

A Project report on

DESIGN AND IMPLEMENTATION OF OF 4-BIT ARITHMETIC LOGIC UNIT USING MAJORITY GATES

EC-399: MINI PROJECT

Submitted By

Vucha Suma sree

Roll no. 621272, Sec-B, ECE

Under the Esteemed guidance of

Mr. Sanath Kumar Tulasi

Ad-Hoc Faculty, Department of ECE



DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

(An autonomous Institute under the aegis of Ministry of Education, Government of India)

Tadepalligudem – 534101, AP

(2021-25)



NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

(An autonomous Institute under the aegis of Ministry of Education, Government of India)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



CERTIFICATE

This is to certify that the project report entitled "**DESIGN AND IMPLEMENTATION OF 4-BIT ARITHMETIC LOGIC UNIT USING MAJORITY GATE**" done by Vucha Sumasree bearing Roll no. 621272, has been carried out under my/our supervision and that this work has not been submitted elsewhere for a minor project.

Signature of Supervisor

Mr. Sanath Kumar Tulasi

Ad-Hoc Faculty

Dept. of Electronics and Communication Engineering

N.I.T. Andhra Pradesh

April 16, 2024

ACKNOWLEDGEMENT

I would like to share my sincere gratitude to our supervisor “Mr. Sanath Kumar Tulasi” and “Mrs. Dhanasree” for helping me in completion of this project. During the work, I faced many challenges as the field is new and due to lack of experience, but they helped me to get over all the difficulties and in the final completion of my project. I have learnt many new things during the project. I’m very thankful to all faculty members for their support.

We profoundly thank “Dr. S. Yuvaraj”, Head of ECE department who has been an excellent guide and also a great source of inspiration to my work. I would like to acknowledge the help for giving me all the required permissions and helping me to complete the project. I would like to thank all the faculty of the ECE department for teaching me and for spending time on the review. I also thank my friends who directly or indirectly helped me in my project work. Lastly, I would like to thank my family for their selfless support.

Vucha Sumasree

Roll no. 621272

DECLARATION

I declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented/fabricated/falsified any idea/data/fact/source in my submission.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Vucha Sumasree

Roll no. 621272

\

CONTENTS

S.NO	CHAPTER NAME	PAGE NO
1	INTRODUCTION	
	1.1 Introduction to CMOS Technology	1
	1.2 Introduction to QCA Technology	2
	1.2.1 Quantum Basic Cell	2
	1.2.2 QCA Polarization	2
	1.3 Introduction to Majority Gate	3
2	LITERATURE SURVEY	4
3	PROBLEM STATEMENT	
	3.1 Problem Statement	5
	3.2 Objectives	5
4	METHODOLOGY	
	4.1 Implementation Methodology	6
	4.1.1 Software Description	6
	4.2 Circuit Diagram of 4-bit ALU	7
	4.3 Block Diagram	7
	4.4 Proposed Design using Majority Gates	8-13
	4.4.1 4-bit Adder	8
	4.4.2 4-bit Subtractor	8
	4.4.3 4-bit Multiplier	9
	4.4.4 4-bit Divider	9-10
	4.4.5 Shift Left Operator	10

4.4.6 Shift Right Operator	11
4.4.7 Rotate Left Operator	11
4.4.8 Rotate Right Operator	11
4.4.9 Logic Gates	12-13
4.4.10 4-bit Equality Comparator	13
5 RESULTS AND DISCUSSION	
5.1 ALU Design using Logic Gates(Structural Modelling)	14-17
5.1.1 Simulation Waveform	14
5.1.2 TCL Console	15
5.1.3 Schematic Diagram	16
5.1.3 Synthesis & Implementation	\ 16-17
5.2 ALU Design using Case Statement (Behavioral Modelling)	17-18
4.2.1 Schematic Diagram	17
4.2.2 Synthesis & Implementation	18
5.3 ALU Design using Majority Gates (Proposed Design)	19
4.3.1 Synthesis & Implementation	19
5.4 Comparison of Power, Timing & Area Utilization	20
5 CONCLUSION AND FUTURE SCOPE	21
6 REFERENCES	22

LIST OF FIGURES:

S. NO	FIGURE NAME	PAGE NO
1.1	Polarization in QCA Cell (Binary 0 & Binary 1)	2
1.2	Majority Gate	3
4.1	Circuit Diagram of 4-bit ALU	7
4.2	Block Diagram	7
4.3	Full Adder using Majority gates	8
4.4	4-bit Ripple Carry Adder using Majority gates	8
4.5	4-bit Subtractor using Majority gates	8
4.6	4-bit Multiplier using Majority gates	9
4.7	4-bit Divider using Majority gates	10
4.8	Shift Left Operator using Majority gates	10
4.9	Shift Right Operator using Majority gates	11
4.10	Rotate Left Operator using Majority gates	11
4.11	Rotate Right Operator using Majority gates	11
4.12	4-bit AND gate using MG	12
4.13	4-bit OR gate using MG	12
4.14	4-bit NAND gate using MG	12
4.15	4-bit NOR gate using MG	12
4.16	4-bit XOR gate using MG	13
4.17	4-bit Equality Comparator gate using MG	13
5.1	Simulation wave form of ALU (1-8 operations)	14
5.2	Simulation wave form of ALU (9-16 operations)	14

5.3	TCL Console	15
5.4	Schematic Diagram of ALU	16
5.5	Power Analysis of traditional ALU	16
5.6	Timing Summary of traditional ALU	17
5.7	Area utilization of traditional ALU	17
5.8	Schematic Diagram of ALU (using case statement)	17
5.9	Power Analysis of ALU (using case statement)	18
5.10	Timing Summary of ALU (using case statement)	18
5.11	Area utilization of ALU (using case statement)	18
5.12	Power Analysis of traditional ALU (using Majority Gates)	19
5.13	Timing Summary of traditional ALU (using Majority Gates)	19
5.14	Area utilization of ALU (using Majority Gates)	19

LIST OF TABLES

S.NO	TABLE NAME	PAGE NO
5.1	Defining Operations of ALU	15
5.2	Power, Timing & Area of ALU (using logic gates)	20
5.3	Power, Timing & Area of ALU (using case statement)	20
5.4	Power, Timing & Area of ALU (using Majority Gates)	20

NOMENCLATURE

ALU: Arithmetic Logic Unit

CMOS: Complementary metal-oxide-semiconductor

QCA: Quantum-Dot Cellular Automata

VLSI: Very Large Scale Integrated Circuit

RTL: Register Transfer Level

HDL: Hardware Description Language

IDE: Integrated Development Environment

FPGA: Field Programmable Gate Array

SoC: System on Chip

MUX: Multiplexer

WCM: Wilder current mirror

FA: Full Adder

HA: Half Adder

MG: Majority Gate

LUT: Look Up Table

IOB: Input-Output Buffer

ABSTRACT

This project focuses on the design and implementation of a 4-bit Arithmetic Logic Unit (ALU) utilizing majority gates. The ALU serves as a key component in digital systems, performing arithmetic and logic operations on binary data. In this project, we explore the utilization of majority gates in constructing the ALU, aiming to achieve compactness, low power consumption, and high performance. Furthermore, we employ Quantum Dot Cellular Automata (QCA) technology for the realization of these majority gates, leveraging its potential for ultra-low power and high-density circuitry.

The design process involves the systematic arrangement and interconnection of majority gates to perform addition, subtraction, Multiplication, Division and logical operations on 4-bit binary inputs. Through simulation and verification, the functionality and performance of the designed ALU are evaluated, considering factors such as speed, area efficiency, and power consumption. This Designed ALU is then implemented to check the above mentioned parameters and compared with the ALU designed using logic circuits and operations.

Overall, this project contributes to the advancement of digital circuit design by exploring innovative approaches that combine majority gate logic with emerging nanotechnology, paving the way for more efficient and compact digital systems in the future.

Vucha Suma Sree

Roll no. 621272

CHAPTER 1

INTRODUCTION

Digital circuits form the backbone of modern computing systems, enabling a myriad of operations ranging from simple arithmetic calculations to complex logical decisions. Central to these circuits is ALU, a crucial component responsible for executing arithmetic and logic operations on binary data. In this project, Majority Gates are used to design and implement Arithmetic Logic Unit. The main goal of utilizing the majority gate is to reduce Area, power consumption and delay. In recent years, there has been a growing interest in exploring alternative approaches to ALU design that offer improvements in terms of compactness, power efficiency, and scalability. One such approach involves the utilization of majority gates, which are simple yet powerful logic gates capable of performing various logical functions based on the majority of their inputs. The Majority Gates forms the basic building blocks in QCA circuit design. Quantum-Dot Cellular Automata is an alternative to current CMOS technology which encounters leakage and short channel effects when further Scaling is done. In this Project, We see how QCA technology (using Majority Gates) benefits us from CMOS technology.

1.1 Introduction to CMOS Technology

CMOS is a predominant technology for manufacturing integrated circuits. CMOS stands for “Complementary Metal Oxide Semiconductor”. CMOS technology is used in the design of today's Very Large Scale Integration (VLSI) circuits, where a few millions or even billions of transistors (MOSFETs to be specific) are integrated into a single chip or die. The reasons for the dominant use of CMOS Technology in the fabrication of VLSI chips are reliability, low power consumption, considerably low cost and most importantly scalability. According to Moore's Law described by Gordan Moore, the number of devices on a chip will double every 18 to 24 months. Even though Gordan Moore did not imply it to CMOS, Moore's Law was successfully fulfilled due to the CMOS technology. Today, we are dealing with channel lengths as small as 7nm (at the time of publishing this tutorial), all because of the scaling ability in CMOS. However the performance of various circuits in current CMOS-based architectures is close to reaching the limit. If the feature size of transistors is further reduced to a nanometer, it will produce quantum effects such as tunneling.

1.2 Introduction to QCA Technology

Quantum-dot cellular automata (QCA) is a rising nanotechnology with the potential for quicker speed, smaller size and low power consumption than CMOS based technology. In this, unlike traditional structures, logical states or values are demonstrated no longer by voltage levels but rather by the position of electrons. At present we cannot replace CMOS technology with QCA technology, but QCA technology has the capacity to replace. To implement QCA circuits, some methods for simplifying and optimizing the design are needed. The main building block of QCA circuits has been the majority gate.

1.2.1 Quantum Basic Cell:

A Cell is a basic element for QCA logic design. Based on the charge of cells, QCA bit representation is determined. Carriers of charges placed in corners of a cube are also called quantum dots. A single QCA cell consists of 4 quantum dots which are aligned near four corners. State of logic in QCA cells are stored based on the electrons position of each cell, and it does not store the information on voltage levels.

1.2.2 QCA Polarization:

A QCA cell can be represented in binary with a specific polarization which is denoted as “P”. In this there are two polar states for particular QCA cell. These two stable states are formed because of Coulombic repulsion between two electrons which are denoted by a polarization, $P = -1$ for logic 0 and $P = +1$ for logic 1. The individual single cell can be configured as normal cell, fixed polarization cell, input cell or output cell as required for the application. A fixed polarization cell will remain in the same state; its state isn't affected by the neighbouring cells or the clock.

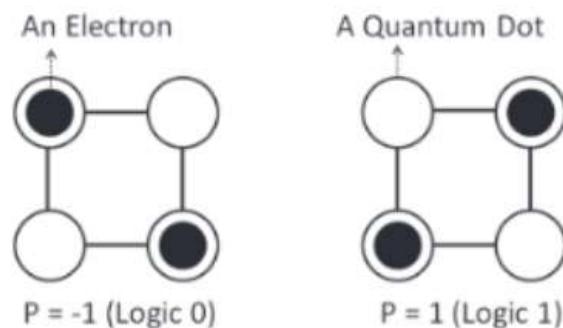


Fig1.1: Polarization in QCA Cell (Binary 0 & Binary 1)

1.3 Introduction to Majority Gate

The Majority Gates form the basic building blocks in QCA circuit design. The Three-input Majority Gate is widely used in the design and synthesis of the circuits. Structure and logic function of the majority gate are shown in Fig1.2. As can be seen from the figure, this gate consists of three inputs, one output and a device cell. The reason behind the naming of this gate is that this gate votes among the three inputs and transmits the majority polarization to the output. According to the figure, the device cell is always forced to have the majority polarization because only in this way the Coulombic repulsion among electrons of the three inputs will be minimized. Equation (1) defines the relationship between the inputs and the output of the majority gate.

$$M(A,B,C) = \text{Maj}(A,B,C) = AB + BC + CA \quad (1)$$

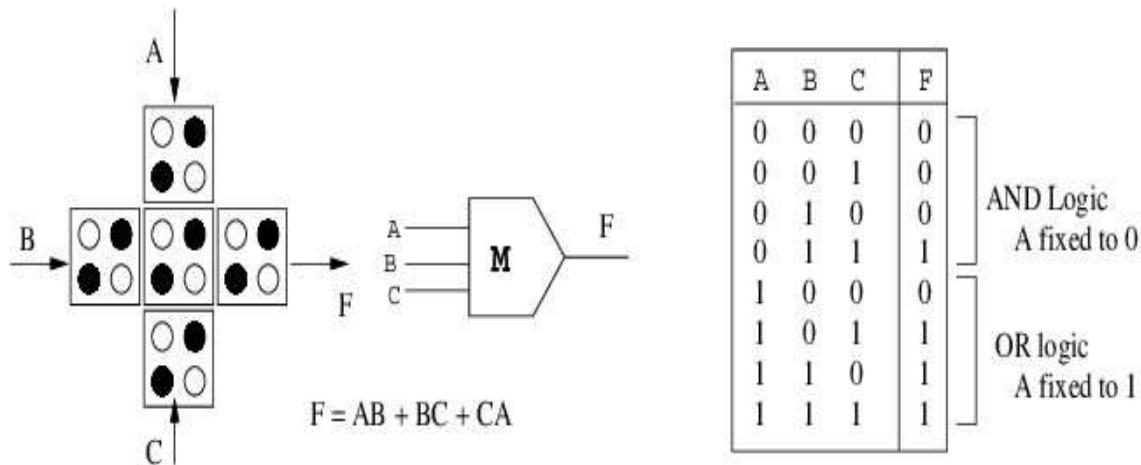


Fig1.2: Majority Gate

When a single input is set to polarization $P = -1$, then the output is said to be AND of the other two inputs. Moreover, if a single input is set to polarization $P = 1$, then the output will be OR of the other two inputs. It is worth mentioning that by using majority and inverter gates, any logical function can be realized. However, developing new gate designs using majority gates, in order to decrease the number of cells applied in a circuit and area are still necessary.

CHAPTER 2

LITERATURE SURVEY

Moore's law states that the number of transistors that could be integrated into a single die would grow exponentially with time. This causes increasing computational complexity of the chip and physical limitations of devices such as power consumption, interconnect will become very difficult. According to recent analysis the minimum limit for transistor size may be reached. Thus, it may not be possible to continue the rule of Moore's law and doubling the clock rate for every three years. and also, Present CMOS technology faces consequences like leakage current, power dissipation, oxide thickness, delay, area utilization etc. In order to Overcome the limitations of CMOS-VLSI design an alternative approach called Quantum dot Cellular Automata (QCA) is developed. In a transistor based system some circuits (logic gates) perform computation and wires are used for signal transfer and communication. In contrast QCA technology is able to perform computation and communication simultaneously. The interconnects designed with QCA would be faster and would work almost up to the speed of the processing device which would drastically enhance the system performance. The main block of QCA technology is Majority Gate. Majority gates are fundamental components in digital circuitry, known for their simplicity and efficiency.

So, In this project, we explore the utilization of majority gates in constructing the ALU, aiming to achieve compactness, low power consumption, and high performance. The proposed design will have better parameters compared to the traditional ALU. It produces less power dissipation, lesser delay and better utilization of Area. Each and every operation containing digital circuits should have the majority gate as the fundamental block.

CHAPTER 3

PROBLEM STATEMENT

3.1 Problem Statement

- ❖ Traditionally, Arithmetic Logic Units are implemented using complex combinations of logic gates, face challenges related to circuit complexity, power consumption, and scalability and often result in bulky circuitry with high power requirements, limiting their applicability in resource-constrained environments and inhibiting further miniaturization efforts.
- ❖ Therefore, By using majority gates, we seek to develop an ALU that offers significant improvements in terms of power efficiency, speed and scalability compared to traditional implementations.

3.1.1 Objectives:

- i. **Power Efficiency:** Majority gates inherently offer potential improvements in power efficiency compared to traditional logic gates, owing to their simpler structure and reduced transistor count, the goal is to reduce power consumption compared to traditional approaches while maintaining high performance.
- ii. **Area Utilization:** ALUs using majority gates offer potential area savings compared to traditional ALU designs, especially for certain operations where majority gates can effectively replace multiple basic logic gates. This method of designing ALU can reduce the area utilization and ensures scalability.
- iii. **Performance:** Majority gates can potentially offer performance benefits in terms of propagation delay since they operate on a principle of parallelism. By leveraging majority gates, ALU aims to achieve minimal propagation delays and fast operation, enabling rapid execution of computational tasks.
- iv. **Complexity Reduction:** Traditional ALU designs involve intricate arrangements of logic gates, leading to complex circuitry that is difficult to analyze, optimize, and scale. By leveraging the simplicity and versatility of majority gates, this project seeks to streamline the ALU design process, reducing circuit complexity and enhancing overall efficiency.

CHAPTER 4

METHODOLOGY

4.1 Implementation Methodology

1. **Define Operations:** Firstly, define the arithmetic and logical circuits that need to be designed using majority gates to design ALU. This typically includes addition, subtraction, multiplication, and, or, xor, etc.
2. **Design Logic Circuits for Operations:** For each operation, design the Arithmetic and logic circuits using majority gates. For example, for addition, design a circuit for adding two binary numbers.
3. **Integrating Circuits to form ALU:** Implement control logic/selection line to select the desired operation based on the input control signals and integrate all the circuits to Multiplexer to design ALU.
4. **Simulation:** Now, Create a testbench for the designed ALU and simulate to verify the Waveforms obtained and check the RTL schematic diagram. Test your ALU thoroughly to ensure that it performs the desired operations correctly.
5. **Synthesis & Implementation:** Synthesis is the process of translating RTL design of ALU to Gate-level netlist, whereas implementation involves the physical realization of ALU. This synthesis and implementation process ensures that the majority gate-based ALU design meets specified performance, area, and power requirements.
6. **Optimization:** Estimate the area utilization and power consumption of the implemented design to assess its resource requirements and potential impact on the overall system and analyze the timing constraints and perform timing optimization to ensure that the design meets the required timing specifications.

4.1.1 Software Description:

Tool Required: Xilinx Vivado 2016.4

Language used: Verilog HDL

Xilinx Vivado 2016.4 is an integrated development environment (IDE) used for designing, synthesizing, implementing, and verifying digital circuits targeting Xilinx FPGAs and SoCs. It supports various design entry methods, synthesis techniques, implementation algorithms, verification tools, and IP integrations. Vivado 2016.4 offers features for power analysis, debugging, scripting, and automation. It supports a wide range of Xilinx FPGA families and devices and provides extensive documentation and community support.

4.2 Circuit Diagram of 4-bit ALU

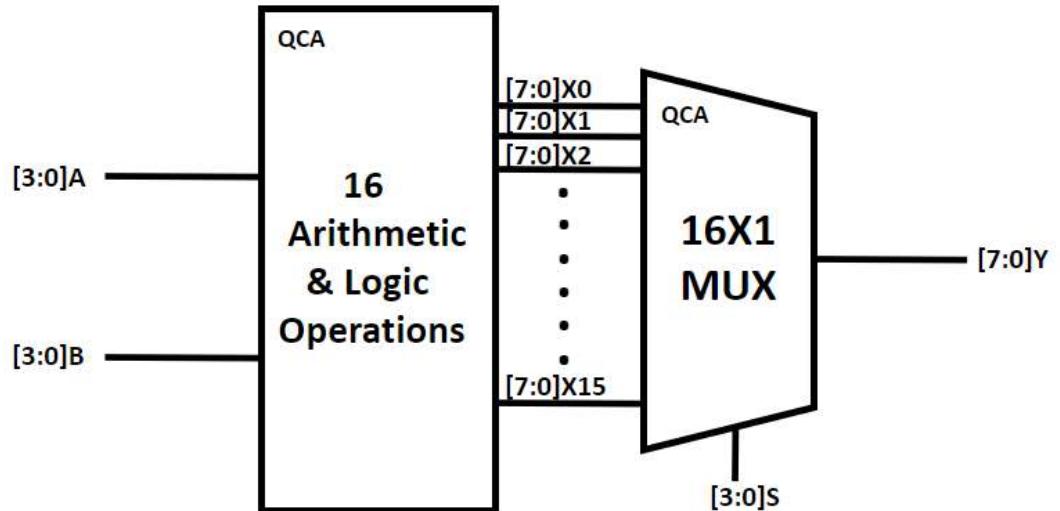


Fig 4.1: Circuit Diagram of 4-bit ALU

4.3 Block Diagram

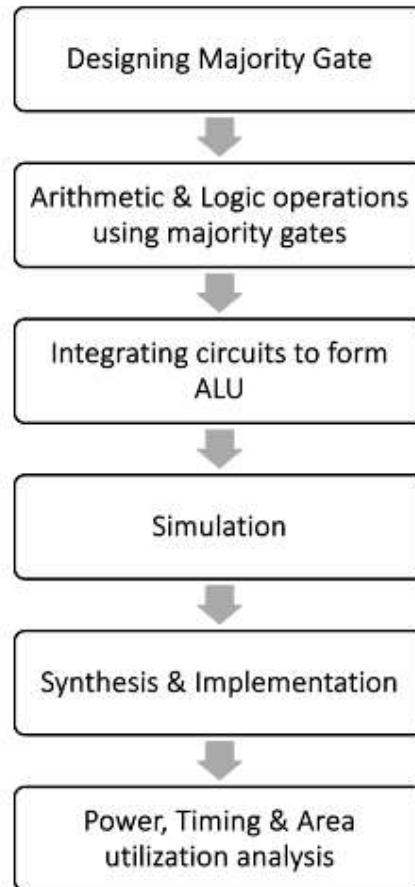


Fig 4.2: Block Diagram

4.4 Proposed Designs using Majority gates

4.4.1 4-bit Adder:

Adder is the basic operation required for ALU. It is the central to many digital circuits which perform addition and subtraction operations.

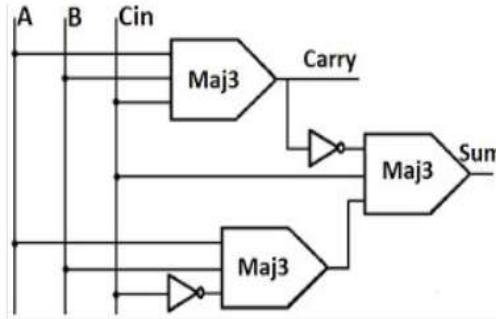


Fig 4.3: Full Adder using Majority gates

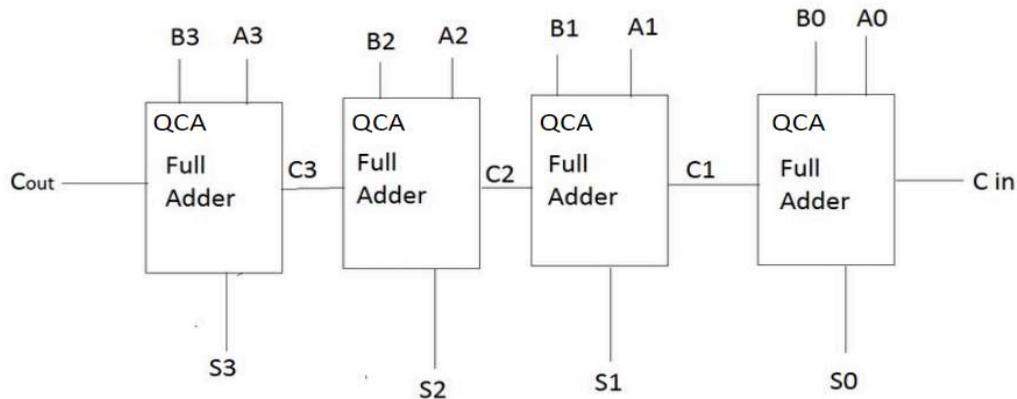


Fig 4.4: 4-bit Ripple Carry Adder using Majority gates

4.4.2 4-bit Subtractor:

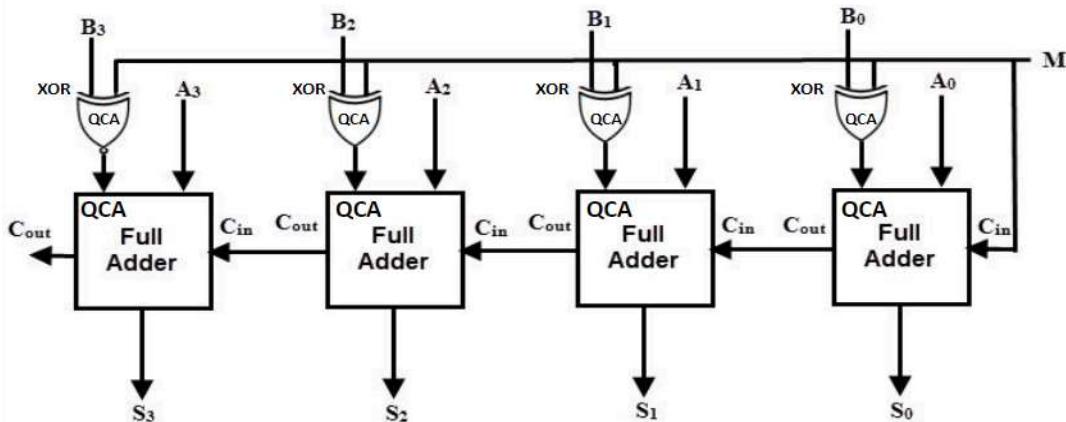


Fig 4.5: 4-bit Subtractor using Majority gates

4.4.3 4-bit Multiplier:

Array multipliers employ an array of adders to represent the direct implementation of manual multiplication. This type of multiplier is easy to implement, but requires a big area of design that increases proportionally with the increase in size of the multiplied operands. But by utilizing majority gates, we can reduce the area consumption,

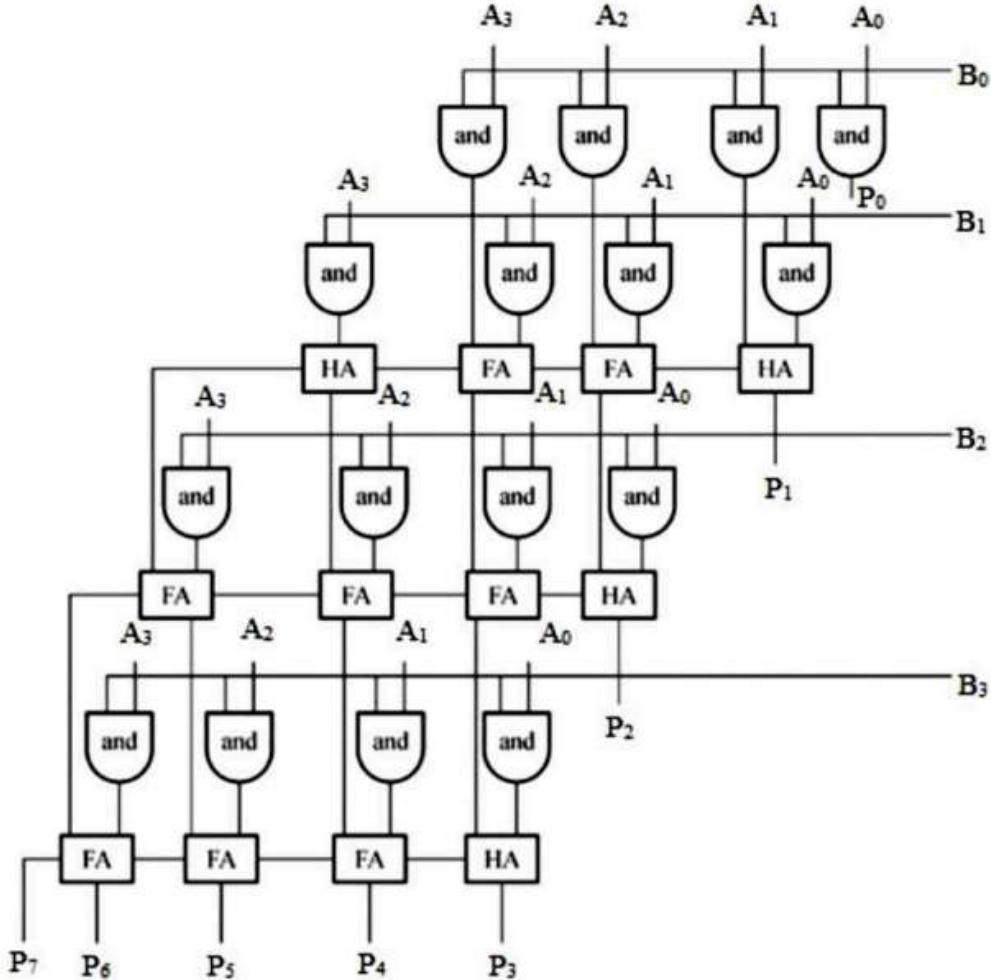


Fig 4.6: 4-bit Multiplier using Majority gates

4.4.4 4-bit Divider:

The division operation is carried out by subsequent stages where each stage determines a quotient bit. Thus it is called an array divider. This Divider produces a quotient and remainder in the output stage. Division is possible for both signed and unsigned numbers.

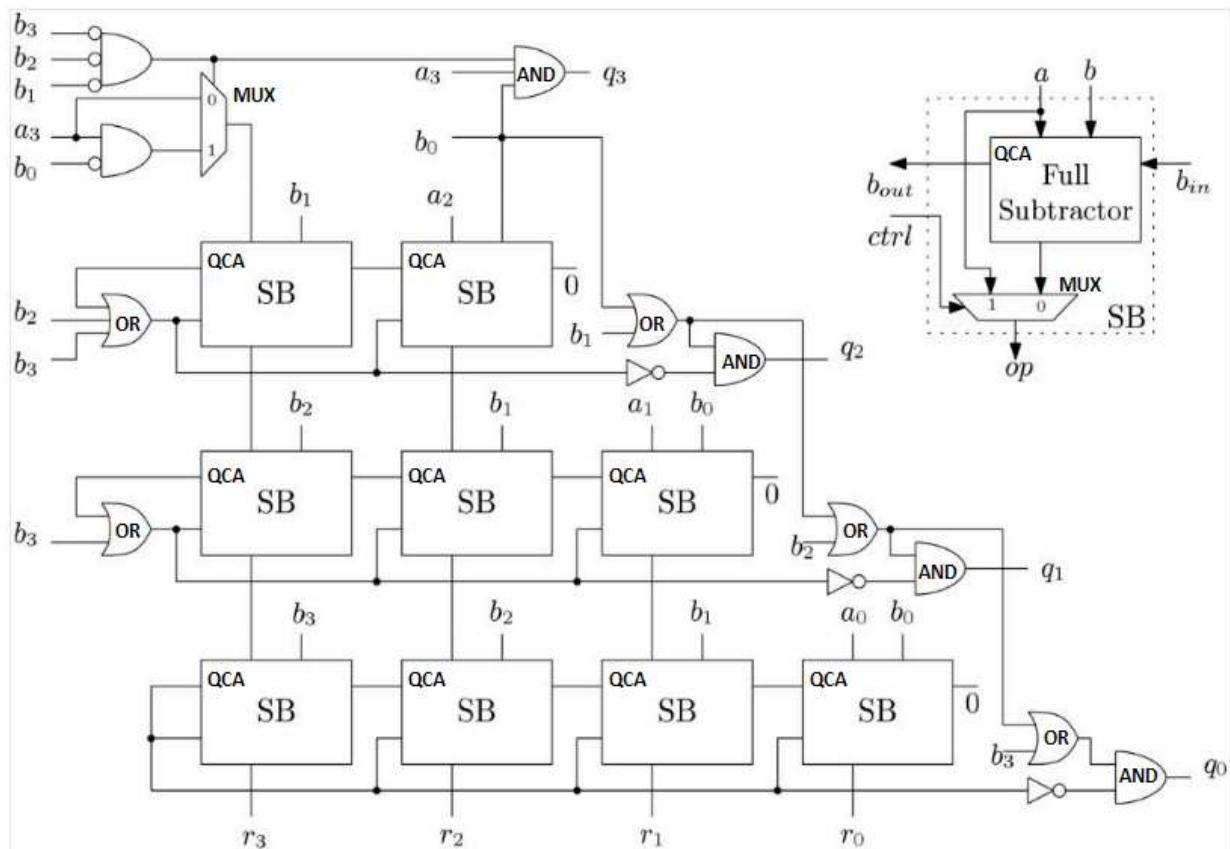


Fig 4.7: 4-bit Divider using Majority Gates

4.4.5 Shift Left Operator:

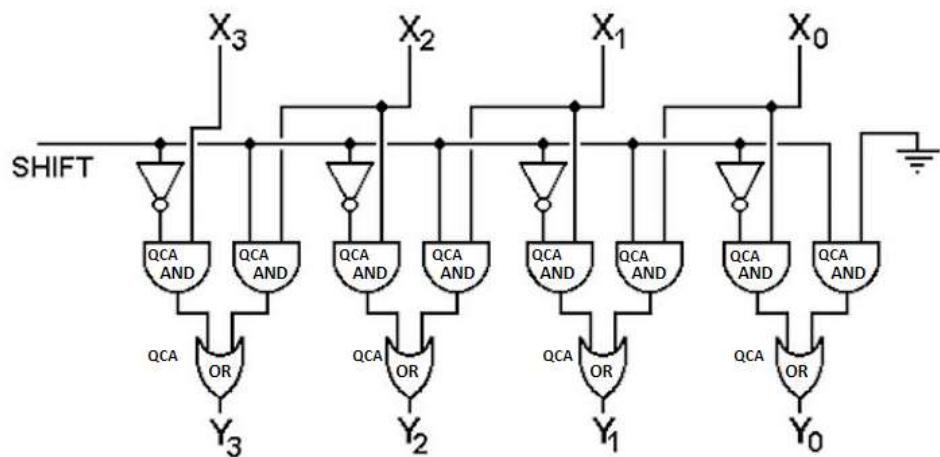


Fig 4.8: Shift Left Operator using Majority gates

4.4.6 Shift Right Operator:

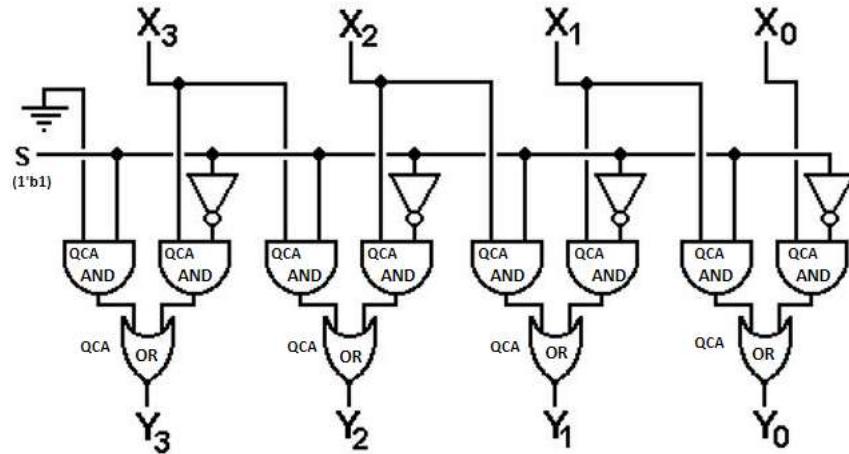


Fig 4.9: Shift Right Operator using Majority gates

4.4.7 Rotate Left Operator:

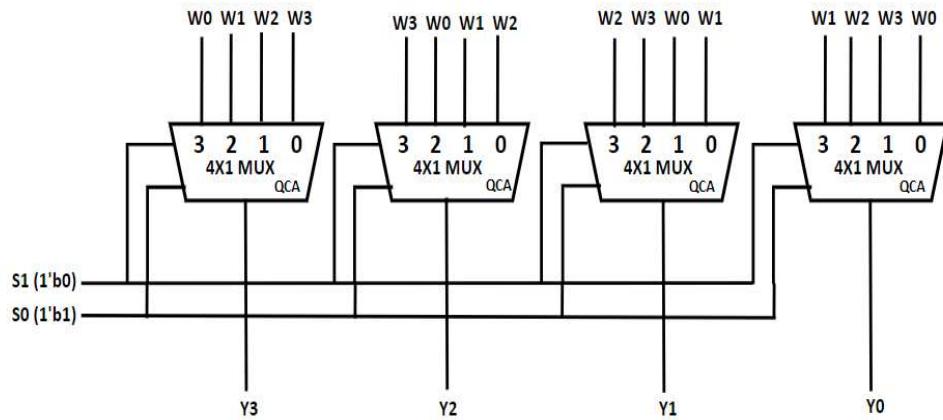


Fig 4.10: Rotate Left Operator using Majority gates

4.4.8 Rotate Right Operator:

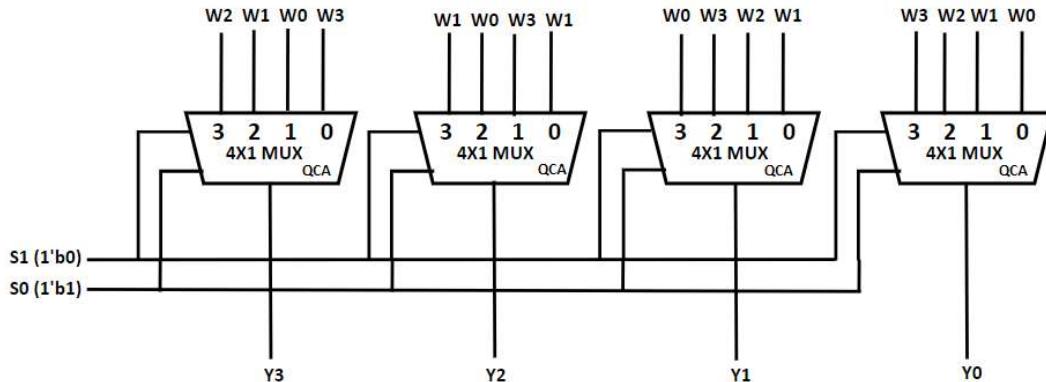


Fig 4.11: Rotate Right Operator using Majority gates

4.4.9 Logic Gates:

1) AND Gate Using Majority Gate

From the Eq(1), If Single input is set to $P = -1$ i.e, $C=0$. Eq(2) represents AND Logic Gate.

$$\text{If } C \Rightarrow 0, F = A \cdot B + B \cdot 0 + 0 \cdot A = A \cdot B \quad (2)$$

2) OR Gate Using Majority Gate

From the Eq(1), If Single input is set to $P = +1$ i.e, $C=1$. Eq(3) represents OR Logic Gate.

$$\text{If } C \Rightarrow 1, F = A \cdot B + B \cdot 1 + 1 \cdot A = A \cdot B + A + B = A(1+B) + B = A + B \quad (3)$$

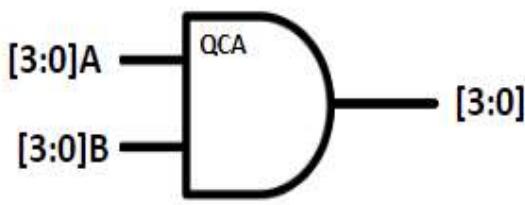


Fig 4.12: 4-bit AND gate using MG

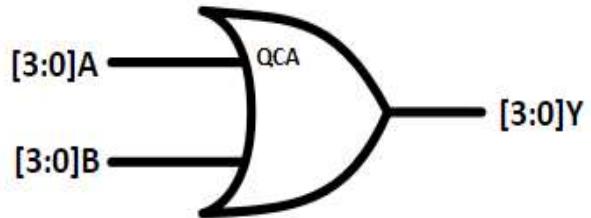


Fig 4.13: 4-bit OR gate using MG

3) NAND Gate using Majority Gate

It is designed by adding inverter to AND Gate

4) NOR Gate using Majority Gate

It is designed by adding inverter to OR Gate

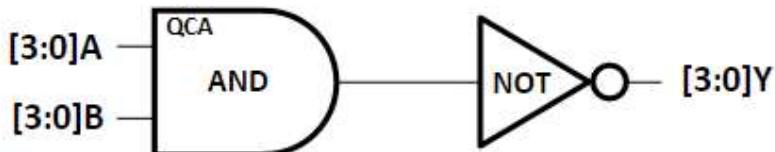


Fig 4.14: 4-bit NAND gate using MG

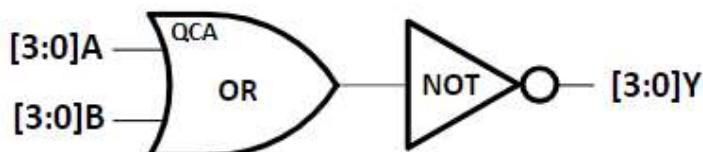


Fig 4.15: 4-bit NOR gate using MG

- 5) NOT Gate is designed using a single bit inverter
 6) XOR Gate using Majority Gate

It is designed by using 3 majority gates and 2 inverters

- 7) XNOR Gate using Majority Gate

It is designed by adding inverter to XOR Gate

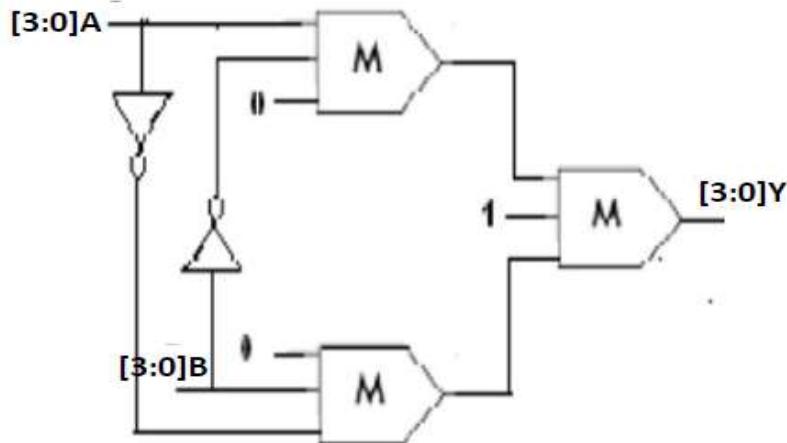


Fig 4.16: 4-bit XOR gate using MG

4.4.10 4-bit Equality Comparator:

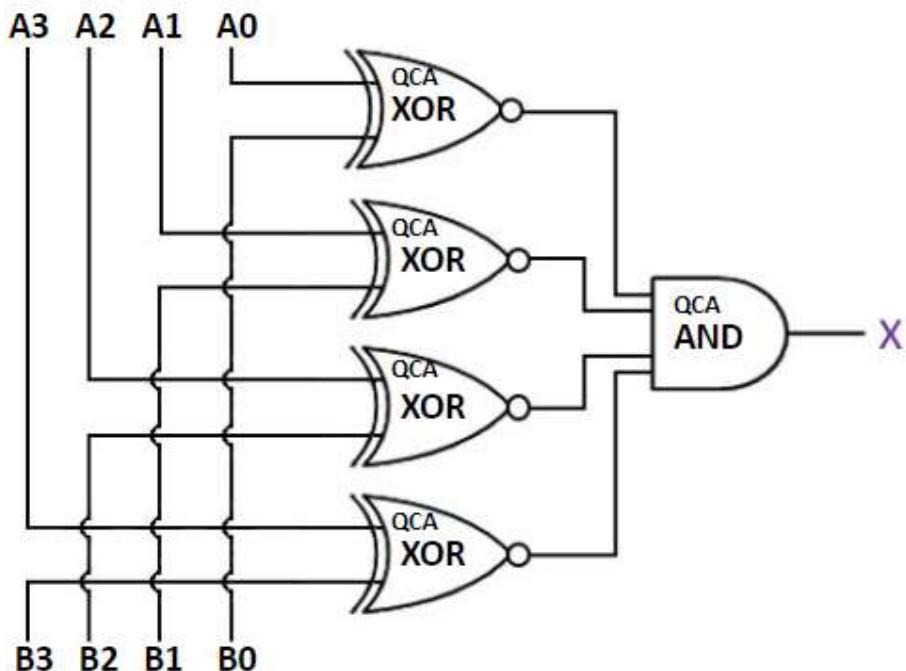


Fig 4.17: 4-bit Equality Comparator using MG

CHAPTER-5

RESULTS AND DISCUSSION

Traditional ALU (Existing Design)

5.1 ALU Design using Logic Gates (Structural Modelling)

After designing ALU in Verilog, we need to write testbench code to verify whether it performs the desired operations correctly.

Here, In my testbench Code, the Input values are:

- ❖ Input1: a = 14 (1110)
- ❖ Input2: b = 11 (1011)
- ❖ Selection line/Control Input: s = 0 to 15
- ❖ Output Y (8bit)

5.1.1 Simulation Waveform:

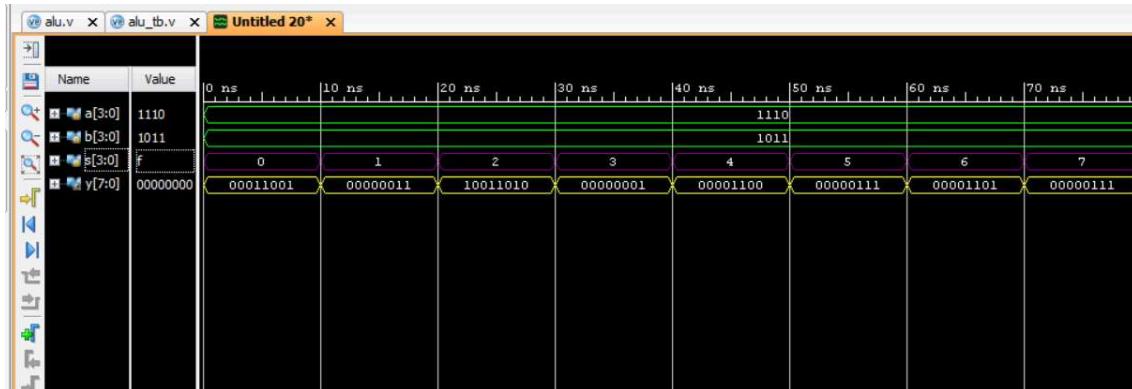


Fig 5.1: Simulation waveform of ALU (1-8 operations)

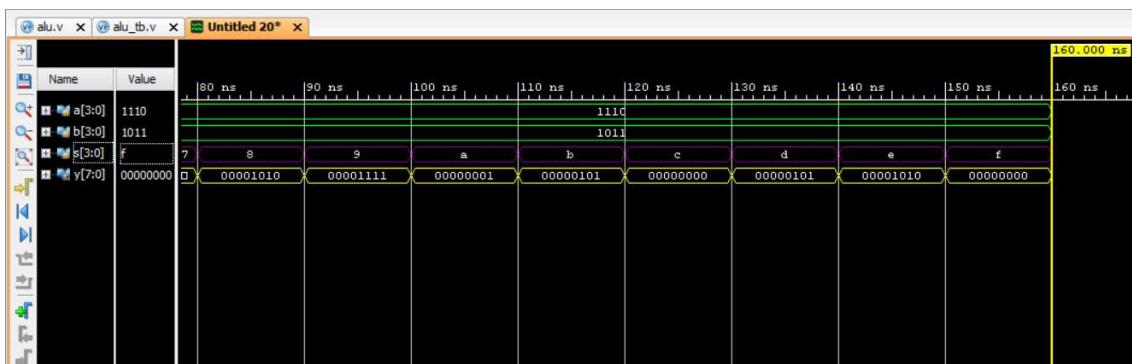


Fig 5.2: Simulation waveform of ALU (9-16 operations)

5.1.2 TCL Console:

Sel	Operation	Sel	Operation
0	Addition	8	AND
1	Subtraction	9	OR
2	Multiplication	10	NOT
3	Division	11	NAND
4	Shift Left	12	NOR
5	Shift Right	13	XOR
6	Rotate Left	14	XNOR
7	Rotate Right	15	Equality Comparator

Table 5.1: Defining Operations of ALU

```
# run 1000ns
a:1110 b:1011 Sel: 0 Output:00011001
a:1110 b:1011 Sel: 1 Output:00000011
a:1110 b:1011 Sel: 2 Output:10011010
a:1110 b:1011 Sel: 3 Output:00000001
a:1110 b:1011 Sel: 4 Output:00001100
a:1110 b:1011 Sel: 5 Output:00000111
a:1110 b:1011 Sel: 6 Output:00001101
a:1110 b:1011 Sel: 7 Output:00000111
a:1110 b:1011 Sel: 8 Output:00001010
a:1110 b:1011 Sel: 9 Output:00001111
a:1110 b:1011 Sel:10 Output:00000001
a:1110 b:1011 Sel:11 Output:00000101
a:1110 b:1011 Sel:12 Output:00000000
a:1110 b:1011 Sel:13 Output:00000101
a:1110 b:1011 Sel:14 Output:00001010
a:1110 b:1011 Sel:15 Output:00000000
```

Fig 5.3: TCL Console

5.1.3 Schematic Diagram:

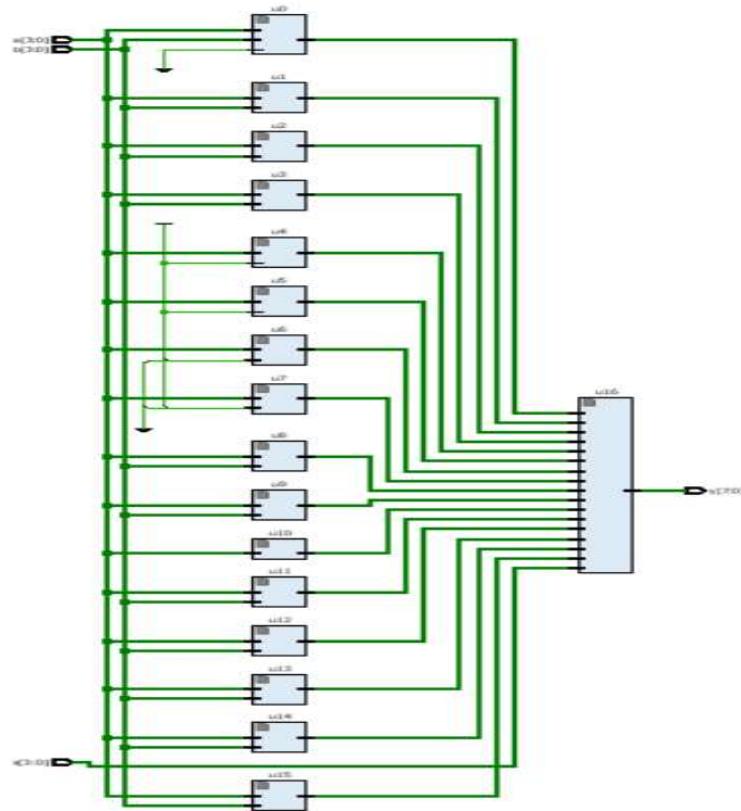


Fig 5.4: Schematic Diagram of ALU

5.1.4 Synthesis & Implementation:

❖ Power:

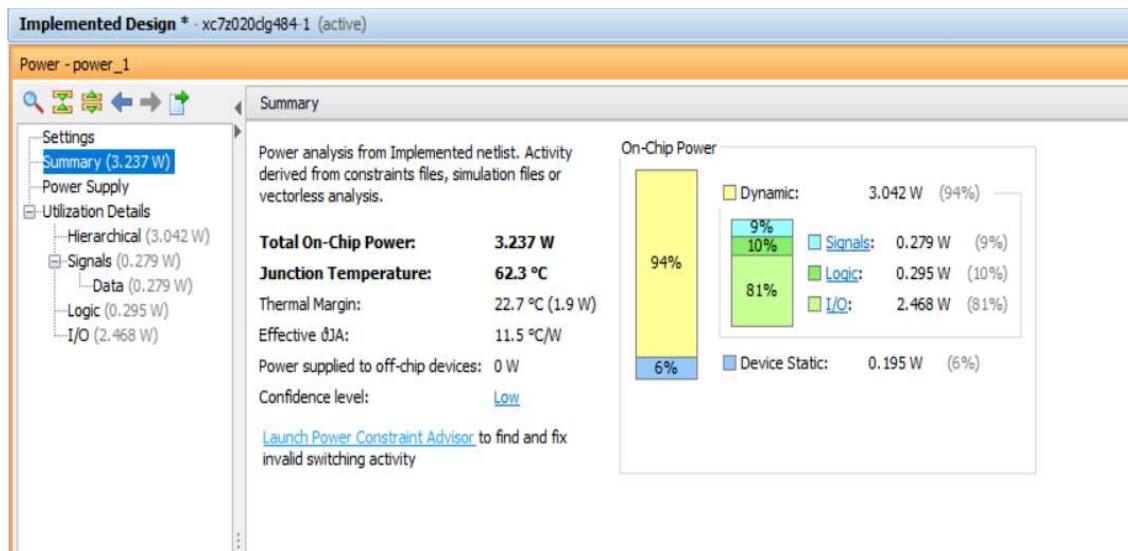


Fig 5.5: Power analysis of traditional ALU

❖ Timing Summary:

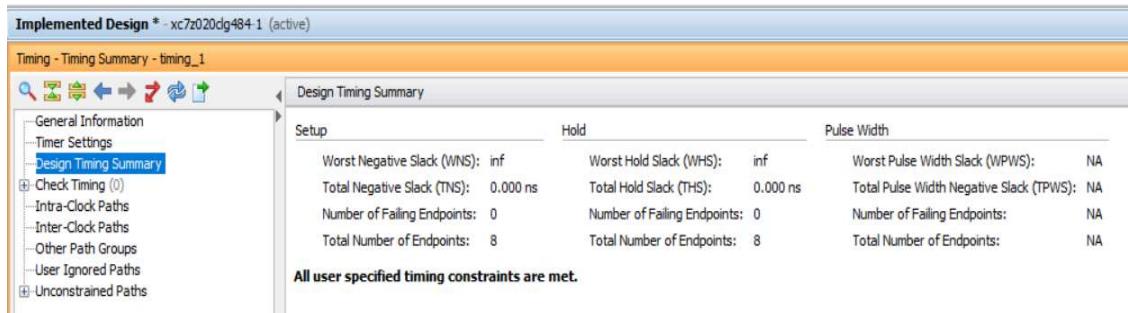


Fig 5.6: Timing Summary of traditional ALU

❖ Utilization:

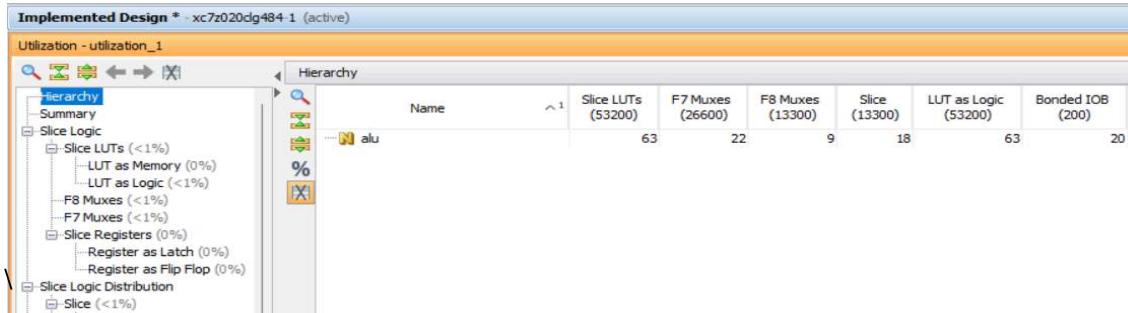


Fig 5.7: Area utilization of traditional ALU

5.2 ALU Design using Case Statement (Behavioural Modelling)

5.2.1 Schematic Diagram:

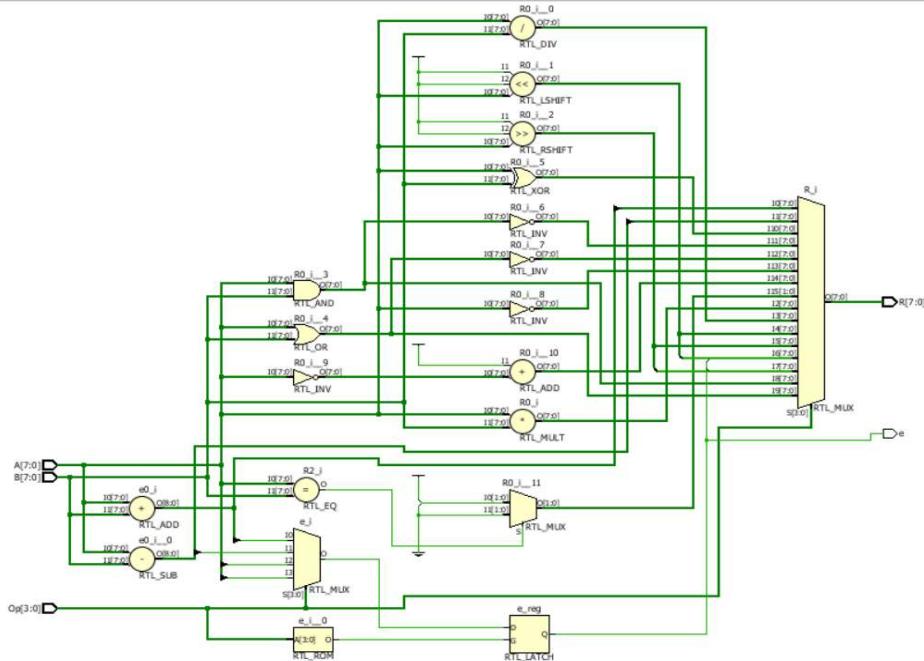


Fig 5.8: Schematic Diagram of ALU (using case statement)

5.2.2 Synthesis & Implementation:

❖ Power:

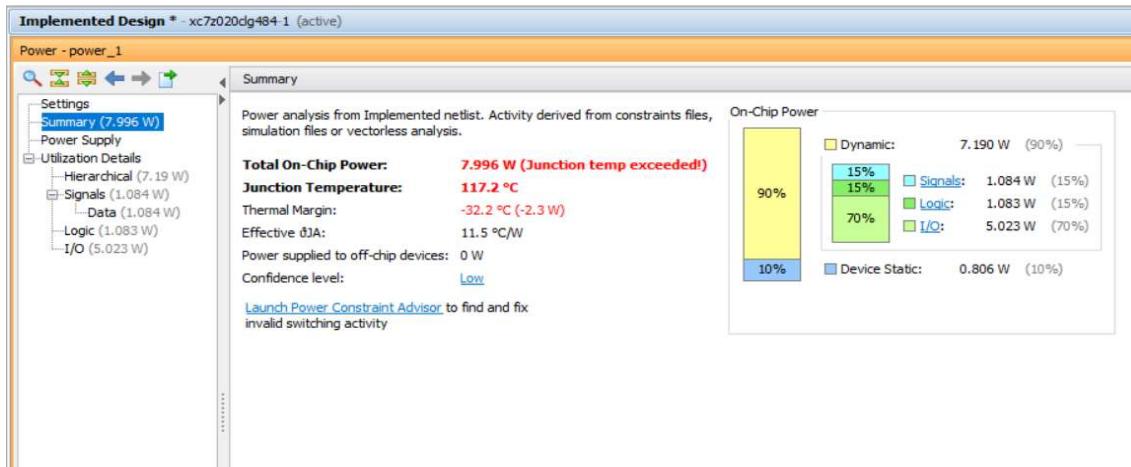


Fig 5.9: Power Analysis of ALU (using case statement)

❖ TimingSummary:

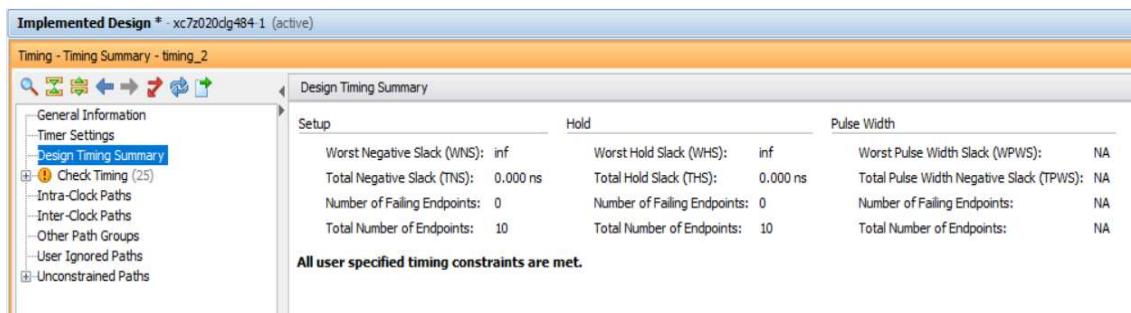


Fig 5.10: Timing Summary of ALU (using case statement)

❖ Utilization:

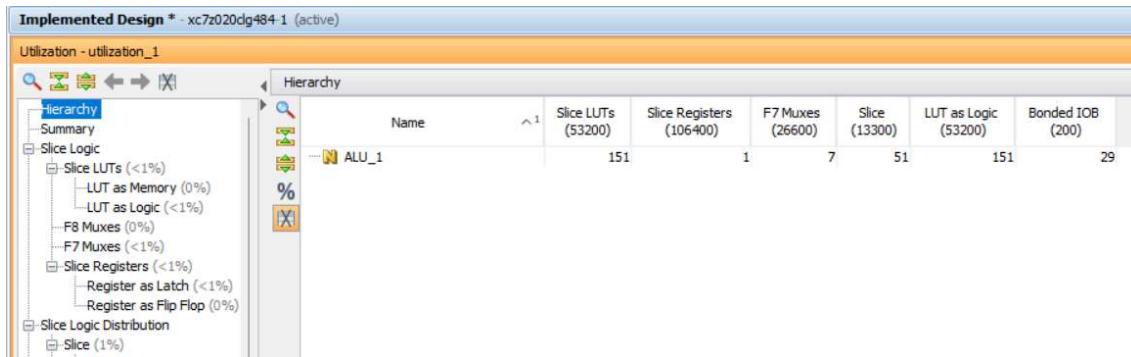


Fig 5.11: Area utilization of ALU (using case statement)

Proposed Design of ALU

5.3 ALU Design using Majority Gates

5.3.1 Synthesis & Implementation:

❖ Power:

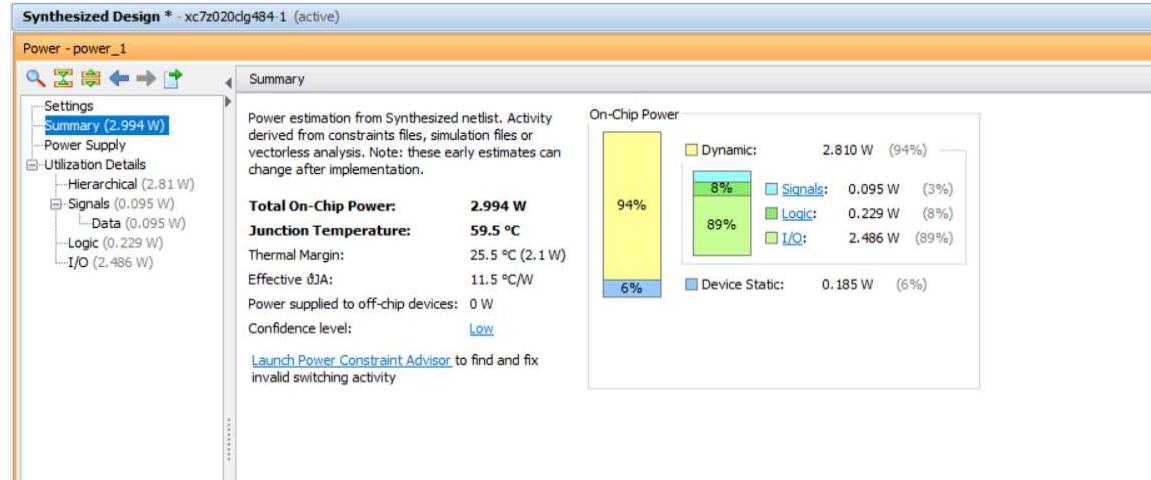


Fig 5.12: Power Analysis of ALU (using Majority Gates)

❖ Timing Summary:

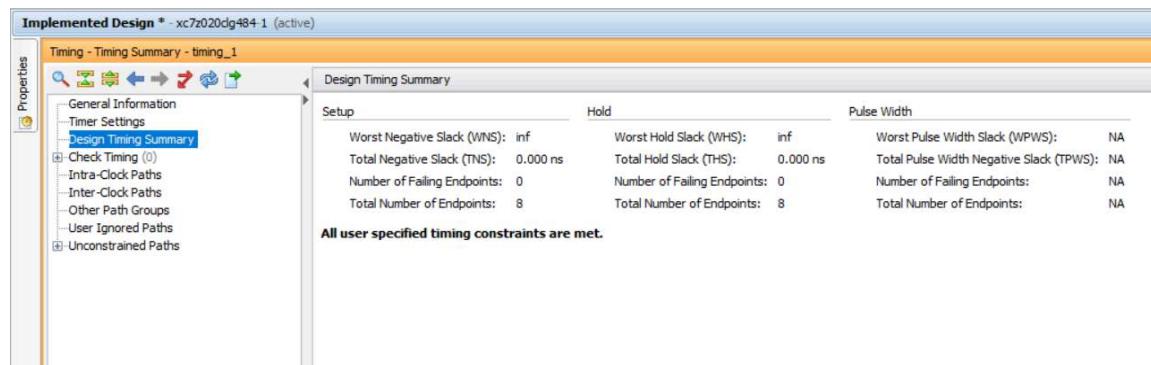


Fig 5.13: Timing Summary of ALU (using Majority Gates)

❖ Utilization:

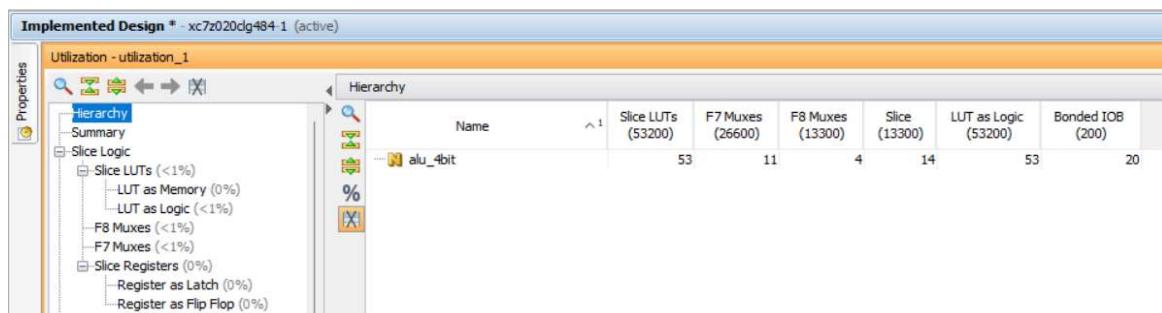


Fig 5.14: Area utilization of ALU (using Majority Gates)

5.4 Comparison of Power, Timing, Area Utilization

5.4.1 Simulation Parameters of Traditional ALU:

Power	Total On-Chip Power: 3.237W; Junction Temp: 62.3°C Dynamic Power: 3.042W (94%) Device Static: 0.195W (6%)
Timing	Total no. of end points: 8 (setup, hold, pulse width) All user specified timing constraints are met.
Utilization(Area)	Slice LUTs (<1%): 63 Bounded IOB: 20

Table 5.2: Power, Timing, Area of ALU (using logic gates)

5.4.2 Simulation Parameters of ALU (using case statement):

Power	Total On-Chip Power: 7.996W; Junction Temp: 117.2°C (Junction Temp exceeded!) Dynamic Power: 7.190W (94%) Device Static: 0.806W (6%)
Timing	Total no. of end points: 10 (setup, hold, pulse width) All user specified timing constraints are met.
Utilization(Area)	Slice LUTs (<1%): 151 Bounded IOB: 29

Table 5.3: Power, Timing, Area of ALU (using case statement)

5.4.3 Simulation Parameters of ALU (using Majority Gates):

Power	Total On-Chip Power: 2.994W; Junction Temp: 59.5°C Dynamic Power: 2.810W (94%) Device Static: 0.185W (6%)
Timing	Total no. of end points: 8 (setup, hold, pulse width) All user specified timing constraints are met.
Utilization(Area)	Slice LUTs (<1%): 53 Bounded IOB: 20

Table 5.4: Power, Timing, Area of ALU (using majority gates)

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

In conclusion, this project marks a significant advancement in the realm of digital design by showcasing the potential of majority gate-based ALUs. Through meticulous research, design, and testing, we've validated the efficiency of this approach in achieving comparable or even superior performance metrics compared to traditional ALU designs. The reduced power consumption, delay and area utilization (scalability) of majority gate-based architectures offer promising prospects for future developments.

Looking ahead, further optimization efforts could refine our designs, enhancing performance and efficiency. Utilizing the design of ALU using Majority gates in Quantum-Dot Cellular Automata (QCA) Technology helps to find the alternative to CMOS technology and overcomes the limitations such as Power dissipation, Current leakage, Area utilization, etc. Additionally, real-world hardware implementations would validate our findings and pave the way for practical deployment in various computing systems.

In essence, this project not only serves as a testament to the versatility and adaptability of majority gate-based ALUs but also sets the stage for continued innovation and exploration in the fascinating field of digital design and computing.

REFERENCES

- 1) K. Walus; G. Schulhof; G.A. Jullien; R. Zhang; W. Wang,"Circuit design based on majority gates for applications with quantum-dot cellular automata", Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.
- 2) V M Nanditha; U Sanath Rao; Manasa Murali; R Swathi; K Chethana,"Design and Analysis of Digital Circuits using Quantum Cellular Automata and Verilog", 2020 7th International Conference on Computing for Sustainable Global Development.
- 3) J. Maharaj and S. Muthurathinam, "Effective RCA design using quantum dot cellular automata", *Microprocessors and Microsystems*, vol. 73, pp. 102964, 03 2020.
- 4) P. D. Tougaw and C. S. Lent, "Logical devices implemented using quantum cellular automata", *Journal of Applied Physics*, vol. 75, pp. 1818, 1994.
- 5) IJSRD - International Journal for Scientific Research & Development| Vol. 9, Issue 4, 2021 | ISSN (online): 2321-0613
- 6) H. Rashidi, A. Rezai, and S. Soltany, "High-performance multiplexer architecture for quantum-dot cellular automata," *J. Comput. Electron.*, vol. 15, no. 3, pp. 968– 981, 2016.
- 7) Seyed-Sajad Ahmadpour, Mohammad Mosleh & Saeed Rasouli Heikalabad,"The design and implementation of a robust single-layer QCA ALU using a novel fault-tolerant three-input majority gate", Volume 76, pages 10155–10185, (2020).
- 8) M. G. Waje and P. K. Dakhole, "Design and implementation of 4-bit arithmetic logic unit using Quantum Dot Cellular Automata," 2013, pp. 1022–1029.