modified 4-bit UT Multiplier

The power and time delay of this multiplier simulated on T-SPICE has been derived as follows:

- Power consumption = 5.260572 mw
- Time Delay= 221.9566 ps

The screenshot of the above result is presented below:



```
Input file:     UT_Multiplier_Design.sp
Progress:       Simulation completed

Total nodes:    144     Active devices:   268     Independent sources:   9
Total devices:  277     Passive devices:  0       Controlled sources:    0
```

```
Power Results

VV1 from time 0 to 1e-007
Average power consumed -> 5.260572e-003 watts
Max power 6.876628e-002 at time 4.0077e-008
Min power 9.554300e-008 at time 0


Measure information will be written to file "C:\Users\User\AppData\Local\Temp\UT_Multiplier_Design\UT_Multiplier_Design.measure"

Measurement result summary
   tdelay          =  221.9566p

Parsing                      0.13 seconds
Setup                        0.09 seconds
DC operating point           0.11 seconds
Transient Analysis           0.88 seconds
Overhead                     0.34 seconds
-----------------------------------------
Total                        1.57 seconds
```

Figure 5.4 Power and Time Delay of UT Multiplier on TSPICE

5.5 SIMULATION RESULTS OF THE PROPOSED 4-BIT NIKHILAM MULTIPLIER

5.5.1 Output Waveforms of Proposed 4-bit Nikhilam Multiplier

The multiplication operation is performed utilizing proposed 4-bit Nikhilam multiplier. Both Multiplicand and Multiplier are used as input of this multiplier. In this section, the multiplication operation is done with the following data as shown below:

- The value of the Multiplicand → 0010
- The value of the Multiplier →0100
- Output of the Multiplication → 00001000

The output waveforms of the 4-bit Nikhilam multiplier corresponding to the above inputs are shown below:

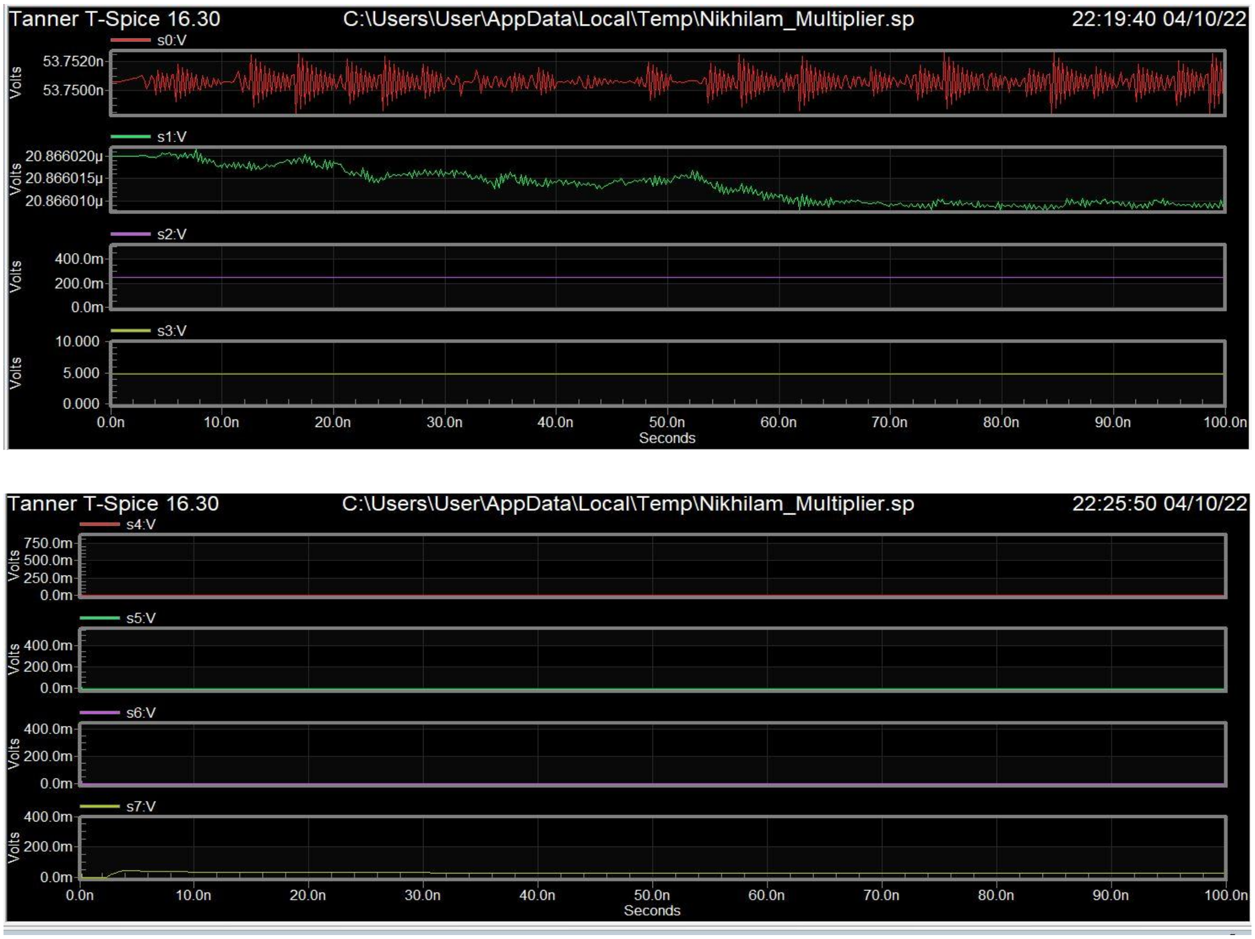


Figure 5.5 Output Waveform of Proposed 4-bit Nikhilam Multiplier

The verification of the results of multiplication has been summarized in Table 5.3

Table 5.3 Multiplication result of proposed 4-bit Nikhilam Multiplier:

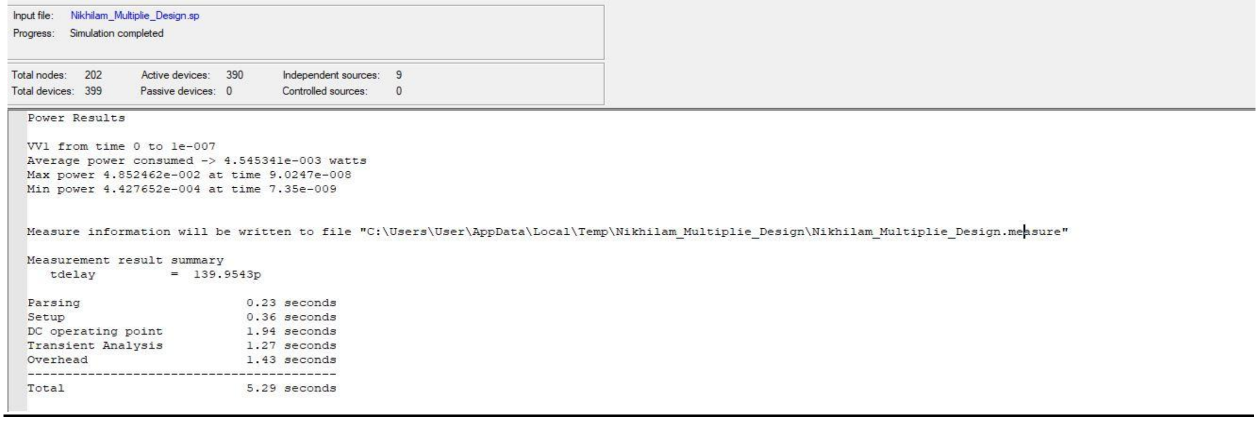
Multiplicand				**Multiplier**				**Multiplication Result**							
a1	**a2**	**a3**	**a4**	**b1**	**b2**	**b3**	**b4**	**S0**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
‘0’	‘1’	‘0’	‘0’	‘0’	‘0’	‘1’	‘0’	‘0’	‘0’	‘0’	‘1’	‘0’	‘0’	‘0’	‘0’

5.5.2 Power and Time Delay of Proposed 4-bit Nikhilam Multiplier

The power and time delay of this multiplier simulated on T-SPICE has been derived as follows:

- Power consumption = 4.545341 mw
- Time Delay= 139.9543 ps

The screenshot of the above result is presented below:



Input file: Nikhilam_Multiplie_Design.sp
Progress: Simulation completed

Total nodes: 202 Active devices: 390 Independent sources: 9
Total devices: 399 Passive devices: 0 Controlled sources: 0

```
Power Results

VV1 from time 0 to 1e-007
Average power consumed -> 4.545341e-003 watts
Max power 4.852462e-002 at time 9.0247e-008
Min power 4.427652e-004 at time 7.35e-009


Measure information will be written to file "C:\Users\User\AppData\Local\Temp\Nikhilam_Multiplie_Design\Nikhilam_Multiplie_Design.measure"

Measurement result summary
   tdelay          =  139.9543p

Parsing                        0.23 seconds
Setup                          0.36 seconds
DC operating point             1.94 seconds
Transient Analysis             1.27 seconds
Overhead                       1.43 seconds
-----------------------------------------
Total                          5.29 seconds
```

Figure 5.6 Power and Time Delay of Nikhilam Multiplier on TSPICE

5.6 COMPARISON OF PROPOSED NIKHILAM MULTIPLIERS WITH MODIFIED UT AND EXISTING ARRAY MULTIPLIER IN TERMS OF POWER DISSIPATION AND DELAY

5.6.1 Comparison Table

The proposed work is compared with 4×4-bit array multipliers with MOD-GDI logic ("P. V. Sridevi" et.al. [2017]. The results are presented in Table 5.4

Table 5.4 Comparison of Results of Various Multipliers in terms of Power and Delay:

Serial No.	Types of Multiplier	Power Dissipation	Time Delay
1.	Existing Array Multiplier	8.030755 (mw)	9.3722 (ns)
2.	Modified UT Multiplier	5.260572 (mw)	221.9566 (ps)
3.	Proposed Nikhilam Multiplier	4.545341 (mw)	139.9543 (ps)

Table 5.5 Percentage Improvement of Various Multipliers in terms of Power and Delay:

% Improvement	Modified UT multiplier over Array multiplier	Proposed Nikhilam multiplier over Array multiplier	Proposed Nikhilam multiplier over Modified UT multiplier
Based on Power Dissipation	34%	43%	14%
Based on Time Delay	97%	99%	37%

5.6.2 Analysis in terms of power dissipation:

- When compared to the UT and Nikhilam multipliers, the overall power consumption of Array multiplier is 8.0307 (mw), which is the highest amongst the three designs.
- When compared to the Array multiplier, the UT Multiplier consumes 5.2605 (mw), which is lower yet higher when compared to the Nikhilam multiplier.
- Finally, when compared to the array and UT multipliers, the proposed Nikhilam multiplier consumes 4.5453 (mw), which is the lowest amongst all three designs.

5.6.3 Analysis in terms of time delay:

- The Array multiplier has a time delay of 9.3722 (ns), which is quite higher than Modified UT and Nikhilam multipliers.
- The UT Multiplier has a time delay of 221.9566 (ps), which is much lower when compared to the Array multiplier but little higher when compared to the Nikhilam multiplier.
- Finally, as compared to the array and UT multipliers, the Nikhilam multiplier requires 139.9543 (ps), which is the lowest among all three designs.

5.7 ANALYSIS OF NIKHILAM MULTIPLIER IN TERMS OF DIFFERENT NEAREST BASE

The performance of the Nikhilam algorithm is investigated in this work for 4-bit length of multiplicands and multipliers. In addition, the performance of the Nikhilam Sutra based multiplier is examined in terms of the number of multiplications and the entire process time. The performance analysis programs are written in MATLAB R2015a (8.5.0.197613) 64-bit. There are the specifications of the computer used for testing - Windows 7 64-bit OS, 4 GB RAM, Intel Core i5-3317U CPU, processor running at 1.70 GHz. The following tables describes the performance of the Nikhilam Sutra based multiplier in terms of the number of multiplications for various nearby bases, i.e. power of ‘10’ bases such as 10, 100, 1000, 10000.

5.7.1 Result Analysis for Base 10

The performance of the Nikhilam Algorithm has been analyzed using MATLAB in terms of various multiplicand and multiplier w.r.t base 10. The detailed analysis of this Nikhilam algorithm has been shown in Table 5.6.

Table 5.6 Analysis of Nikhilam multiplier for Base 10

Parameter	**Less than Base**			**Greater than Base**		
X (Multiplier)	9	5	2	11	15	18
Y (Multiplicand)	8	6	3	12	16	19
b (base)	10	10	10	10	10	10
m=X-b	-1	-5	-8	1	5	8
n= Y-b	-2	-4	-7	2	6	9
C=m*n	2	20	56	2	30	72
D=X+n=Y+m	7	1	-5	13	21	27
Result=b*D+C	72	30	6	132	240	342
Elapsed time(sec)	0.001978	0.002078	0.002160	0.001912	0.001996	0.002076

5.7.2 Result Analysis for Base 100

The performance of the Nikhilam Algorithm has been analyzed using MATLAB in terms of various multiplicand and multiplier w.r.t base 100. The detailed analysis of this Nikhilam algorithm has been shown in Table 5.7.

Table 5.7 Analysis of Nikhilam multiplier for Base 100

Parameter	**Less than Base**			**Greater than Base**		
X (Multiplier)	98	56	23	108	145	188
Y (Multiplicand)	99	62	24	109	156	191
b (base)	100	100	100	100	100	100
m=X-b	-2	-44	-77	8	45	88
n= Y-b	-1	-38	-76	9	56	91
C=m*n	2	1672	5852	72	2520	8008
D=X+n=Y+m	97	18	-53	117	201	279
Result=b*D+C	9702	3472	552	11772	22620	35908
Elapsed time(sec)	0.002401	0.002686	0.002698	0.002298	0.002310	0.002508

5.7.3 Result Analysis for Base 1000

The performance of the Nikhilam Algorithm has been analyzed using MATLAB in terms of various multiplicand and multiplier w.r.t base 1000. The detailed analysis of this Nikhilam algorithm has been shown in Table 5.8.

Table 5.8 Analysis of Nikhilam multiplier for Base 1000

Parameter	Less than Base			Greater than Base		
X (Multiplier)	987	544	221	1030	1677	1895
Y (Multiplicand)	897	476	325	1120	1589	1841
b (base)	1000	1000	1000	1000	1000	1000
m=X-b	-13	-456	-779	30	677	895
n= Y-b	-103	-524	-675	120	589	841
C=m*n	1339	238944	525825	3600	398753	752695
D=X+n=Y+m	884	20	-454	1150	2266	2736
Result=b*D+C	885339	258944	71825	1153600	2664753	3488695
Elapsed time(sec)	0.001896	0.001985	0.002129	0.002234	0.002340	0.002438

5.7.4 Result Analysis for Base 10000

The performance of the Nikhilam Algorithm has been analyzed using MATLAB in terms of various multiplicand and multiplier w.r.t base 10000. The detailed analysis of this Nikhilam algorithm has been shown in Table 5.9.

Table 5.9 Analysis of Nikhilam multiplier for Base 10000

Parameter	Less than Base			Greater than Base		
X (Multiplier)	9893	6540	3245	11070	16843	18320
Y (Multiplicand)	8760	5673	2365	11280	15234	17986
b (base)	10000	10000	10000	10000	10000	10000
m=X-b	-107	-3460	-6755	1070	6843	8320
n= Y-b	-1240	-4327	-7635	1280	5234	7986
C=m*n	132680	14971420	51574425	1369600	35816262	66443520
D=X+n=Y+m	8653	2213	-4390	12350	22077	26306
Result=b*D+C	86662680	37101420	7674425	124869600	256586262	329503520
Elapsed time(sec)	0.001948	0.002012	0.002145	0.002159	0.002256	0.002398

5.8 COMPARATIVE ANALYSIS

According to the results of the study, the Nikhilam Sutra provides the benefit of transforming large digit multiplication into equivalent small digit multiplication. From Table 5.5 it shows that when the value of the multiplier and multiplicand are closer to chosen base, then the required elapsed time is also less. But when the value of the multiplier and multiplicand is taken away from base, then elapsed time also increases. For example, when then value of the multiplier, multiplicand and base are 9, 8, and 10 respectively, then the elapsed time required for this multiplication operation is 1978 µs and when the value of the multiplier, multiplicand and base are 11, 12, and 10 respectively, then the elapsed time required for this multiplication operation is 1912 µs. So, in terms of the total number of multiplications and overall process time, the result of Nikhilam algorithm is better than the classical Array multiplication technique. The reason for this is that the total count of multiplication operations and the required cost of performing multiplication operations are both lower than the classical multiplication technique. Multiplication based on the Nikhilam sutra is ideal for multiplying large numbers. The larger the actual number, the lesser the difficulty of the multiplication.

5.9 DISCUSSIONS

In this chapter, the performance analyses of various 4-bit multipliers such as existing array multiplier, modified UT multiplier and proposed Nikhilam multiplier have been displayed. From the simulated waveforms, it becomes evident that when compared to the array and UT multipliers, the Nikhilam multiplier needs significantly lower power and relatively shorter time delay by using MOD-GDI logic. The performance of the Nikhilam multiplier w.r.t. the closeness of the numbers to the chosen base has also been explored here. The performance investigation is carried out for various multiplier and multiplicand values in terms of different nearby bases in terms of the power of ‘10’ (i.e. 10, 100, 1000, 10000, and so on). The elapsed time is used as an analysis parameter.

CHAPTER 6

CONCLUSION

6.1 CONCLUDING REMARKS

In this thesis, efficient multipliers are proposed based on the Urdhva and Nikhilam sutras of Vedic Mathematics. Design is carried out using MOD-GDI logic and implemented using T-SPICE. We investigated the feasibility of using the Nikhilam sutra of Vedic mathematics to multiply binary numbers and also analyzed the results using MATLAB. The findings of this work are listed below:

- The proposed multiplier is designed using one of the popular Sutras in Vedic Arithmetic called Nikhilam Sutra. The important feature of the Nikhilam sutra is that it converts large digit multiplication to equivalent small digit multiplication. When both the multiplier and the multiplicand are close to the similar base power (radix) power, this sutra is much more effective. The proposed multiplier is designed based on the structure of half adder and full adder by using MOD-GDI logic.

- The existing Array multiplier and UT multiplier are also designed using the same MOD-GDI Logic for the purpose of performance comparison of the proposed design with the existing ones. All designs have used the 6T half adder and 10T full adder cell which found to be more efficient in terms of power consumption, area, and time delay.

- From Table 5.4 it is evident that a 4-bit multiplier developed with Vedic mathematics utilizing UT and Nikhilam sutra is significantly better than the Array Multiplier in terms of power consumption, area, and speed. It indicates that the Nikhilam multiplier has a lower Combinational path delay, takes up less space, and uses less power than the UT multiplier as well as Array multipliers.

- The results show that UT and Nikhilam with modified full adder and half adder using MOD- GDI logic are 97% and 99% faster and also consume 34% and 43% lesser power than that of the existing Array multiplier.

6.2 FUTURE EXTENSIONS

This work has made a sincere attempt to meet its proposed aim. There are several potential directions for future work and here are some proposals that can be considered for further extension.

- This work can be further extended by changing the inherent design by utilizing less number of transistor structures which can reduce power consumption as well as time delay.

- Extending this method to large-digit multiplication and using its features to achieve fast integer multiplication could be a future goal.

- This study can be utilized to develop a 64-bit ALU as well as an integrated multiplier employing these two sutras.

- This work can be extended to design divider block, multiply and accumulate (MAC) unit and integrating them into a Vedic Arithmetic and Logical Unit (ALU).

REFERENCES

- Jagadguru Swami Sri Bharathi Krisna Tirathji, [1986], Vedic Mathematics or Sixteen Simple Sutras from the Vedas, Motilal Banarsidas, Varanasi (India).

- Moumita Ghosh [2007], 'Design And Implementation Of Different Multipliers Using VHDL', Thesis, National Institute of Technology, Rourkela. Available from (http://ethesis.nitrkl.ac.in/66/1/moumita.pdf)

- Saurabh Sunil Bengali [2011], 'Vedic Mathematics and Its Applications in Computer Arithmetic'.(http://repository.lib.ncsu.edu/ir/bitstream/1840.16/7232/1/etd.pdf)

- Abhishek Gupta, Utsav Malviya & Prof. Vinod Kapse [2012], ' A Novel Approach to Design High Speed Arithmetic Logic Unit Based On Ancient Vedic Multiplication Technique', International Journal of Modern Engineering Research (IJMER), vol. 2, no. 4, pp. 2695-2698 ISSN: 2249-6645.

- Kokila, S Ramadhurai, R & Sarah, L [2012], 'VHDL Implementation of Fast 32X32 Multiplier based on Vedic Mathematics', International Journal of Engineering Technology and Computer Applications, vol. 2, no. 1, pp. 46-50.

- Pohokar, S.P Sisal, R.S Gaikwad, M Patil, M.M & Rushikesh Borse [2015], 'Design and Implementation of 16x16 Multiplier using Vedic Mathematics, International Conference on Industrial Instrumentation and Control (ICIC), pp. 1174-1177.

- Jin Hyuk kim, je-Huk Ryu & jun Dong cho [2000], 'A high speed and low power VLSI Multiplier using a redundant binary booth encoding'. Avialble from (http://citeseerx.ist.psu.edu/viewdoc/summary?doi:10.1.1.46.9104)

- Morgenshtein A, Fish A and Wagner I A [2002] , “Gate-Diffusion Input (GDI) – A power efficient method for digital combinatorial circuits”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 10, no. 5, Oct., pp. 566-581.

- Lee P M, Hsu C H and Hung Y H [2007], “Novel 10-T full adders realized by GDI structure”, In Proc. of International Symposium on Integrated Circuits, Singapore, pp. 115-118.

- Moradi F, Wisland D T, Mahmoodi H, Aunet S, Cao T V and Peiravi A [2009], “Ultra low power full adder topologies”, In Proc. of IEEE International Symposium on Circuits and Systems, pp. 3158-3161.

- Dan Wang, Maofeng Yang, Wu Cheng, Xuguang Guan, Zhangming Zhu and Yintang Yang [2009], "Novel low power full adder cells in 1 0nm CMOS technology", In Proc. of IEEE Conference on Industrial Electronics and Applications, pp. 430-433.

- Uma R and Dhavachelvan P [2012], "Modified gate diffusion input technique: a new technique for enhancing performance in full adder circuits", In Proc. of International Conference on Communication, Computing and Security, pp. 74-81.

- Shrivas J, Akashe S and Tiwari N [2012] , "Design and performance analysis of 1 bit full adder using GDI technique in nanometer era", In Proc. of World Congress on Information and Communication Technologies, pp. 822-825.

- Dhar K [2014] , "Design of a low power, high speed, energy efficient full adder using modified GDI and MVT scheme in 45nm technology", In Proc. of International Conference on Control, Instrumentation, Communication and Computational Technologies, pp. 36-41.

- Archana S and Durga G [2014], "Design of low power and high speed ripple carry adder", In Proc. of IEEE International Conference on Communications and Signal Processing, pp. 939-943.

- Morgenshtein A, Shwartz I and Fish A [2014] , "Full swing Gate Diffusion Input (GDI) logic – case study for low power CLA adder design", Integration, the VLSI Journal, vol. 4 , no. 1, Jan., pp. 62-70.

- Shinde K D and Nidagundi J C [2014], "Design of fast and efficient 1- bit full adder and its performance analysis", In Proc. of International Conference on Control, Instrumentation, Communication and Computational Technologies, pp.1275-1279.

- Foroutan, V, Teheri M, Navi K and Mazreah A [2014], "Design of two low power full adder using GDI structure and hybrid CMOS logic style", Integration, the VLSI Journal, vol. 4 , no.1, Jan., pp. 48-61.

- Soundharya M and Arunkumar R [2015], "GDI based area delay power efficient carry select adder", In Proc. of International Conference on Green Engineering and Technologies, pp. 1-5.

- Dhar K, Chatterjee A and Chatterjee S [2014], "Design of an energy efficient, high speed, low power full subtractor using GDI technique", In Proc. of IEEE Students Technology Symposium, pp. 199-204.

- Singh H and Kumar R [2014] , "10-T Full subtraction Logic Using GDI Technique", In Proc. of International Conference on Computational Intelligence and Communication Networks, pp. 956-960.

- Gupta J, Grover A, Wadhwa G K and Grover N [2013], “Multipliers using low power adder cells using 1 0nm technology”, In Proc. Of International Symposium on Computational and Business Intelligence, pp.3-6.

- Reddy B N M, Sheshagiri H N, Vijayakumar B R and Santhala S [2014], “Implementation of low Power -Bit multiplier using gate diffusion input logic”, In Proc. of IEEE International Conference on Computational Science and Engineering, pp. 1868-1871.

- Saberkari A, Shokouhi S B, Kiani A and Poorahangaryan F [2009], “A novel low power static frequency divider based on the GDI technique”, In Proc. of Canadian Conference on Electrical and Computer Engineering, pp. 67-70.

- Khurana S, Grover A and Grover N [2013], “Comparative analysis: power reversible comparator circuits in 0 nm technology”, In Proc. of Asia Modeling Symposium, pp. 103-107.

- Sharma A and Sharma P [2014], “Area and power efficient 4-bit comparator design by using 1-bit full adder module”, In Proc. of International Conference on Parallel, Distributed and Grid Computing pp. 1-6.

- Shekhawat V, Sharma T and Sharma K G [2014], “2-Bit magnitude comparator using GDI technique”, In Proc. of International Conference on Recent Advances and Innovations in Engineering, pp. 1-5.

- Dubey V and Sairam R [2014], “An Arithmetic and Logic Unit (ALU) optimized for area and power”, In Proc. of IEEE International Conference on Advanced Computing and Communication Technologies, pp. 330-334.

- Morgenshtein A, Fish A and Wagner I A [2004], “An efficient implementation of D-flip-flop using the GDI technique”, In Proc. of International Symposium on Circuits and Systems, pp. 673-676.

- Fisher S, Teman A, Vaysman D, Gertsman A, Yadid-Pecht O and Fish A [2009], “Ultra-low power subthreshold flip-flop design”, In Proc. of International Symposium on Circuits and Systems, pp. 1573-1576.

- Swami N, Arora N and Singh B P [2011], “Low Power subthreshold D flip flop”, In Proc. of International Conference on Devices and Communications, pp. 1-4.

- Abiri E, Salehi M R and Darabi A [2014], “Design and simulation of low-power and high speed T-Flip Flap with the modified gate diffusion input technique in nano process”, In Proc. of Iranian Conference on Electrical Engineering, pp. 82-87.

- Dhar K [2014], “Design of a high speed, low power synchronously clocked NOR-based JK flip-flop using modified GDI technique in 4 nm technology”, In Proc. of International Conference on Advances in Computing, Communications and Informatics, pp. 600-606.

- Magesh Kannan P and Prathyusha K [2011], “Implementation of low power RAM in GDI technique with full swing”, In Proc. of International Conference on Signal Processing, Communication, Computing and Networking Technologies, pp. 592-597.

- Hari O P and Mai A K [2011], “Low power and area efficient implementation of N-phase non overlapping clock generator using GDI technique”, In Proc. of International Conference on Electronics Computer Technology, pp. 123-127.

- K. S. Gurumurthy, M. S Prahalad, “Fast and power-efficient 16×16 Array of Array multiplier using Vedic Multiplication”, Proceedings of IEEE 5th International Microsystems Packaging Assembly and Circuits Technology Conference: Taipei, Taiwan, pp-1-4, [2010].

- Sumit Vaidya and Deepak Dandekar, “Delay-Power Performance Comparison of Multipliers in VLSI Circuit Design”, International Journal of Computer Networks & Communications, Vol. 2, No.4, pp.47-56, [2010].

- Chitralekha Mehera [2012], Computing Technique in Ancient India‘, book chapter Bio inspired computing and applications, 7th International Conference on Intelligent Computing, ICIC 2011, Zhengzhou, China, Springer, pp. 282-289.

- Pradhan M, Panda R, Sahu, S. K., [2011], “Speed Comparison of 16X16 Vedic Multipliers”, International Journal of computer applications, 21(6), pp. 16-19.

- Jaina D, Sethi K and Panda R [2011], “Vedic mathematics based multiply accumulate unit”, In Proc. of the International Conference on Computational Intelligence and Communication Networks, pp.754-757.

- Charishma, Ganesh Kumar, G & V [2012], 'Design of High Speed Vedic Multiplier using Vedic Mathematics Techniques', International Journal of Scientific and Research Publications, vol. 2, no. 3, pp. 1-5.

- Ramachandran.S and Kirti.S.Pande, “Design and Implementation Analysis of an Integrated Vedic Multiplier Architecture”, International Journal of Computational Engineering Research, ISSN: 2250–3005, Vol. 2, Issue.3, pp. 697-703, [2012].

- Sree Nivas, A & Kayalvizhi, N [2012], Implementation of Power Efficient Vedic Multiplier‘, International Journal of Computer Applications, vol. 43, no. 16, pp. 21-24.

- Harish Kumar, Ch [2013], 'Implementation and analysis of power, area, delay of Array, Urdhava, Nikhilam Vedic multipliers', International Journal of Scientific and Research Publication, ISSN 2250-3153, vol. 3, no. 1, pp. 1-5.

- Somani, Dheeraj Jain, Sanjay Jaiswal, Kumkum Verma and Swati Kasht, “Compare Vedic Multipliers with Conventional Hierarchical array of array multiplier”, International Journal of Computer Technology and Electronics Engineering, Vol. 2, Issue.6, pp.52-55, [2012].

- Virendra Magar, “Area and Speed-wise superior Vedic multiplier for FPGA based arithmetic circuits”, International Journal of Computational Engineering Research, Vol.03, Issue. 5, pp.44-49, [2013].

- Premananda, B. S., Samarth S. Pai, B. Shashank, and Shashank S. Bhat. "Design and implementation of 8-bit Vedic multiplier." International Journal of Advanced Research in Electrical, Electronics, and Instrumentation Engineering 2, no.12 pp. 5877-5882, [2013].

- Goyal & Shamim Akhter [2013], 'An advancement in the N×N Multiplier Architecture Realization via the Ancient Indian Vedic Mathematic', International Journal of Computer Applications (0975 – 8887), vol. 127, no. 2, pp. 24-27.

- Gokhale.G.R & Bahingonde P.D [2015], 'Design of Vedic Multiplier using area efficient carry select adder', International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, pp. 576-581.

- Itawadiya, A. K., Mahle, R., Patel, V., Kumar, D., [2014], “Design a DSP Operations using Vedic Mathematics", In Proc. of IEEE Conference ICCSP-13(Tamil Nadu,India),pp 897-902.

- Kayal, D Mostafa, P Dandapat, A & Sarkar, C.K [2014] , 'Design of High Performance 8 bit Multiplier using Vedic Multiplication Algorithm with McCMOS Technique', Journal of Signal Processing and Systems, vol. 76, no. 1, pp. 1-9.

- Banu Priya N, Shanmugapriya K, Vinitha R and Gokilavan. M, “Design and Comparison of Vedic Multiplier”, International Journal for Advance Research in Engineering and Technology, ISSN 2320-6802, pp. 31-35, [2014].

- P. Tuwanuti and N. Thongbai, "Implementation of Vedic multiplier technique on multicore processor," TENCON 2014 - 2014 IEEE Region 10 Conference, [2014], pp. 1-6, doi:10.1109/TENCON.2014.7022325.

- Jain, S. Pancholi, M.Garg, H., Saini, S., [2014], “Binary Division Algorithm and High Speed Deconvolution Algorithm (Based on Ancient Indian Vedic Mathematics)”, In Proc. of 11th IEEE Conference ECTICON-2014 (Nakhon Ratchasima, Thailand), pp 1-5.

- Angshuman Khan & Rupayan Das [2015], ‗Novel Approach of Multiplier Design using ancient Vedic Mathematics‘: proceedings of Second International Conference on

Information systems Design and Intelligent Applications (INDIA-2015), Springer series of Advances in Intelligent Systems and Computing (AISC), pp. 265-272.

- Tadas, A., Rotake, D., [2015], “64 Bit Divider using Vedic Mathematics", In Proc. of IEEE Conference on International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (Chennai, T.N., India), pp-317-320.

- Ram G.C, Rani D.S, Balasaikesava R., Sindhuri K.B. [2016], “VLSI Architecture for delay efficient 32-bit Multiplier using Vedic Mathematic sutras", In Proc. of IEEE Conference RTEICT-16(Bangalore, India), pp. 1873-1877.

- Pinninti Kishore, P. V. Sridevi, K. Babulu [2017], “Low Power and High Speed Optimized 4-bit Array Multiplier using MOD-GDI Technique”, IEEE 7th International Advance Computing Conference.

- Bilanchi V., Munari I. D., [2020], “A Modular Vedic Multiplier Architecture for Model-Based Design and Deployment on FPGA Platforms”, Microprocessor and Microsystems, Elsevier, pp. 1-9.

- Manikadan S.K [2017] “Analysis and Design of Vedic Multipliers for Nikhilam Sutra with Reduced Bit Size using Karatsuba Algorithm”.

- Pankaj Verma [2013] “Modified GDI Technique – A Power Efficient Method for Digital Circuit Design”.

DESIGN AND ANALYSIS OF HIGH SPEED POWER EFFICIENT MOD-GDI LOGIC BASED 4-BIT VEDIC MULTIPLIER USING NIKHILAM SUTRA

ORIGINALITY REPORT

6%
SIMILARITY INDEX

PRIMARY SOURCES

1	arresearchpublication.com Internet	114 words —	1%
2	eprints.umm.ac.id Internet	98 words —	1%
3	patents.justia.com Internet	87 words —	1%
4	www.yumpu.com Internet	68 words —	< 1%
5	eprints-bangaloreuniversity.in Internet	56 words —	< 1%
6	www.scribd.com Internet	46 words —	< 1%
7	www.erpublication.org Internet	43 words —	< 1%
8	ijasret.com Internet	36 words —	< 1%
9	citeseerx.ist.psu.edu Internet	34 words —	< 1%

10	Kuheli Dutta, Sudipta Chattopadhyay, Vaishna Biswas, Suman Roy Ghatak. "Design of Power Efficient Vedic Multiplier using Adiabatic Logic", 2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2019 Crossref	33 words — < 1%
11	ijarcce.com Internet	33 words — < 1%
12	d-nb.info Internet	23 words — < 1%
13	electronics.etfbl.net Internet	23 words — < 1%
14	www.ijecse.org Internet	23 words — < 1%
15	Satya Ranjan Sahu, Bandan Kumar Bhoi, Manoranjan Pradhan. "Fast signed multiplier using Vedic Nikhilam algorithm", IET Circuits, Devices & Systems, 2020 Crossref	19 words — < 1%
16	www.worldwidejournals.com Internet	19 words — < 1%
17	open.metu.edu.tr Internet	17 words — < 1%
18	docplayer.gr Internet	16 words — < 1%
19	ijcset.com Internet	16 words — < 1%

20	www.ijctee.org Internet	16 words — < 1%
21	www.ijert.org Internet	15 words — < 1%
22	www.researchgate.net Internet	15 words — < 1%
23	M. Anitha, J.Princy Joice, Rexlin Sheeba.I. "A New-High Speed-Low Power-Carry Select Adder Using Modified GDI Technique", International Journal of Reconfigurable and Embedded Systems (IJRES), 2015 Crossref	14 words — < 1%
24	scholar.it.kmitl.ac.th Internet	14 words — < 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 14 WORDS
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	< 14 WORDS