

ANALYSIS OF RECONFIGURABLE ENERGY EFFICIENT APPROXIMATE ARITHMETIC ARCHITECTURE

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

by

CHILLAKURU SAI VINAY REDDY (37130083)

DATLA AVINASH VARMA (37130103)



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

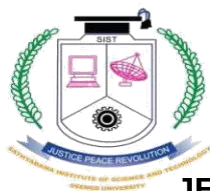
SATHYABAMA

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

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL – 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with “A” grade by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119
www.sathyabama.ac.in

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of CHILLAKURU SAI VINAY REDDY (Reg.No:37130083) and DATLA AVINASH VARMA (Reg.No:37130103), who carried out the project entitled “**ANALYSIS OF RECONFIGURABLE ENERGY EFFICIENT APPROXIMATE ARITHMETIC ARCHITECTURE**” under my supervision from November 2020 to March 2021.

Internal Guide & Head of the Department
Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **CHILLAKURU SAI VINAY REDDY (Reg.No:37130083)** and **DATLA AVINASH VARMA (Reg.No:37130103)**, hereby declare that the Project Report entitled “**ANALYSIS OF RECONFIGURABLE ENERGY EFFICIENT APPROXIMATE ARITHMETIC ARCHITECTURE**” done by us under the guidance of **Dr. T. RAVI, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of the Bachelor of Engineering degree in Electronics and Communication Engineering.

1)

2)

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

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

We convey our thanks to **Dr. N.M. NANDHITHA, M.E., Ph.D., Dean, School of Electrical and Electronics** and **Dr. T. RAVI, M.E., Ph.D., Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. T. RAVI, M.E., Ph.D., Head of the Department, Department of Communication and Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

We express our gratitude to our parents for their constant encouragement and support for the completion of the project.

ABSTRACT

Sequential Circuits are extensively used in the design of memory elements such as flip-flops and registers. Registers are used as data storage devices in low power VLSI designs. A shift register consists of a chain of flip-flops in cascade, with the output of one flip-flop connected. To achieve the high performance in universal shift register, flip-flops play the important role. A Universal Shift Register (USR) is a type of register which is a combination of unidirectional, bidirectional, and parallel loading register. The mode of operation switches with the select lines of the design. The design and testing of the Universal Shift Register will be carried out in Hspice tool. The universal shift register using edge triggered D flip flop with low power and area efficient applications. The D flip-flop is realized using minimum number of transistors hence reducing the area and is easy to implement. The aim is to evaluate and analyze the performance parameters of universal shift register so that it can further be utilized in complex VLSI designs to achieve better performances.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
No.		No.
	ABSTRACT	V
	LIST OF FIGURES	IX
	LIST OF TABLES	X
1	INTRODUCTION	1
	1.1 MOTIVATIONS	2
	1.2 LOGIC GATES	3
	1.2.1 NAND GATE	4
	1.2.2 AND GATE	5
	1.2.3 NOR GATE	6
	1.2.4 OR GATE	6
	1.2.5 NOT gate & XNOR gate	7
	1.3 MULTIPLEXER	8
	1.3.1 2×1 Multiplexer	9
	1.3.2 4×1 Multiplexer	10
	1.4 ORAGANIZATION OF THE REPORT	12
2	LITERATURE SURVEY AND OBJECTIVES	
	2.1 LITERATURE SURVEY	13
	2.2 LITERATURE SUMMARY	14
	2.3 PROBLEM DEFINITION	14
	2.4 OBJECTIVE OF THE PROJECT	15

3	AIM AND SCOPE OF PRESENT INVESTIGATION	
	3.1 PROPOSED METHODOLOGY	16
	3.2 AREA OF PROPOSED WORK AND APPLICATION	16
	3.3 FULL SUBTRACTOR	16
	3.4 4×1 MULTIPLEXER	18
	3.5 EXISTING RECONFIGURABLE FULL SUBTRACTOR	20
	3.6 PROPOSED RECONFIGURABLE CIRCUITS	22
4	PROPOSED RECONFIGURABLE CIRCUITS	
	4.1 RECONFIGURABLE RIPPLE BORROW SUBTRACTOR	23
	4.2 RECONFIGURABLE FULL ADDER	24
	4.3 RECONFIGURABLE RIPPLE CARRY ADDER	26
5	RESULT AND DISSCUSION	
	5.1 EXISTING FULL SUBTRACTOR SIMULATION OUTPUT	28
	5.2 PROPOSED RIPPLE BORROW SUBTRACTOR SIMULATION OUTPUT	29
	5.3 PROPOSED RECONFIGURABLE FULL ADDER SIMULATION OUTPUT	30
	5.4 RECONFIGURABLE RIPPLE CARRY ADDER	31
	5.5 PERFORMANCE ANALYSIS	
	5.5.1 Power and Delay Analysis of Existing System	32
	5.5.2 Power and Delay Analysis (simulation graphs)	33
	5.5.3 Power and Delay Analysis of Proposed Reconfigurable circuits	34

6	CONCLUSION AND FUTURE SCOPE	35
	REFERENCES	36

LIST OF FIGURES

FIGURE.NO.	DESCRIPTION	PAGE NO.
1.1	NAND gate	4
1.2	AND gate	5
1.3	NOR gate	6
1.4	OR gate	7
1.5	Symbolic Diagram of Multiplexer	9
1.6	Logic Diagram of 2x1 Multiplexer	10
1.7	Logic Diagram of 4x1 Multiplexer	11
3.1	Block Diagram of Full Subtractor	17
3.2	Block Diagram of 4×1 Multiplexer	19
3.3	reconfigurable full subtractor	20
3.4	Block Diagram of traditional full subtractor	21
4.1	ripple borrow subtractor	24
4.2	Existing full adder circuit	25
4.3	Ripple carry adder	27
5.1	Output Waveform for existing full subtractor	28
5.2	Output Waveform for proposed ripple borrow subtractor	29
5.3	Output Waveform for Full adder	30
5.4	Output Waveform for ripple carry adder	31

LIST OF TABLES

TABLE NO.	DESCRIPTION	PAGE NO.
1.1	Truth table of 4×1 Multiplexer	11
3.1	Truth Table of Full subtractor	17
3.2	Truth Table of 4×1 Multiplexer	19
3.3	Truth Table of Full subtractor	22
5.1	Design analysis of existing circuit	32
5.2	Design Analysis of proposed circuits	34

CHAPTER 1

INTRODUCTION

VLSI is the process of integrating thousands of transistors into a single chip. VLSI design is mainly used to minimize the interconnecting fabrics area. The main factors which limit the development of smaller and more complex IC chip are the IC fabrication technology, designer productivity and the cost. Depending on the application, different performance aspects become important. The Arithmetic Logic Unit (ALU) could be an elementary building block of the Central process Unit (CPU) of a laptop. Even one of the simplest microprocessor contains one ALU for purposes such as maintaining timers.

ALU is a core component of all central processing unit within in a computer and is an integral part of the execution unit. ALU is capable of calculating the results of a wide variety of basic arithmetical and logical computations. The ALU takes as input the info to be operated on (called operands) and a code from the management unit indicating that operation to perform. The output is the result of the computation. The ALU implemented will perform the following operations: Arithmetic operations (addition, subtraction, increment, decrement, transfer). Logic operations (AND, NOT, OR, NAND, NOR, EX-OR, EX-NOR).

A digital system can be represented at different levels of abstraction. This keeps the description and design of complex systems manageable. This keeps the outline and style of advanced systems manageable. the best level of abstraction is that the activity level that describes a system in terms of what it will (or however it behaves) instead of in terms of its parts and interconnection between them. The behavioural style makes use of a process statement. A process statement is the main construct in behavioural modelling that allows using sequential statements to describe the behaviour of a system over time. Process is declared within architecture and is a concurrent statement.

In general, external circuitry controls an ALU by applying signals to its inputs. Typically, the external circuitry employs sequential logic to control the ALU

operation, which is paced by a clock signal of a sufficiently low frequency to ensure enough time for the ALU outputs to settle under worst-case conditions proportional to the number of flip flops, as each flip flop is capable of storing only one bit at a time. A flip flop is an electronic circuit with two stable states that can be used to store binary data. Flip-flops and latches are fundamental building blocks of digital electronics systems. The stored data can be changed by applying varying inputs. When the flip-flops in a register are connected in such a way that the output of one flip flop, becomes the input of the other, a shift register is created.

The CPU begins an ALU addition operation by routing operands from their sources (which are usually registers) to the ALU's operand inputs, while the control units simultaneously apply a value to the ALU's opcode input, configuring it to perform addition. At the same time, the CPU also routes the ALU result output to a destination register that will receive the sum. The ALU's input signals, which are held stable until the next clock, are allowed to propagate through the ALU and to the destination register while the CPU waits for the next clock. When the next clock arrives, the destination register stores the ALU result and, since the ALU operation has completed, the ALU inputs may be set up for the next ALU operation.

1.1 MOTIVATIONS

Energy potency has emerged mutually of the foremost essential style parameters in up to date computer system style. Approximate computing could be a new computing paradigm to attain energy potency by trading- off energy/area/latency enhancements with accuracy for error-resilient applications. There is a persistent demand for energy-efficient yet compute- intensive designs for modern applications such as multimedia, digital signal processing (DSP), and machine learning-based applications. To achieve transformational energy reductions the improvements in technology nodes, voltage scaling, or designing parallel architectures are insufficient. There is an urgent need to promote alternative computing techniques to cope with high energy requirements for emerging applications.

Out of four basic arithmetic operations, the division operation is performed infrequently while its power consumption, latency, and complexity are greater as compared to other arithmetic operations. Approximate multiplier design components like adders and compressor have been simplified to be used in approximate multiplier circuit. Division operation is used in many of the image/video processing tasks such as compression, pixel division, foreground extraction, etc. However, as the use of digital image/video and signal processing techniques becomes more prevalent, the importance of arithmetic operation like division is gaining momentum.

There is a persistent demand for energy-efficient yet compute intensive designs for modern applications such as multimedia, digital signal processing (DSP), and machine learning-based applications. To achieve transformational energy reductions the improvements in technology nodes, voltage scaling, or designing parallel architectures are insufficient.

One class of techniques which can be applied in such scenario is based on this. However, using approximate computing techniques results in a trade-off between circuit parameters (area, delay, and power) with the output accuracy. The computational quality as well as the available energy requirements of an application varies at runtime.

1.2 LOGIC GATES

A logic gate is an idealized model of computation or physical electronic device implementing a Boolean function, a logical operation performed on one or more binary inputs that produces a single binary output. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses. An n-bit shift register can be formed by connecting n flip-flops where each flip flop stores a single bit of data. A register can be defined as when a set of FFs can be connected within the series, the definition of the shift register is when the stored data can be moved in the registers. It is a sequential circuit, mainly used to store include such devices as multiplexers, registers, arithmetic logic units (ALUs), and computer memory, all the way up through

complete microprocessors, which may contain more than 100 million gates. In modern practice, most gates are made from MOSFETs (metal–oxide–semiconductor field effect transistors).

A. Types of Logic Gates

Logic gates are of many types majorly by their mode of operation. NAND and NOR gates are known as universal gates, because using those we can design any other logic circuits. There are six (6) basic types of logic gates which are listed below.

- NAND gate – (not AND)
- AND gate
- NOR gate – (not OR)
- OR gate
- XOR gate
- NOT gate

1.2.1 NAND GATE

It is a logic gate which produces an output which is false only if all its inputs are true; thus, its output is complement to that of an AND gate. The output will be LOW (0) only if all the inputs are HIGH; if any input is LOW (0), a HIGH (1) output results. A NAND gate is made using transistors and junction diodes. The NAND gate has the property of functional completeness, which it shares with the NOR gate. That is, any other logic function (AND, OR, etc.) can be implemented using only NAND gates.



Fig:1.1: NAND gate

There are two inputs and one output basically, it is also one of the universal gates. It is simply an opposite to AND gate and the function is also the opposite to the AND. Hence, this the brief explanation about NAND gate.

1.2.2 AND GATE

The second type of logic gate we will be considering is the Serial in – Parallel out shift register. The **AND gate** is a basic digital logic gate that implements logical conjunction. The output is HIGH (1) only if all the inputs to the AND gate are HIGH (1). If none or not all inputs to the AND gate are HIGH, LOW output results. An AND gate can be designed using only N-channel (pictured) or P-channel MOSFETs, but is usually implemented with both (CMOS). The digital inputs **a** and **b** cause the output to have the same result as the AND function. The function can be extended to any number of inputs. It is simply an opposite to NAND gate and the function is also the opposite to the NAND.

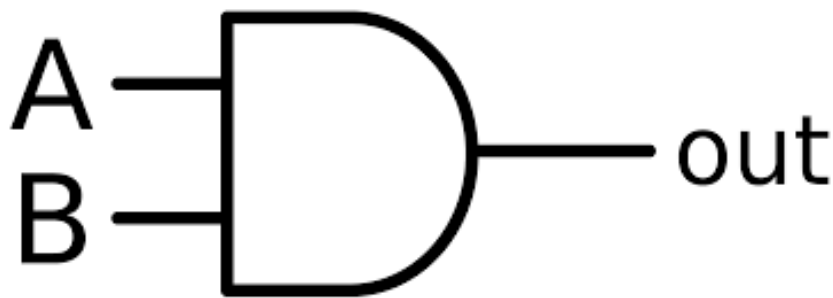


Fig:1.2: AND gate

The function can be extended to any number of inputs. It is simply an opposite to NAND gate and the function is also the opposite to the NAND. Hence, this the brief explanation about AND gate.

1.2.3 NOR GATE

The **NOR gate** could be a digital computer circuit that implements logical NOR it behaves per the reality table to the proper. A HIGH output (1) results if each the inputs to the gate area unit LOW (0); if one or every input is HIGH (1), a output (0) result. neither is the results of the negation of the OR operator. It also can in some senses be seen because the inverse of AN AND circuit. neither is a operationally complete operation—NOR gates will be combined to get the other

logical function. This property is shared with NAND gate. against this, the OR operator is monotonic because it will solely modification LOW to HIGH however not contrariwise.

In most, but not all, circuit implementations, the negation comes for free—including CMOS and TTL. In such logic families, OR is the more complicated operation; it may use a NOR followed by a NOT. NOR Gates are basic logic gates, and as such they are recognised in TTL and CMOS IC'S. Hence, this the brief explanation about AND gate.

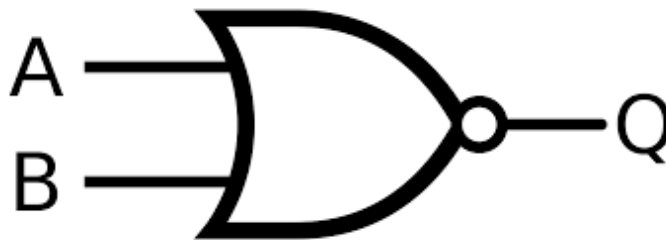


Fig:1.3: NOR gate

1.2.4 OR GATE

The **OR gate** is a digital logic gate that implements logical disjunction – it acts according to the truth table of it. The output is HIGH if one or both of the inputs are HIGH. If neither input is high, a LOW output results. In another sense, the function of OR effectively finds the maximum between two binary digits, just as the complementary AND function finds the minimum. OR gates are basic logic gates, and are available in TTL and CMOS ICs logic families. The standard 4000 series CMOS IC is the 4071, which includes four independent two-input OR gates. The TTL device is the 7432. In addition to the standard 2-input OR gate, 3- and 4-input OR gates are also available. The logic circuit given below in figure 1.4[12] shows the OR gate.

With active low open collector logic outputs, as used for control signals in many circuits, an OR function can be produced by wiring together several outputs. This arrangement is called a wired OR. This implementation of an OR function

typically is also found in integrated circuits of N or P-type only transistor processes. Hence, this is the brief explanation about OR gate.



Fig:1.4: OR gate
(Source [12])

1.2.5 NOT gate & XNOR gate

XOR gate is a digital logic gate that gives a true (1 or HIGH) output when the number of true inputs is odd. An XOR gate implements an exclusive or; that is, a true output results if one, and only one, of the inputs to the gate is true. If both inputs are false (0/LOW) or both are true, a false output result. XOR represents the inequality function, i.e., the output is true if the inputs are not alike otherwise the output is false. Whereas, an inverter circuit outputs a voltage representing the opposite logic-level to its input. Its main function is to invert the input signal applied. If the applied input is low then the output becomes high and vice versa. Inverters can be constructed using a single NMOS transistor or a single PMOS transistor coupled with a resistor. Since this 'resistive-drain' approach uses only a single type of transistor, it can be fabricated at a low cost.

B. Applications of Logic gates:

Logic gates are used to develop digital electronic circuits like

- Microprocessor
- Microcontroller
- Embedded systems
- And also, in various electronic and electrical circuits.

A logic gate is an idealized model of computation or physical electronic device implementing a Boolean function, a logical operation performed on one or

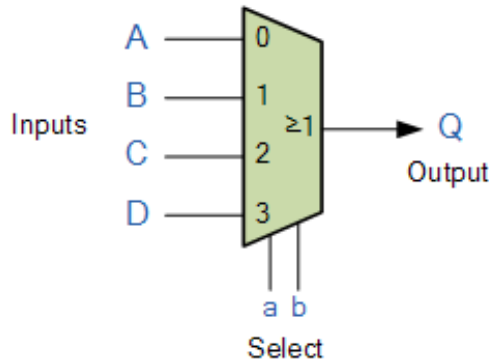
more binary inputs that produces a single binary output. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses. An n-bit shift register can be formed by connecting n flip-flops where each flip flop stores a single bit of data. A register can be defined as when a set of FFs can be connected within the series, the definition of the shift register is when the stored data can be moved in the registers. It is a sequential circuit, mainly used to store include such devices as multiplexers, registers, arithmetic logic units (ALUs), and computer memory, all the way up through complete microprocessors, which may contain more than 100 million gates. In modern practice, most gates are made from MOSFETs (metal–oxide–semiconductor field effect transistors).

1.3 MULTIPLEXER

Multiplexing is the generic term used to describe the operation of sending one or more analogue or digital signals over a common transmission line at different times or speeds and as such, the device we use to do just that is called a Multiplexer. The multiplexer, shortened to “MUX” or “MPX”, is a digital switch, also called as data selector. It is a combinational circuit with more than one input line, one output line and more than one select line. It allows the binary information from several input lines or sources and depending on the set of select lines, particular input line, is routed onto a single output line. The basic idea of multiplexing is shown below in figure 1.5 which data from several sources are routed to the single output line when the enable switch is ON. That is how the multiplexers are also called as ‘many to one’ combinational circuits. If there are m selection lines, then the number of possible input lines is 2^m . Alternatively we can say that if the number of input lines is equal to 2^m , then m selection lines are required to select one of n (consider $2^m = n$) input lines.

The multiplexer is a combinational logic circuit designed to transfer many input lines to a single-output line. Multiplexers act like very fast multiple position rotary switches which link or control multiple input lines called "channels" one at a time to the output. Multiplexers are not restricted to simply converting a variety of separate input lines or channels to one single standard output. There are also

forms that can transfer their inputs to several outputs and have 4-to-2,8-to-3 or even 16-to-4 arrangements etc.



**Fig:1.5: Symbolic Diagram of Multiplexer
(Source [13])**

Applications of Multiplexers are as follows, a multiplexer is used in numerous applications like, where multiple data can be transmitted using a single line. Communication System – multiplexer is used in communication systems, which has a transmission system and also a communication network. A Multiplexer is used to increase the efficiency of the communication system by allowing the transmission of data such as audio & video data from different channels via cables and single lines. Computer Memory – multiplexer is used in computer memory to keep up a vast amount of memory in the computers, and also to decrease the number of copper lines necessary to connect the memory to other parts of the computer. Telephone Network – multiplexer is used in telephone networks to integrate the multiple audio signals on a single line of transmission. Transmission from the Computer System of a Satellite: multiplexer is used to transmit the data signals from the computer system of a satellite to the ground system by using a GSM communication.

1.3.1 2×1 Multiplexer

A 2×1 multiplexer consists of two I0 and I1 inputs, one selects S input and one Select Y output. The output is connected to one of the inputs, depending on the select signal. Since there are two input signals, it is possible to link the inputs to the outputs only in two ways, so one selection is sufficient to do these

operations. If the select line is low, the output is switched to input I0, while if the select line is high, the output is switched to input I1. The following figure 1.6[13] shows the block diagram of a 2×1 multiplexer that connects two 1-bit inputs to a common destination.

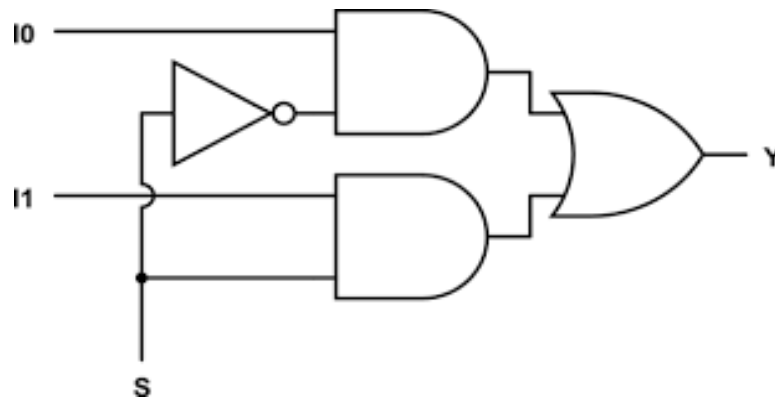


Fig:1.6: Logic Diagram of 2x1 Multiplexer

(Source [13])

1.3.2 4×1 Multiplexer

A 4×1 multiplexer consists of four data input lines which are from D0 to D3, two select lines S0 and S1 respectively, and one single output line from Y. To connect the output line, the select lines S1 and S2 choose one of four input lines. On select lines the specific input combination selects one of the inputs (D0 through D3) to the output. A 4×1 multiplexer's truth table is shown below in table 1.1, in which four input combinations 00, 10, 01 and 11 on the select lines transfer the inputs D0, D2, D1 and D3 to the output respectively. That means the output at Y is D0 when S1=0 and S0=0, similarly Y is D1 when the select inputs S1=0 and S0=1 and so on.

Table 1.1: Truth Table of 4×1 Multiplexer

Select Data Inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

The following figure 1.7[13] illustrates the block diagram of a 4-to-1 multiplexer in which the multiplexer decodes the input by select line.

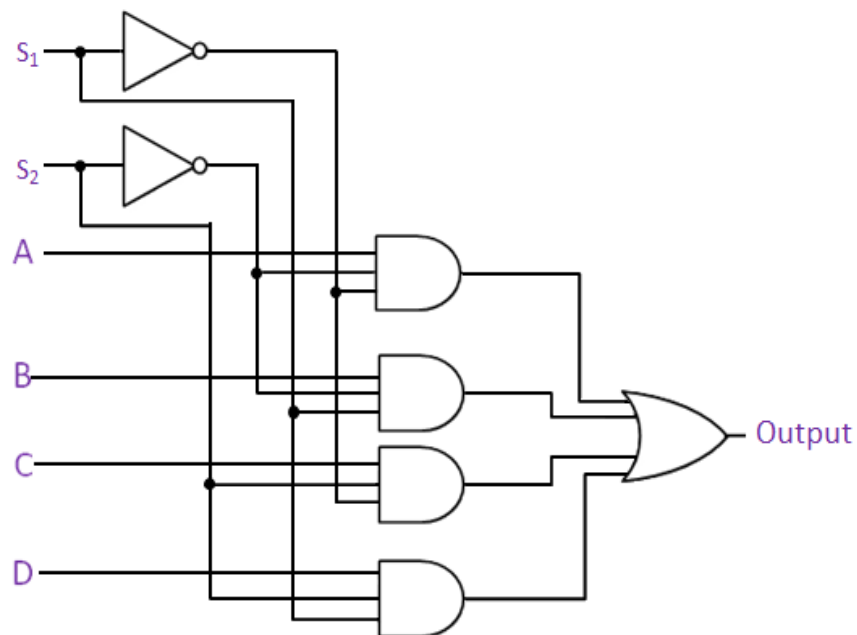


Fig:1.7: Logic Diagram of 4x1 Multiplexer

(Source [13])

1.4 ORGANIZATION OF THE REPORT

The project report is organized into six chapters

Chapter 1 Discusses the background of VLSI, motivation, introduction of arithmetic logic unit (ALU), logic gates. It also discusses about the various multiplexers and organization of the project work.

Chapter 2 Delivers detailed literature survey related to the project, literature summary, problem definition, objectives of the project.

Chapter 3 Describes the working and existing techniques of reconfigurable full subtractor.

Chapter 4 Discusses the proposed energy efficient reconfigurable adder, ripple carry adder and ripple borrow subtractor to minimise the delay.

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

Chapter 6 Provides the conclusion project work. It also gives the schema of the future work that is how the project could be enhanced further. Finally, the references referred to lead the project are given in brief.

CHAPTER 2

LITERATURE SURVEY AND OBJECTIVES

This chapter discusses the about the literature survey and summary. It also gives a detailed idea on the existing reconfigurable circuits and how their structures are synthesised.

2.1 LITERATURE SURVEY

Suganthi Venkatachalam et al. (2019) evaluated the performance metrices of two approximate restoring dividers AD-M1 and AD-M2 are proposed. AD-M1 introduces approximation to the p least-significant columns by replacing p $(p - 1)/2$ exact cells with approximate cells. The approximate cells have simplified circuitry which therefore allows for reduction of area and power metrics in the AD-M1 divider. The AD-M2 design introduces approximation by eliminating p cells of each row by reducing in the divider and making use of an equality-detector circuit for the most-significant $n - p$ bits of a and b . Reducing the number of cells in the divider design reduces area and power consumption. Both dividers are used to demonstrate a change detection application.

Farhad Ebrahimi Azandaryani et al. (2019) designed a block-based carry speculative approximate adder (BCSA), which was based on dividing an exact adder into some non-overlapped blocks operated in parallel. Each block may be composed of any desired type of adders. In this adder, the length of carry chain was reduced to was utilized to estimate the carry. A select logic was suggested to speculate the carry input of each block based on some input operand bits of the current and next block. In addition, to decrease the accuracy loss, an error detection and recovery mechanism was suggested. Based on the results, for the different approximate operating modes, BCSAERU showed, on average, about 50% improvements in the cost function, compared to the recently proposed approximate adders.

Manikantta Reddy K et al. (2018) focused design approximate non restoring and restoring dividers for unsigned integer division. Three designs of approximate subtractor cells are proposed and are used in the design of dividers, which reduced the power consumption and delay compared to that of exact dividers available in literature. The efficiency of proposed dividers is extensively analyzed with several error metrics. Finally, this paper concludes that proposed approximate restoring dividers are performing better than approximate non-restoring dividers both in-terms of design parameters and accuracy metrics.

Shaghayegh Vahdat et al. (2017) designed a a low-power and energy-efficient approximate divider. In the proposed divider, truncated form of the dividend is (approximately) multiplied by the approximate inverse value of the divisor. Comparing the design parameters of the proposed approximate divider with those of other relevant dividers revealed that this structure provides 66% and 52% improvements in the area and energy consumption, respectively. Also, delay, area, and energy consumption of the proposed unsigned divider were almost, on average, 94.4%, 94.4%, and 99.93% smaller compared to the exact SRT Radix-4 divider. Finally, employing the proposed divider in image division application showed its effectiveness in signal processing applications.

2.2 LITERATURE SUMMARY

Instead of having different circuits, reconfigurable circuits can be used to do the desired work. Thus, the paper presents reconfigurable subtractor. The performance analysis of conventional circuit has been shown using different techniques.

2.3 PROBLEM DEFINITION

To design a reconfigurable energy efficient approximate arithmetic architecture and analyze the performance using Synopsis Hspice tool. Therefore, the performance of the circuits has become the bottle neck for the overall system performance.

2.4 OBJECTIVE OF THE PROJECT

- To analyse the existing adder, subtractor.
- To design the proposed reconfigurable energy efficient adder, ripple carry adder, ripple borrow subtractor.
- To analyse the performance of proposed reconfigurable circuits using Hspice tool.

CHAPTER 3

AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 PROPOSED METHODOLOGY

- The proposed reconfigurable architectures provide multiple output functions.
- The number of output functions produced in the proposed concept is more as compared to conventional circuits.
- The proposed system is more efficient in terms of output functions.

3.2 AREA OF THE PROPOSED WORK AND APPLICATION

Low power and fast arithmetic units have been proposed in the literature for compute-intensive applications. The arithmetic units designed are mostly over-engineered and establishing the fact that most applications dealing with real-time data like audio, video, images are error-resilient (due to analog nature of data) and thus can be explored for approximations. A vast literature on approximate adders and multipliers is available applying approximate computing techniques from device to architecture level.

3.3 FULL SUBTRACTOR

The full subtractor is one among the combinational circuits, which is used for performing subtraction of three input bits; the minuend A, subtrahend B, and borrow in B_{in} . The output of subtractor is two bits: the difference D and borrow B_{out} . When a borrow out is generated, two is added in the current digit. This is similar to the subtraction algorithm in decimal.

A subtractor can be created using the same method as that of an adder. The binary subtraction is explained here. Like an adder, within the general case of calculations on multi-bit numbers, three bits area unit concerned in playacting the subtraction for every little bit of the distinction. Subtractors are usually implemented within a binary adder for only a small cost when using the

standard two's complement notation, by providing an addition/subtraction selector to the carry-in and to invert the second operand.

The inputs which are A, B and Bin, represent the minuend, subtrahend, and previous borrow. The two outputs, D and Bout represent the difference and output borrow, respectively. The full subtractor is one among the combinational circuits, which is used for performing subtraction of three input bits; the minuend A, subtrahend B, and borrow in Bin. The output of subtractor is two bits: the difference D and borrow Bout. When a borrow out is generated, two is added in the current digit. This is similar to the subtraction algorithm in decimal.

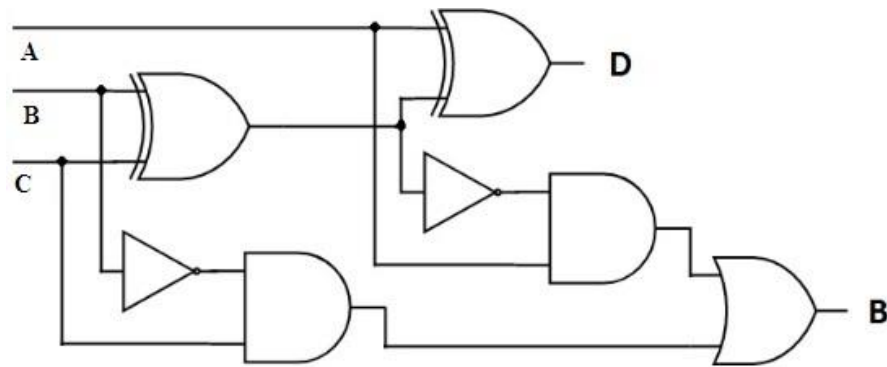


Fig:3.1: Block Diagram of Full Subtractor

(Source [14])

Table 3.1: Truth Table of Full subtractor

Inputs			Outputs	
A	B	Borrow _{in}	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Hence, this is the truth table of full subtractor and it explains the logic and functioning of the circuit.

3.4 4×1 MULTIPLEXER

Multiplexer is a combinational logic circuit which allows only one input at a particular time to generate the output. The signals which control which input will be reflected at the output end is determined by the select input lines. A multiplexer is often written as MUX in the abbreviated form. It is also called as Many-to-One circuit. This is because of its ability to select one signal out of many inputs. Multiplexing is the property of combining one or more signals and transmitting on a single channel. This is achieved by the device multiplexer. A multiplexer is the most frequently used combinational circuits and important building block in many in digital systems.

These are mostly used to form a selected path between multiple sources and a single destination. A basic multiplexer circuit consists of various data input lines and a single output line. These are found in many digital system applications such as data selection and data routing, logic function generators, digital counters with multiplexed displays, telephone network, communication systems, waveform generators, etc. The multiplexer or MUX is a digital switch, also called as data selector. It is a combinational circuit with more than one input line, one output line and more than one select line. It allows the binary information from several input lines or sources and depending on the set of select lines, particular input line, is routed onto a single output line. A 4-to-1 multiplexer contains of four data input lines from D0 to D3, two select lines which are S0 and S1 and an output line Y as shown in figure 3.2[13]. The choose lines S1 and S2 choose one in all the four input lines to attach the output line. the actual input combination on choose lines selects one in all input (D0 through D3) to the output.

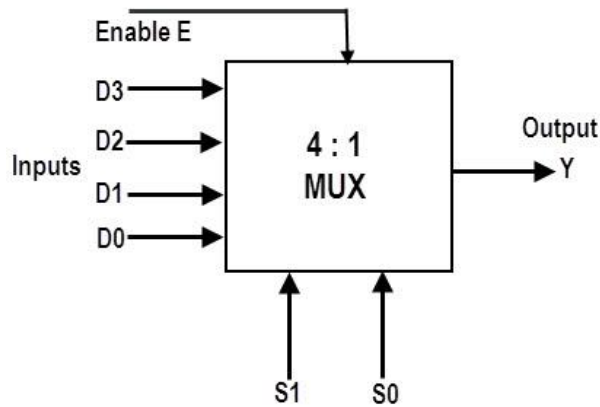


Fig:3.2: Block Diagram of 4×1 Multiplexer

The truth table of a 4-to-1 multiplexer is shown below in table 3.2[13] which four input combinations 00, 10, 01 and 11 on the select lines respectively switches the inputs D0, D2, D1 and D3 to the output. That means when $S_1=0$ and $S_0=0$, the output at Y is D0, similarly Y is D1 if the select inputs $S_1=0$ and $S_0=1$.

Table 3.2: Truth Table of 4×1 Multiplexer

Select Data Inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

From the above expression of the output, a 4-to-1 multiplexer can be implemented by using basic logic gates. The figure 1.7 shows the logic circuit of 4:1 MUX which is implemented by four 3-inputs AND gates, two 1-input NOT gates, and one 4-inputs OR gate. The figure 3.2 above shows the block diagram of a 4-to-1 multiplexer in which the multiplexer decodes the input through select line

3.5 Existing Reconfigurable Full Subtractor

The full subtractor is one among the combinational circuits, which is used for performing subtraction of three input bits; the minuend A, subtrahend B, and borrow in B_{in} . The output of subtractor is two bits: the difference D and borrow B_{out} . When a borrow out is generated, two is added in the current digit. This is similar to the subtraction algorithm in decimal.

The binary subtraction process is explained here. As with an adder, in the general case of calculations on multi-bit numbers, three bits are involved in doing the subtraction for respective bit of the difference. Subtractors are usually implemented within a binary adder for only a small cost when using the standard two's complement notation, by providing an addition/subtraction selector to the carry-in and to invert the second operand.

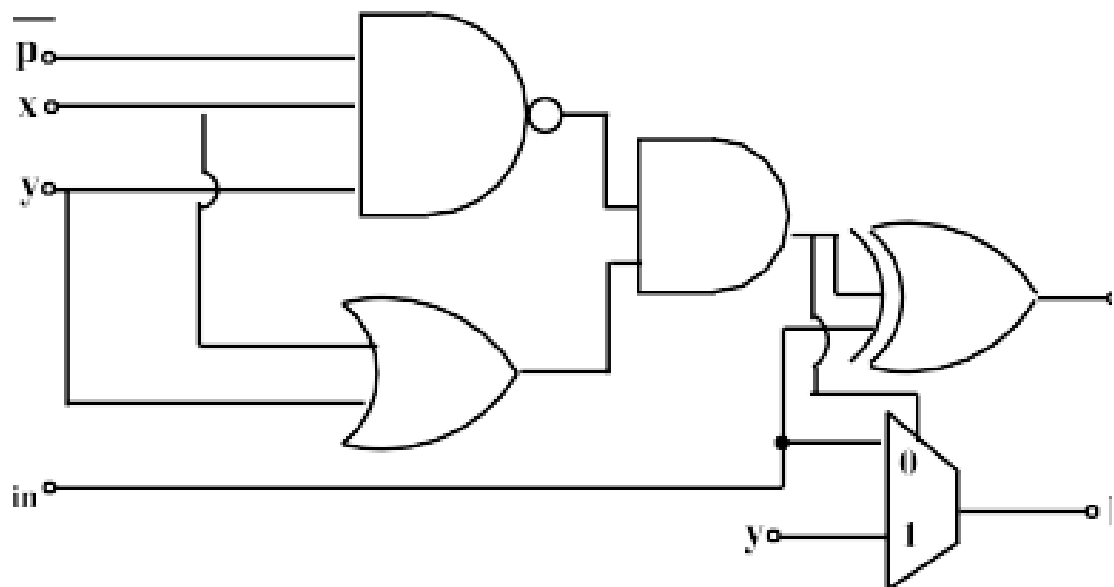


Fig:3.3: reconfigurable full subtractor

(Source [15])

The design shown above in figure 3.3 is the reconfigurable full subtractor. It has inputs x, y, zin connected to respective logic gates. All of the multiplexers in the circuit share the same select lines, S_1 and S_0 , in order to select the mode in which the shift registers operate.

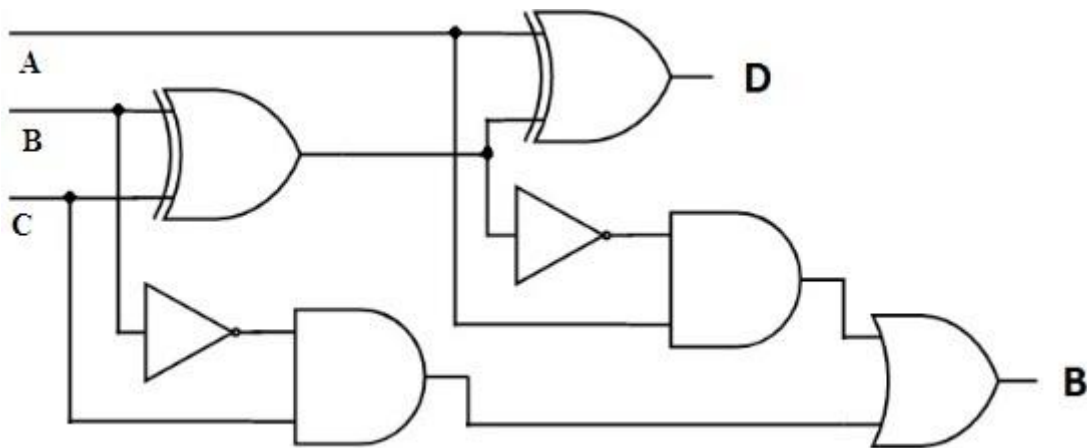


Fig:3.4: Conventional full subtractor

Basic connections—The first input (zeroth pin of multiplexer) is connected to the output pin of the corresponding flip-flop. The second input (first pin of multiplexer) is connected to the output of the very-previous flip flop which facilitates the right shift. The third input (second pin of multiplexer) is connected to the output of the very-next flip-flop which facilitates the left shift. The fourth input (third pin of multiplexer) is connected to the individual bits of the input data which facilitates parallel loading.

The working of the Universal shift register depends on the inputs given to the select lines. The register operations performed for the various inputs of select lines are given in table 3.3 as follows:

Table 3.3: Truth Table of Full subtractor
(Source [15])

Inputs			Outputs	
A	B	Borrow _{in}	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

3.6 PROPOSED RECONFIGURABLE CIRCUITS

Energy potency has emerged in concert of the foremost essential style parameters in modern computer system style. Approximate computing could be a new computing paradigm to realize energy potency by trading- off energy/area/latency enhancements with accuracy for error-resilient applications. The paper conjointly introduces the look of overflow detector victimisation lowest hardware resources. A comprehensive accuracy and hardware analysis on CMOS 45-nm technology node area unit performed for the planned dividers furthermore as different state-of-the- art reconfigurable circuit styles. Compared to the correct circuit styles, the planned circuits shows an improvement of forty ninth in terms of energy potency and is 1.26x faster, whereas introducing lowest errors. The planned style is in contestable for its potency in image process tasks and shows nominal result on the output quality.

CHAPTER 4

PROPOSED RECONFIGURABLE CIRCUITS

In the existing work a Reconfigurable full subtractor has designed using logic gates and 4*1 multiplexer. In the proposed work Reconfigurable full adder, ripple carry adder and ripple borrow subtractor are designed using a Hspice Tool and the proposed design have less delay compared to the existing circuits and also with conventional circuits.

4.1 RECONFIGURABLE RIPPLE BORROW SUBTRACTOR

The redundant transitions and the long-stacked transistor paths in the previous or general ripple borrow circuit can significantly increase the power dissipation. To overcome the aforementioned challenges, a Reconfigurable ripple borrow subtractor is proposed to effectively reduce the power consumption without degrading the speed performance. The schematic of the proposed RBS is illustrated in Figure 4.1. It is made up of four full subtractors cells and approximate signal. The borrow is generated from the last subtractor cell. As we know these are sequential circuits, the output of one subtractor cell is given as the input to the successive subtractor cell.

Subtracting circuits use two N-bit operands to produce an N-bit result and a borrow out signal. Subtractor circuits are rarely encountered in digital systems (for reasons that will be explained later), but they nevertheless provide an interesting design opportunity. Like adders, the simplest subtracting circuits perform subtraction bit-by-bit from the LSB to the MSB, and they are most easily designed using the bit-slice approach. The design process closely follows adder design, in that sample subtraction problems can be studied to gain insight into subtractor requirements, a truth table can be prepared based on the

observations, and then a circuit can be designed from the truth table. Once a bit-slice subtractor circuit has been designed, it can be replicated N time to create an N-bit subtractor.

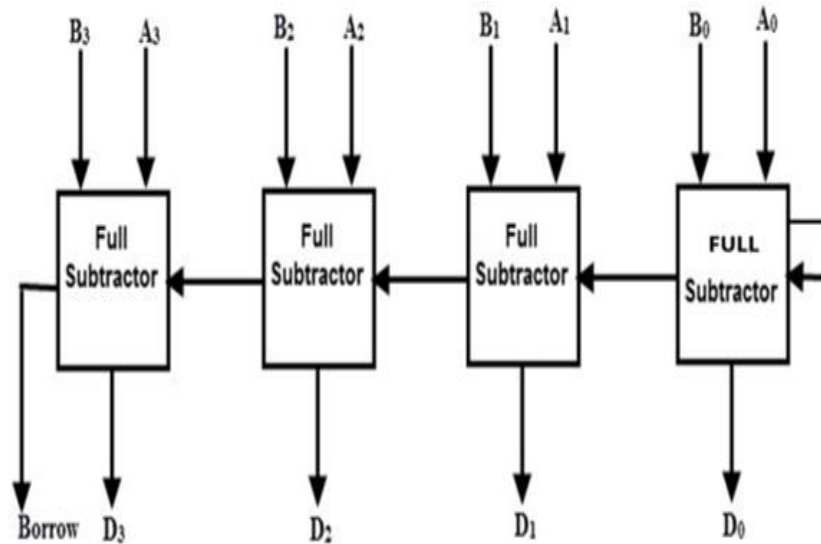


Fig:4.1: Ripple borrow subtractor

4.2 RECONFIGURABLE FULL ADDER

The redundant transitions and the long-stacked transistor paths in the previous or general full adder circuit can significantly increase the power dissipation. To overcome the aforementioned challenges, a Reconfigurable full adder is proposed to effectively reduce the power consumption without degrading the speed performance. The schematic of the proposed Reconfigurable full adder is illustrated in Figure 4.2.1. It is made up of logic gates and approximate signal.

Full Adder is that the adder that adds three inputs and produces two outputs. the primary two inputs area unit A associated with B and therefore the third input is an input carry as C-IN. The output carry is selected as C-OUT and therefore the traditional output is selected as S that is total. A full adder logic is intended in such a way that may take eight inputs along to form a byte-wide adder and cascade the carry bit from one adder to the another. The circuit is represented in fig.4.2.

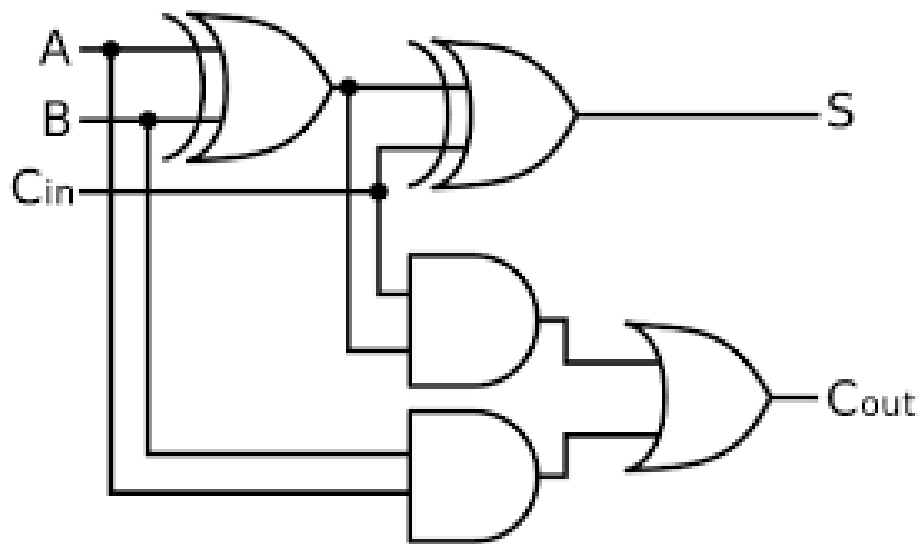


Fig:4.2: Existing full adder circuit

The proposed circuit has many changes compared to the existing one. The logic gates used are almost similar but here an additional OR, AND gates are used. The power consumption and delay values are way better than the previous ones. This technique allows us in reduction of power consumption, delay and area of digital circuits, along with maintaining low complexity of logic design.

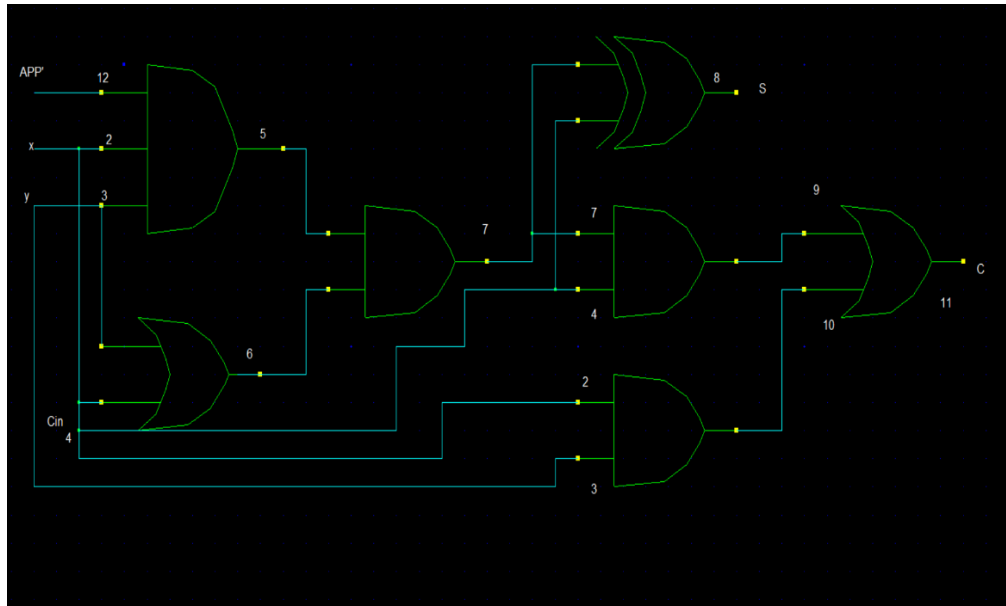


Fig:4.2.1: Reconfigurable full adder

The existing full adder will be modified, such that the proposed architecture will be more efficient in terms of average power, speed and energy consumption. Proposed Reconfigurable full adder has designed using Hspice tool. The existing conventional full adder has designed and analyzed.

The proposed low power 4×1 multiplexer using GDI Technology along with proposed low power D-flip-flop has been used in the proposed Universal Shift Register to analyze its performance. In this proposed architecture of 4×1 Multiplexer using GDI Technology and low power D-Flip-Flop which has been used to reduce the power and delay of the Universal Shift Register.

4.3 RECONFIGURABLE RIPPLE CARRY ADDER

The redundant transitions and the long-stacked transistor paths in the previous or general full adder circuit can significantly increase the power dissipation. To overcome the aforementioned challenges, a Reconfigurable ripple carry adder is proposed to effectively reduce the power consumption without degrading the speed performance. The schematic of the proposed RCA is illustrated in Figure 4.3. It is designed using logic gates and approximate signal.

The existing full adder has been designed and analyzed using conventional method only.

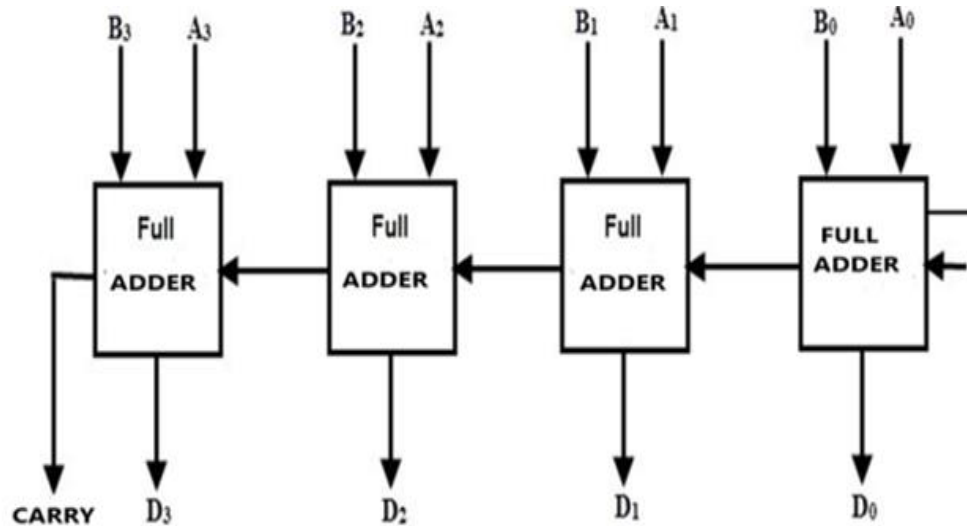


Fig.4.3 Ripple carry adder

The circuit is designed using multiple full adders to add N -bit numbers. Each full adder inputs C_{in} , which is the C_{out} of the previous adder. This kind of adder is called a ripple-carry adder, since each carry bit "ripples" to the next full adder. The layout of a ripple-carry adder is simple, which allows fast design time; however, the ripple-carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder.

The existing ripple carry adder will be modified, such that the proposed architecture will be more efficient in terms of average power, speed and energy consumption. Proposed Reconfigurable ripple carry adder has been designed using Hspice tool. The existing full adder has been designed and analyzed using conventional method only.

CHAPTER 5

RESULTS AND DISCUSSION

The existing architectures are analyzed using H-spice tool. And the proposed Architectures are designed and analyzed in H-spice environment. Output of the existing Full adder, Reconfigurable full subtractor and proposed Reconfigurable circuits are analysed through the analysis done in Hspice environment and it is focuses that the proposed architecture reduces the delay compared with the existing architectures.

The design has been evaluated in 130 nanometre technology. The spice code of the schematic design can be viewed in edit NL in H-spice tool. It is simulated. The avan waves in H-spice tool shows the transient analysis of the corresponding design with the specified time values.

5.1 EXISTING FULL SUBTRACTOR SIMULATION OUTPUT

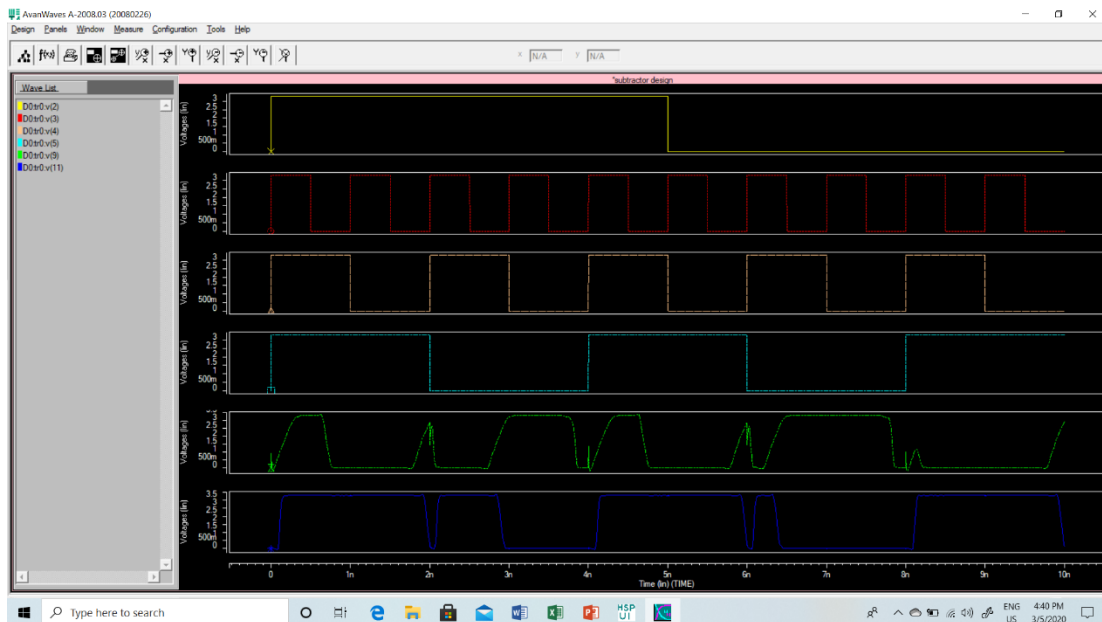


Fig:5.1: Output Waveform for existing full subtractor

Figure 5.1 shows the simulation output of an existing full subtractor. The input and output are given as: v (2)-approximate signal, v (3)-input, v (4)-input, v (5)-input, v (9)-output, v (11)-output.

5.2 PROPOSED RIPPLE BORROW SUBTRACTOR SIMULATION OUTPUT

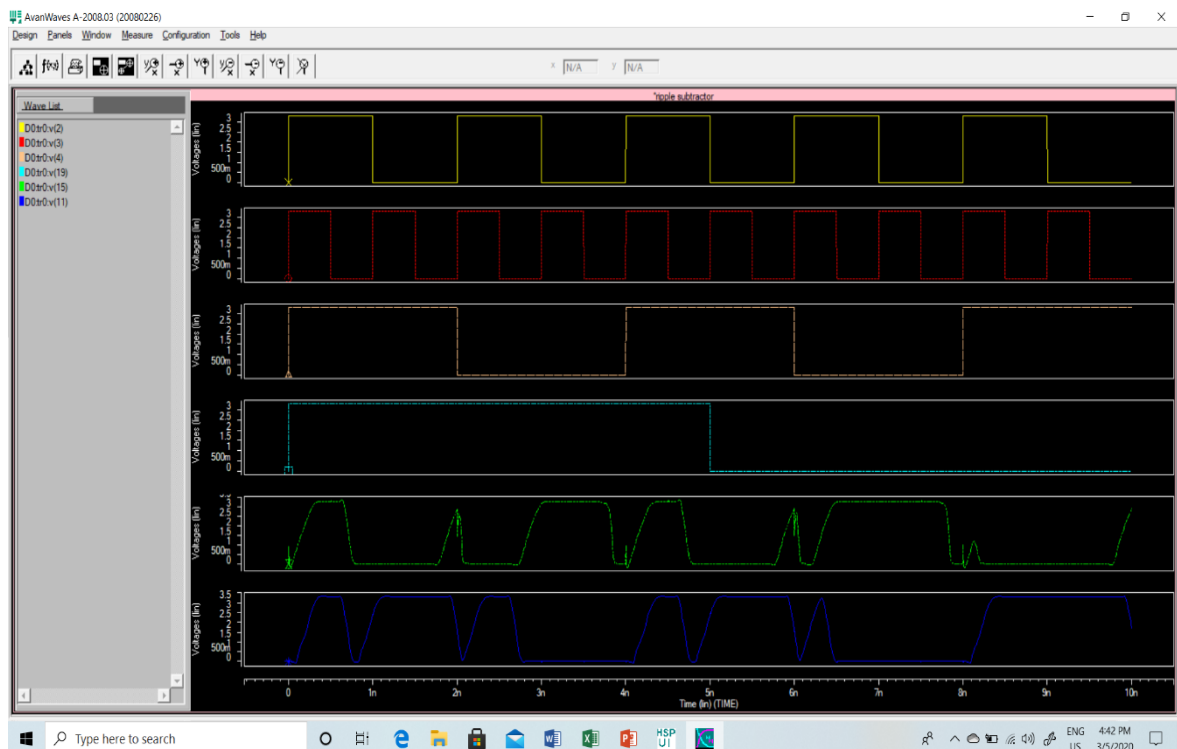


Fig:5.2: Output Waveform for proposed ripple borrow subtractor

The Figure 5.2 shows the simulation output for the proposed ripple borrow subtractor. The inputs are given as: v (2), v (3), v (4) whereas the outputs are & v (15), v (11) and v (19) are the approximate signal.

5.3 PROPOSED RECONFIGURABLE FULL ADDER SIMULATION OUTPUT

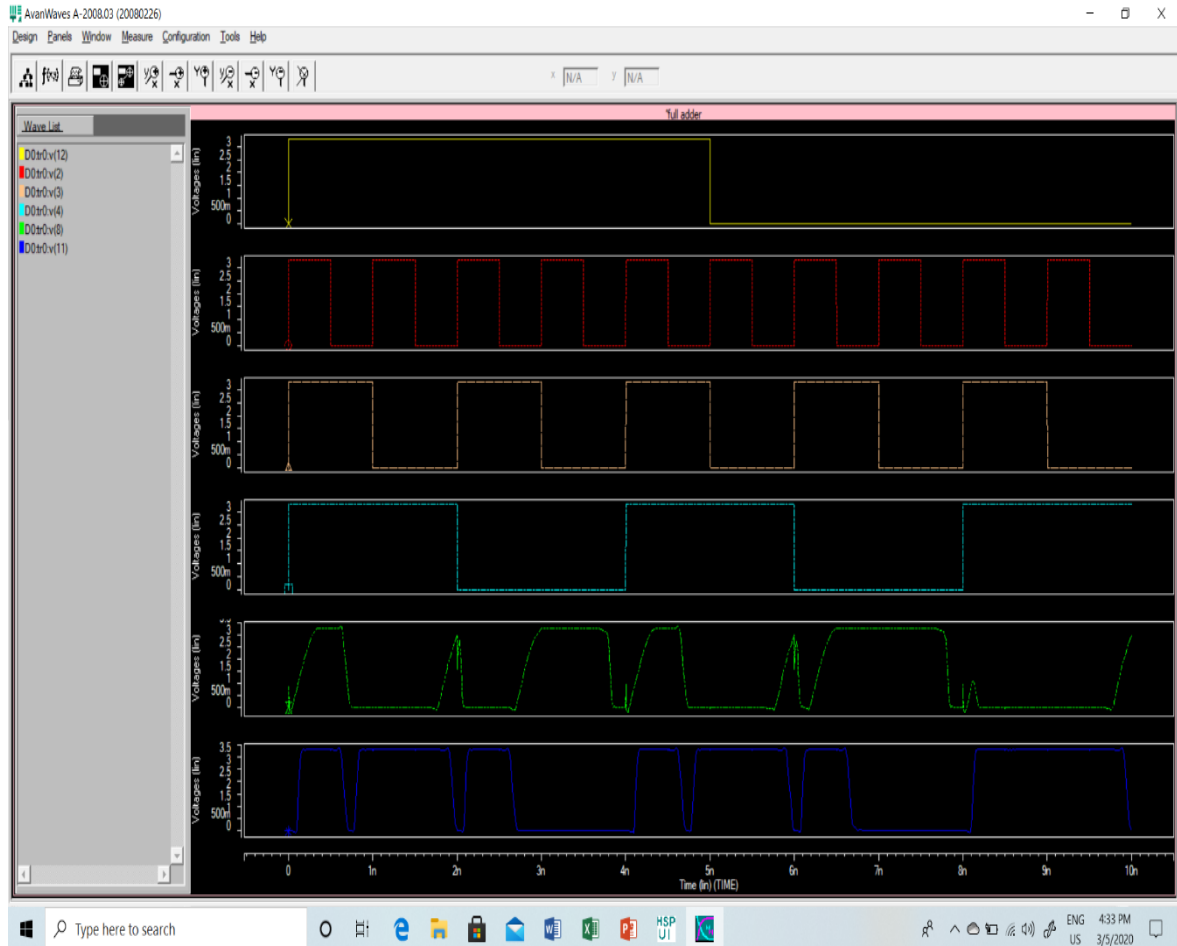


Fig:5.3: Output Waveform for Full adder

Figure 5.3 shows the simulation results for the proposed reconfigurable full adder. The inputs are given as v (2), v (3), v (4) and the outputs are v (8), v (11) and the approximate signal is v (12).

5.4 RECONFIGURABLE RIPPLE CARRY ADDER

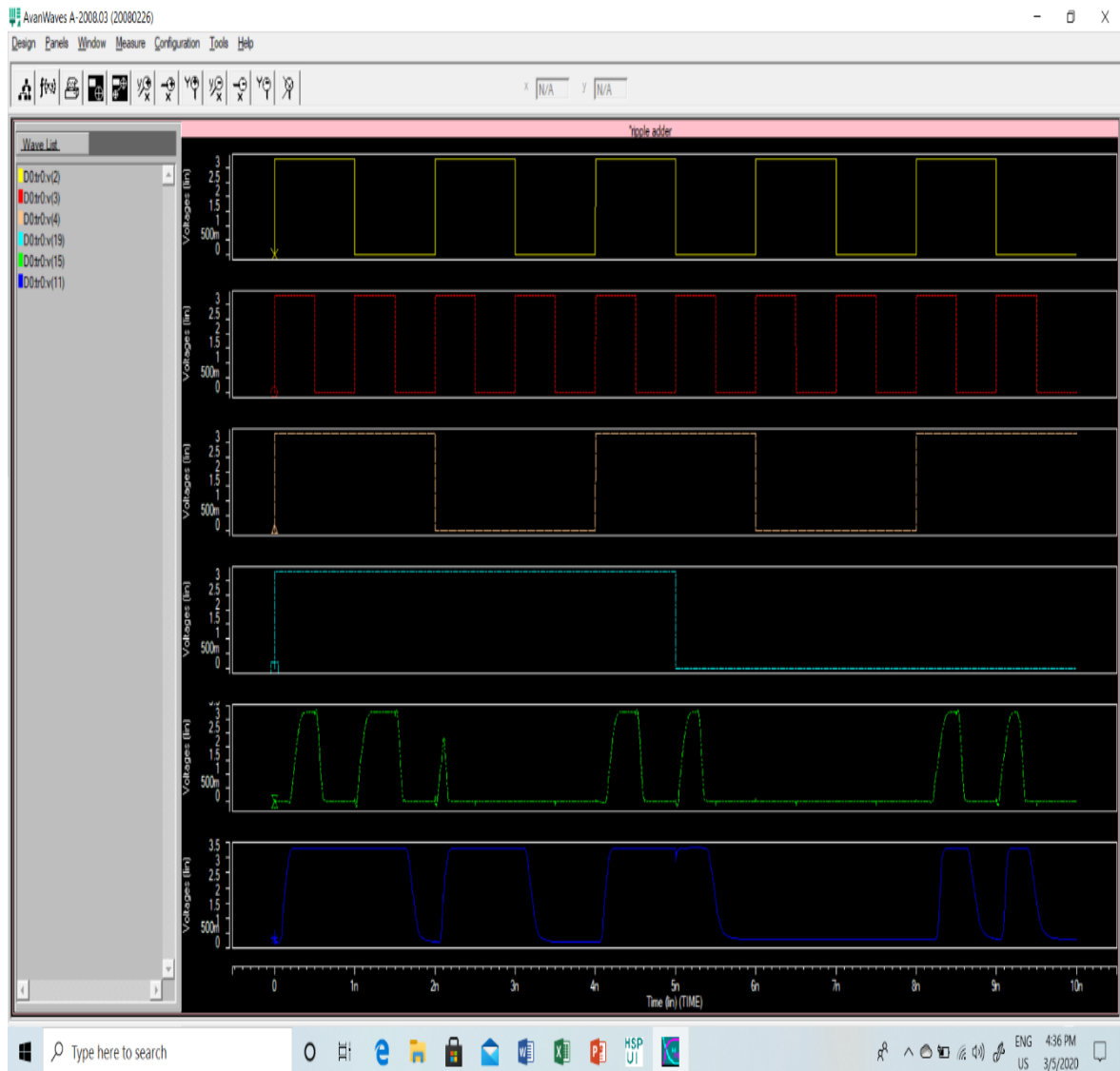


Fig:5.4: Output Waveform for ripple carry adder

Figure 5.4 shows the simulation output for proposed reconfigurable ripple carry adder. The inputs are given as v (2), v (3), v (4) and the outputs are v (15), v (11) and approximate signal is v (19).

5.5 PERFORMANCE ANALYSIS

The performance of Reconfigurable full subtractor, Ripple borrow subtractor, Full adder, Ripple carry adder are analyzed in terms of power and delay and energy consumption.

5.5.1. Power and Delay Analysis of Existing circuits

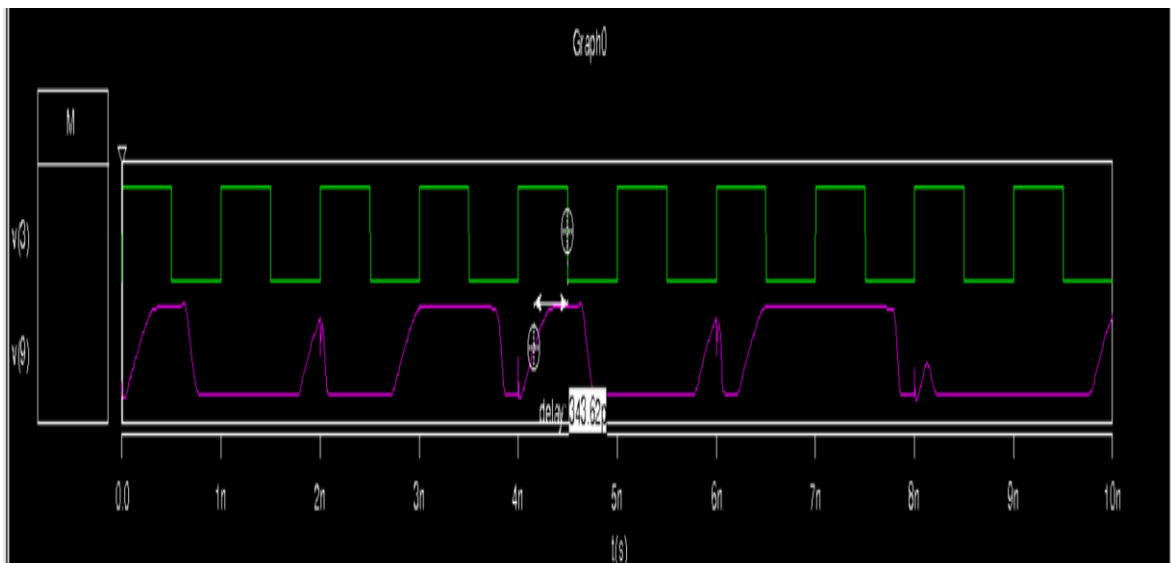


Fig:5.5.1: Delay analysis of RFS

From the Table 5.1 the power and delay of existing work has been shown.

Table 5.1: Performance analysis of existing reconfigurable full subtractor

EXISTING CIRCUIT	AVERAGE POWER (μ W)	DELAY (pS)	PDP (fJ)
Reconfigurable full subtractor	137.3	527.72	72.297

5.5.2 Power and Delay Analysis of proposed circuits

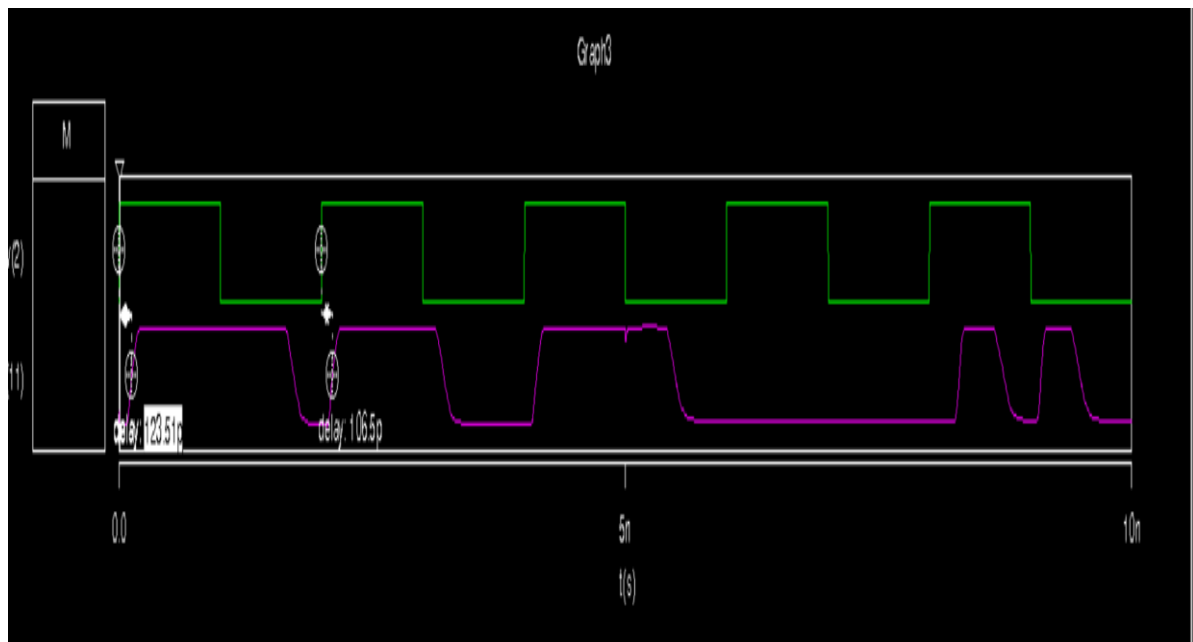


Fig:5.5.2: Delay analysis of RCA

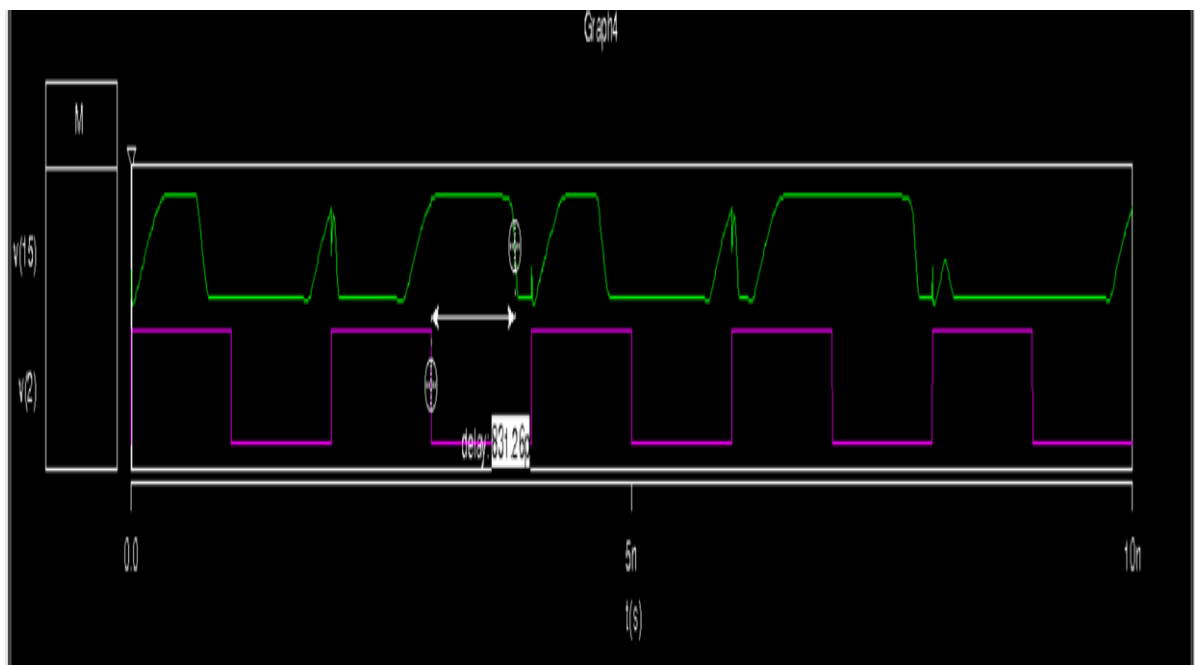


Fig:5.5.2.1: Delay analysis of RBS

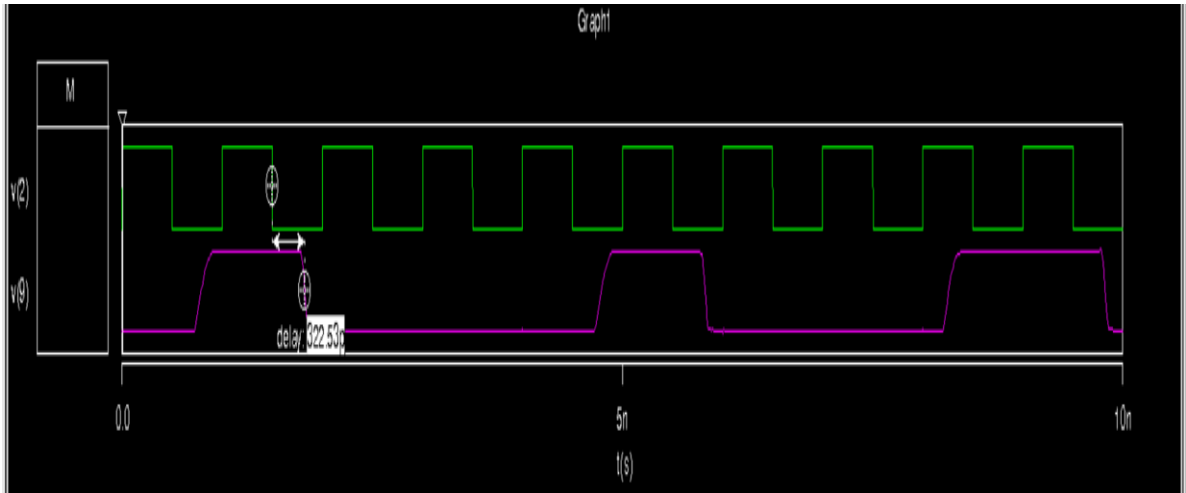


Fig:5.5.2.2: Delay analysis of RFA

5.5.3 Performance analysis of Proposed Reconfigurable circuits

From the Table 5.3 the power and delay of proposed work of the full adder, reconfigurable ripple carry adder, reconfigurable ripple borrow subtractor have been shown.

Table 5.3: Performance analysis of proposed circuits

PROPOSED CIRCUIT	AVERAGE POWER (μW)	DELAY (pS)	PDP (fJ)
Full adder	127.6	322.53	40.96
Reconfigurable Ripple carry adder	144.0	716.76	243.21
Reconfigurable Ripple borrow subtractor	166.9	333.26	556.21

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

A. CONCLUSION

The proposed design of reconfigurable full adder, ripple carry adder and ripple borrow subtractors gives more output functions as compared to the conventional circuits. The proposed reconfigurable architectures are designed and the timing response are analyzed through transient analysis and the performance analysis are analyzed through power, delay and energy consumption analysis. So the proposed reconfigurable computational blocks are more suitable for reconfigurable architectures.

B. FUTURE SCOPE

Reconfigurable energy efficient approximate arithmetic logic units can be designed using these proposed computational blocks. In future Reconfigurable processor can be designed using reconfigurable computational blocks such as reconfigurable ALU and Memory and so on.

REFERENCES

- [1]. A. Raghunathan, v. Gupta, D. Mohapatra, K. Roy, Low-power digital signal processing using approximate adders, *IEEE Trans. Compute. -Aided Des. Integer. Circuits Syst.* 32 (1) (2012) 124–137.
- [2]. A. Raha, V. Raghunathan, Towards full-system energy-accuracy trade-offs: A case study of an approximate smart camera system, in: 2017 54th ACM/EDAC/IEEE Design Automation Conference, DAC, IEEE, 2017, pp. 1–6.
- [3]. F. Ebrahimi-Azandaryani, O. Akbari, M. Kamal, A. Afzali Kusha, M. Pedram, Block-based carry speculative approximate adder for energy-efficient applications, *IEEE Trans. Circuits Syst. II* (2019)
- [4]. H. Saadat, H. Javaid, S. Parameswaran, Approximate integer and floating-point dividers with near-zero error bias, in: 2019 56th ACM/IEEE Design Automation Conference, DAC, IEEE, 2019, pp. 1–6.
- [5]. M. Schulte, F. Lombardi, W. Liu, A retrospective and prospective view of approximate computing point of view, *Proceedings of the IEEE* 108 (3) (2020) 394–399.
- [6]. M. Vasantha, K.M. Reddy, Y.N. Kumar, D. Dwivedi, Design of approximate dividers for error tolerant applications, in: 2018 IEEE 61st International Midwest Symposium on Circuits and Systems, MWSCAS, IEEE, 2018, pp. 496–499.
- [7]. M. Imani, R. Garcia, A. Huang, T. Rosing, Cade: Configurable approximate divider for energy efficiency, in: 2019 Design, Automation & Test in Europe Conference & Exhibition, DATE, IEEE, 2019, pp. 586–589.

- [8]. Raj Kumar Mistri, Rahul Ranjan, Pooja Prasad, Anupriya, "IC Layout Design of 4-bit Universal Shift Register using Electric VLSI Design System", International Journal of Engineering Research & Technology (IJERT), 2017
- [9]. Yu Chien-Cheng, Chin Ping-Yuan "Design of Low-Power Double Edge Triggered Flip-Flop Circuit" 2007 sixth IEEE Conference on ASIC, vol1, May 2005
- [10]. A.M. Devices, Software Optimization Guide for AMD Family 15h Processors, Number, 2014.
- [11]. M. Morris mano, Michael D. Ciletti, Digital Design, 5th edition, Pearson 2013.
- [12]. S.M. Sze, VLSI technology ,2nd edition, McGraw hill companies, 1988.
- [13]. www.elsevier.com/locate/vlsi

ORIGINALITY REPORT

7%

SIMILARITY INDEX

5%

INTERNET SOURCES

4%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Copperbelt University

Student Paper

1%

2

Suganthi Venkatachalam, Elizabeth Adams, Seok-Bum Ko. "Design of Approximate Restoring Dividers", 2019 IEEE International Symposium on Circuits and Systems (ISCAS), 2019

Publication

1%

3

en.wikipedia.org

Internet Source

1%

4

podelise.ru

Internet Source

<1%

5

www.electronicshub.org

Internet Source

<1%

6

Neelam Arya, Teena Soni, Manisha Pattanaik, G.K. Sharma. "READ: A fixed restoring array based accuracy-configurable approximate divider for energy efficiency", Integration, 2020

Publication

<1%

7	connection.ebscohost.com Internet Source	<1 %
8	Submitted to Guru Jambheshwar University of Science & Technology Student Paper	<1 %
9	www.scribd.com Internet Source	<1 %
10	Submitted to UNIVERSITY OF LUSAKA Student Paper	<1 %
11	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	<1 %
12	pages.rediff.com Internet Source	<1 %
13	Submitted to Far Eastern University Student Paper	<1 %
14	theses.gla.ac.uk Internet Source	<1 %
15	Submitted to Sogang University Student Paper	<1 %
16	Submitted to Ghana Technology University College Student Paper	<1 %
17	Submitted to KUMARAGURU COLLEGE OF TECHNOLOGY	<1 %

18

Submitted to Sardar Vallabhbhai National
Inst. of Tech.Surat

Student Paper

<1 %

19

ddfe.curtin.edu.au

Internet Source

<1 %

20

Submitted to International School of
Management and Technology (ISMT), Nepal

Student Paper

<1 %

21

A Complete Concordance Or Verbal Index to
Words Phrases and Passages in the Dramatic
Works of Shakespeare with a Supplementary
Concordance to the Poems, 2003.

Publication

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On