[GROUP 7:1 BIT ALU]

Abstract

This paper presents the design and implementation of a 1-bit Arithmetic Logic Unit (ALU) employing a novel approach that integrates five 2-to-4 decoders and additional logic gates. The 1-bit ALU serves as a fundamental building block in digital circuits, facilitating essential arithmetic and logical operations. The utilization of multiple 2-to-4 decoders offers a unique strategy for input selection, allowing dynamic routing of data bits based on control signals. Complemented by carefully designed additional logic gates, the ALU can perform various operations, including addition, subtraction, AND, OR .

The proposed design leverages the inherent capabilities of 2-to-4 decoders to efficiently decode control signals, directing input bits to specific arithmetic or logical pathways. Additional logic gates, including AND, OR, and NOT gates, are strategically incorporated to realize the desired operations on the selected inputs. This approach contributes to a modular and scalable design, offering flexibility and ease of integration into larger digital systems.

Simulation results demonstrate the functionality and reliability of the 1-bit ALU across diverse input combinations and control signals. The design's efficiency is further highlighted through its ability to execute a range of operations while minimizing hardware complexity. The modular nature of the implemented design, along with the use of 2-to-4 decoders, makes it suitable for various applications, including microprocessors, digital signal processors, and other integrated circuits.

This work not only provides insights into an unconventional 1-bit ALU design but also underscores the potential of harnessing decoder-based input selection for enhanced flexibility and modularity in digital circuitry. Future work may explore optimizations, scalability for multi-bit ALUs, and integration into more complex digital architectures.

1. Introduction

1-BIT ALU:

In the realm of digital circuit design, the Arithmetic Logic Unit (ALU) stands as a fundamental component, indispensable for executing arithmetic and logical operations in computing systems. This paper introduces a novel approach to the design of a 1-bit ALU, leveraging the distinctive capabilities of five 2-to-4 decoders along with additional logic gates. This unconventional configuration aims to enhance the ALU's versatility, modularity, and efficiency in performing a range of operations.

Traditional ALU designs often rely on multiplexers and logic gates for input selection and operation execution. However, the integration of 2-to-4 decoders in our approach offers a fresh perspective. By dynamically decoding control signals, these decoders serve as key elements for selecting input pathways, providing a unique mechanism for directing data bits within the ALU. This, coupled with carefully designed additional logic gates, empowers the ALU to perform not only fundamental arithmetic operations like addition and subtraction but also various logical manipulations, including AND, OR, XOR, and logical shifts.

The significance of this novel design lies not only in its ability to execute diverse operations but also in its modular and scalable nature. The paper elucidates how the use of five 2-to-4 decoders contributes to an adaptable and compact ALU architecture, suitable for integration into larger digital systems. The modularity inherent in this design facilitates future extensions or modifications, making it an appealing solution for applications ranging from microprocessors to digital signal processors.

This introductory section sets the stage for exploring the intricacies of the proposed 1-bit ALU design. The subsequent sections will delve into the methodology, implementation details, simulation results, and potential future directions. Through this exploration, we aim to showcase the advantages and capabilities of our unconventional approach in advancing the field of digital circuit design and contributing to the evolution of Arithmetic Logic Units.

2. Problem Statement

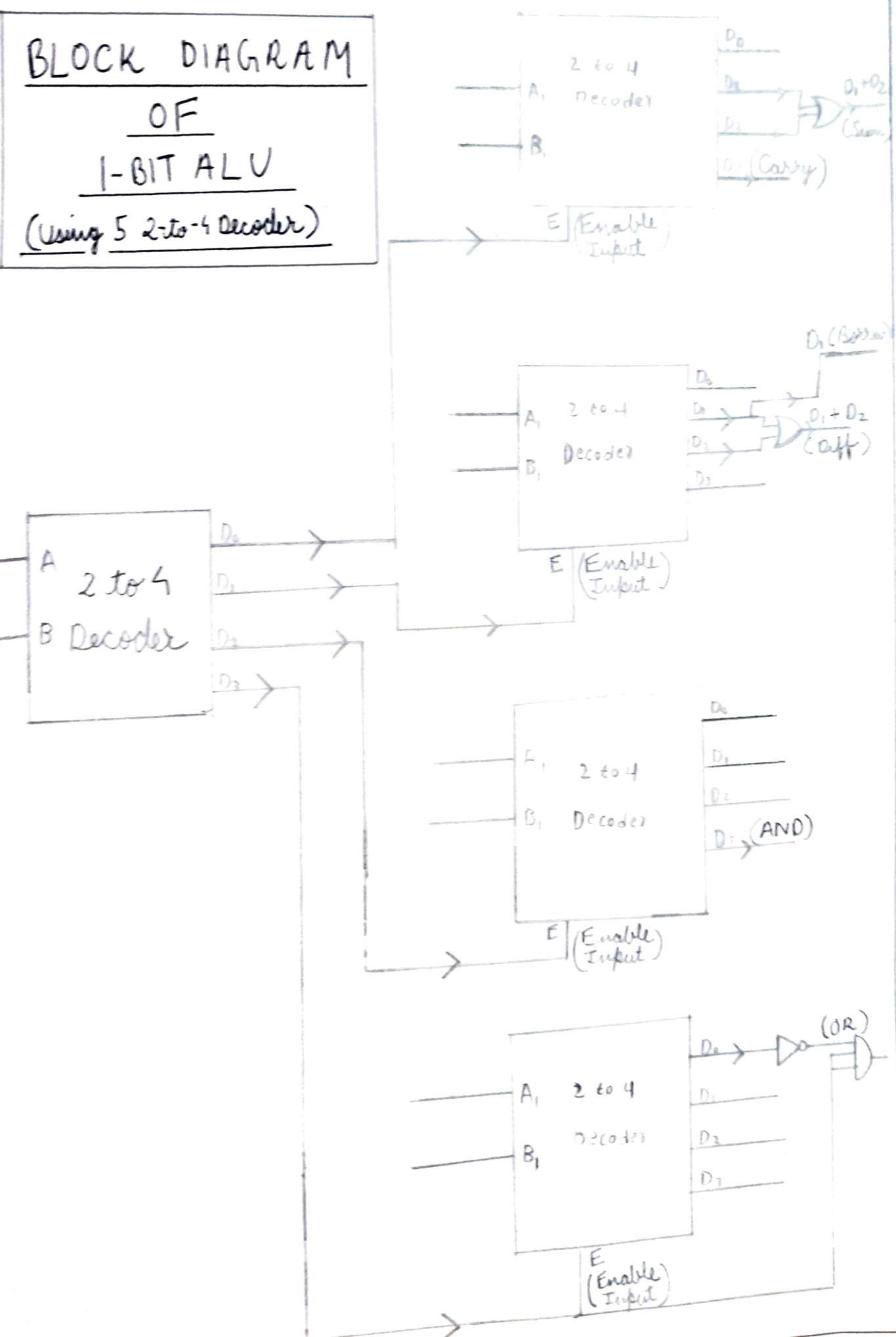
Design a 1 bit ALU that performs the following operations on two 1 bit numbers A and B, based on the values of two bit control input.
Control input Operation of ALU :-

SL.NO.	CONTROL INPUT	OPERATION OF ALU
1.	00	Addition
2.	01	Subtraction
3.	10	Logical AND
4.	11	Logical OR

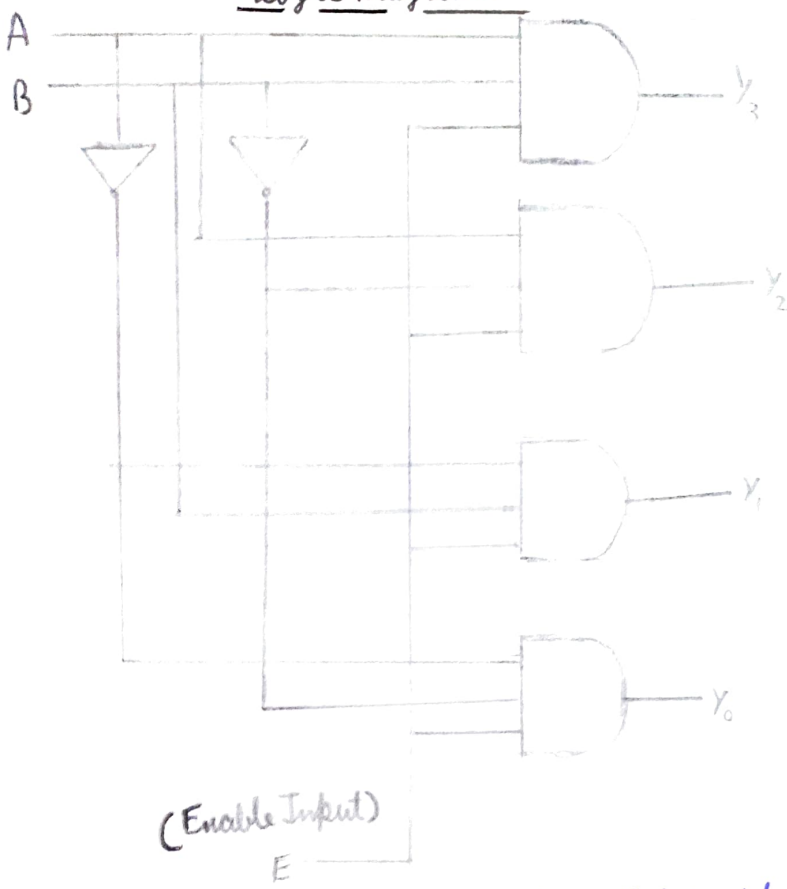
At a time, the ALU will take two 1 bit numbers and perform any one operation according to its control line inputs and produce a two bit output (01 00) . 01 bit will produce the carry value for addition or borrow value for subtraction or 0 for AND and OR operation. 00 bit will produce the sum value for addition or difference value for subtraction or result of AND and OR operation. Design the circuit using five 2-to-4 line decoders, where one decoder will be used for providing the control input values for each operation and remaining 4 decoders will be used to produce a 2 bit output corresponding to each operation according to the control input values generated from the first decoder.

BLOCK DIAGRAM OF 1-BIT ALU

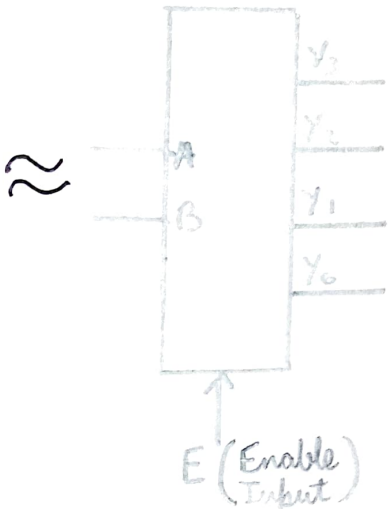
(Using 5 2-to-4 decoder)



Logic Diagram

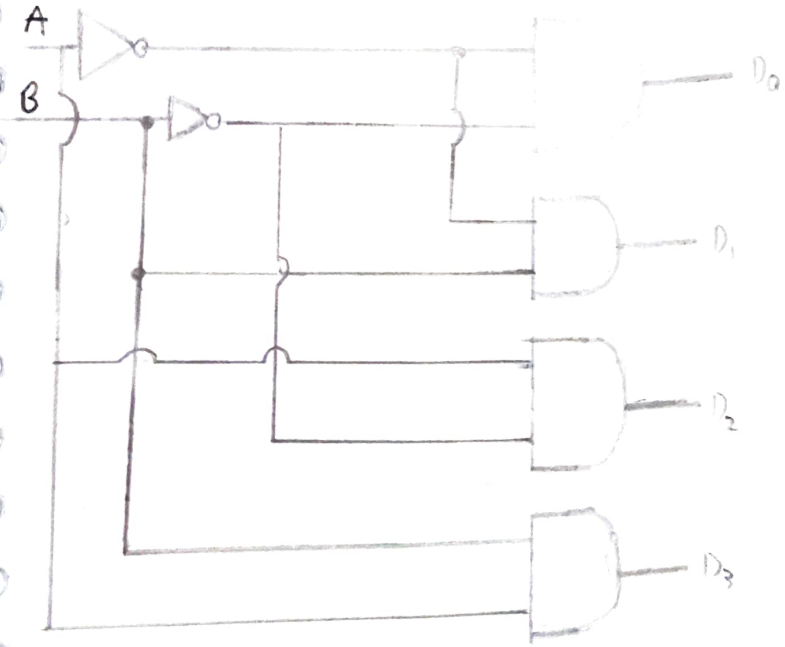


Block Diagram

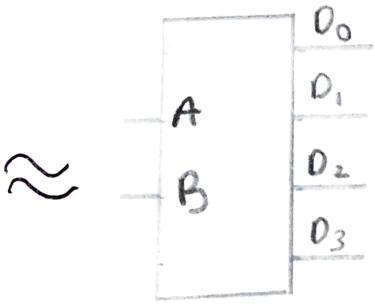


2 to 4 Decoder with E (Enable input)

Logic Diagram



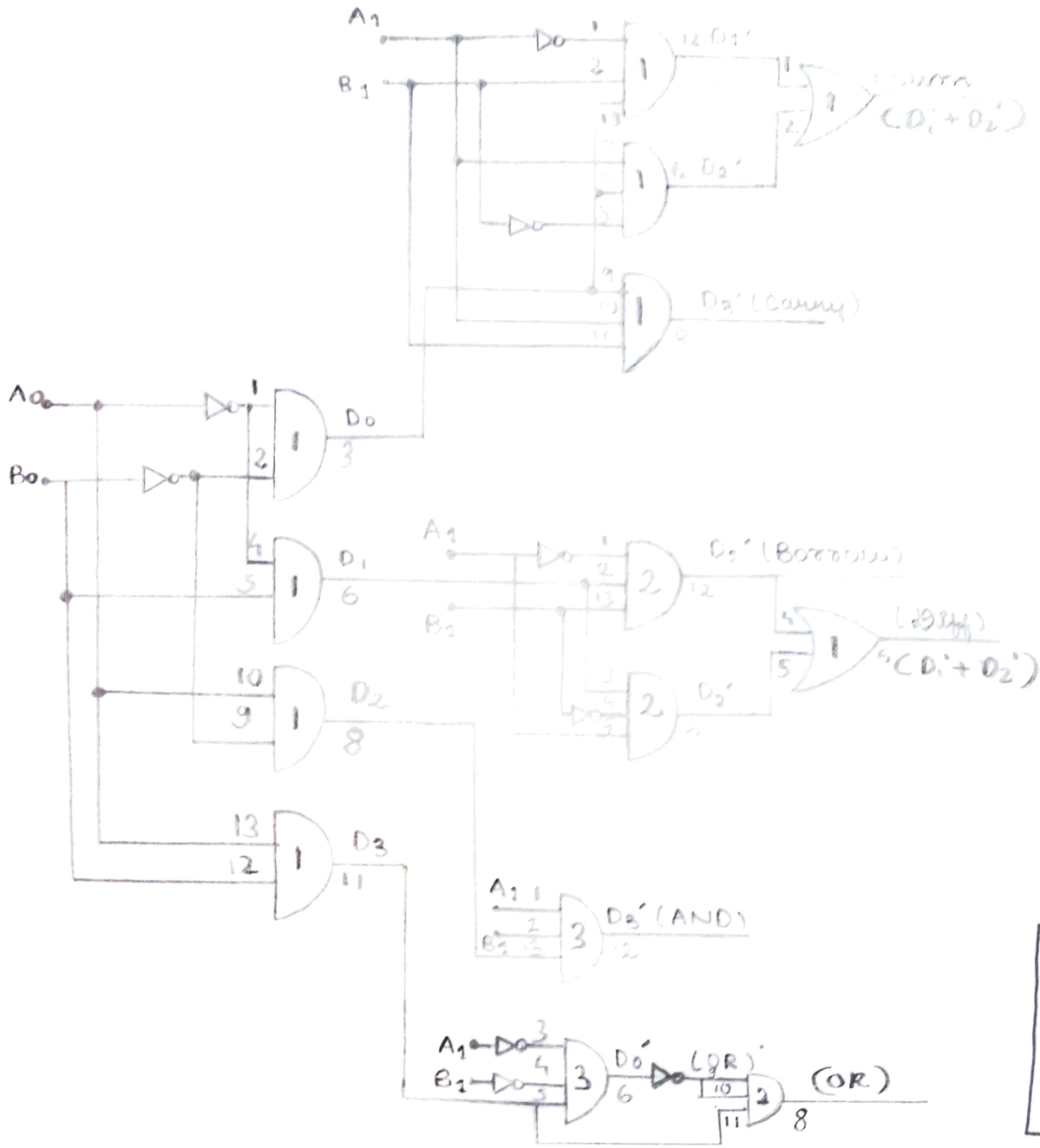
Block Diagram



2 to 4 Decoder (without Enable Input)

1-BIT ALU USING DECODERS

(Simplified Figure)



A_0, B_0
Control Input

A_1, B_1
User Input

Circuit Diagram

(Simplified Figure)

Simplification of Addition operation using decoder

Decoder				Input		Output	
D_0	D_1	D_2	D_3	A_1	B_1	Sum	Carry
1	0	0	0	0	0	0	0
0	1	0	0	0	1	1	0
0	0	1	0	1	0	1	0
0	0	0	1	1	1	0	1

By observing the similarities between D_i's and output we can observe that :- $Sum = D_1 + D_2$

$Carry = D_3$

Simplification of substraction operation using decoder

Decoder				Input		Output	
D_0	D_1	D_2	D_3	A_1	B_1	Difference	Borrow
1	0	0	0	0	0	0	0
0	1	0	0	0	1	1	1
0	0	1	0	1	0	1	0
0	0	0	1	1	1	0	0

By observing the similarities between D_i's and output we can observe that :- $Difference = D_1 + D_2$

$Borrow = D_1$

Note :- The minimization of circuit will be done on observing and removing the used gates or logical operators

Simplification of Logical AND operation using decoder

Decoder				Input		output
D_0	D_1	D_2	D_3	A_1	B_1	F
1	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	1	1	1

By observing the similarities between D_i's and output we can observe that :-

$$F = D_3$$

Simplification of Logical OR operator using Decoder

Decoder				Input		output
D_0	D_1	D_2	D_3	A_1	B_1	F_1
1	0	0	0	0	0	0
0	1	0	0	0	1	1
0	0	1	0	1	0	1
0	0	0	1	1	1	1

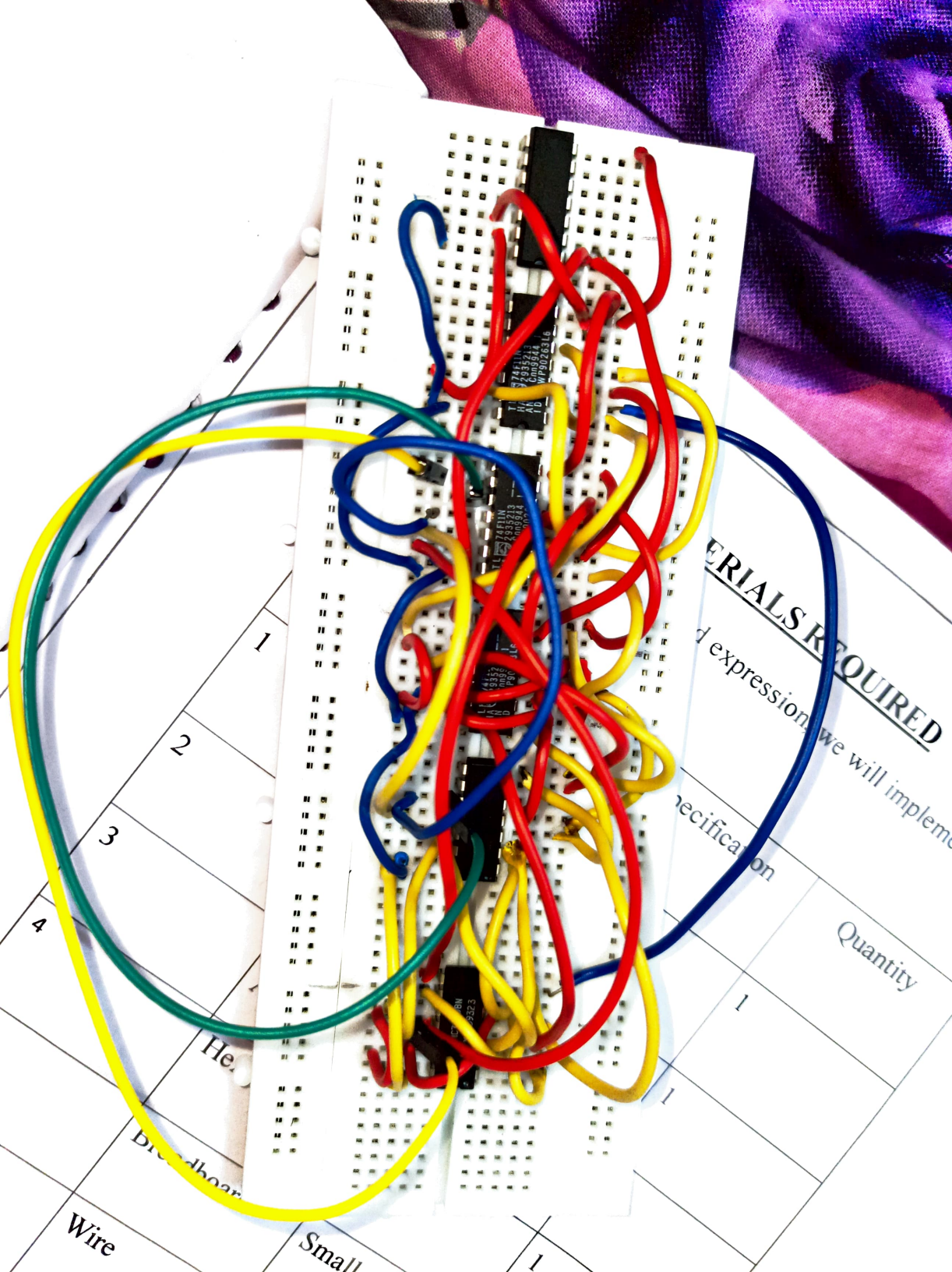
By observing the similarities between D_i's and output we can observe that :-

$$F_1 = \overline{D_0}$$

MATERIALS REQUIRED

Now, as we have got our simplified expression, we will implement it using Logic gates.

Serial no.	Name of the components	Specification	Quantity
1	Quad 2 I/P AND gate	7408	1
2	Quad 2 I/P OR gate	7432	1
3	Quad 3 I/P AND gate	7411	3
4	Hex Inverter	7404	1
5	Breadboard	Small	1
6	Wire		As per Required
7	LED	Single colour	Red-6 (If Required)
8	Battery	6F22 9v	1 (If Required)
9	Switch	SPDT	4 (If Required)
10	Resistors		7 (If Required)


ERIALS REQUIRED
Quantity
1
1
Wire

	Inputs				Outputs					
	A_0	B_0	A_1	B_1	Sum	Carry	Diff	Borrow	OR	AND
IC No	7408	7408	7411	7411	7432	7411	7432	7432	7411	7411
IC Count	1	1	1	1	1	1	1	1	2	2
PIN No	13	12	1	2	3	12	11	13	8	6
	Control input		User input		Addition		Subtraction		OR	AND

Truth Table. (Results)

Control Input		Input		OR	AND	Addition		Substraction	
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	1	0	0	0	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	1	1
0	1	1	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0
A_0	B_0	A_1	B_1	(O) OR	(A) AND	Sum	Carry	Difference	Borrow

6. Conclusion

In conclusion, this paper has presented a novel approach to the design of a 1-bit Arithmetic Logic Unit (ALU) utilizing five 2-to-4 decoders and additional logic gates. The unique integration of 2-to-4 decoders in the ALU architecture introduces a dynamic input selection mechanism, enhancing the versatility of the unit in executing both arithmetic and logical operations. The designed ALU, with its modular structure, offers a promising solution for various digital applications, providing a balance between efficiency and flexibility.

The use of five 2-to-4 decoders allows for efficient control signal decoding, directing data bits to specific pathways within the ALU. This decoding mechanism, combined with additional logic gates, enables the unit to perform a diverse set of operations, including addition, subtraction, AND, OR. Simulation results have demonstrated the functionality and reliability of the ALU across a spectrum of input combinations and control signals, showcasing its effectiveness in digital computing environments.

The modular nature of the implemented design not only facilitates ease of integration into larger digital systems but also sets the stage for potential scalability to multi-bit ALUs. The adaptability of the design, driven by the unconventional use of 2-to-4 decoders, positions it as a valuable contribution to the field of digital circuit design, promising benefits in terms of simplicity, modularity, and scalability.

As with any novel approach, there remain opportunities for further exploration and optimization. Future work may delve into refining the design, exploring power-efficient implementations, and investigating the extension of the proposed approach to multi-bit ALUs. Additionally, real-world hardware implementation and rigorous testing will be essential to validate the design's performance beyond simulation environments.

In summary, the presented 1-bit ALU design showcases the potential of leveraging 2-to-4 decoders in unconventional ways, contributing to the ongoing evolution of digital circuit design. Through its innovative architecture, the designed ALU aligns itself with the pursuit of efficient and adaptable solutions, offering a foundation for future advancements in digital computing and integrated circuitry.

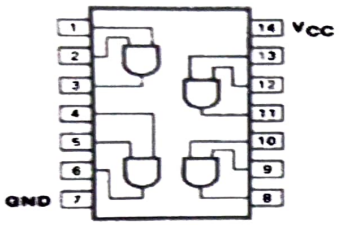
Appendices

Justification of the architecture / digital ICs used for implementation.

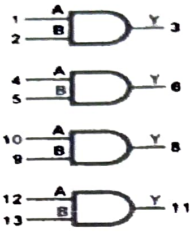
(Attach datasheets of the devices/ ICs that you have used)

7408 • 74S08 • 74LS08
QUAD 2-INPUT AND GATE

PIN ASSIGNMENT



LOGIC DIAGRAM



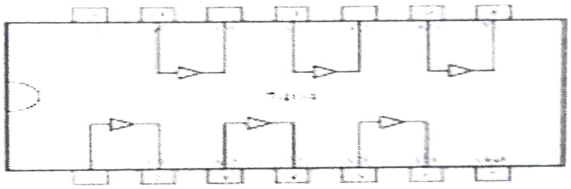
TRUTH TABLE

INPUTS		OUTPUT
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

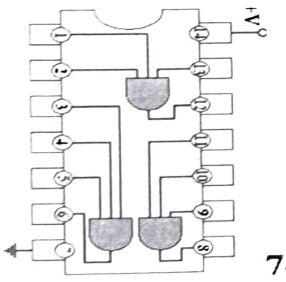
H = HIGH voltage level
L = LOW voltage level

NOT Gate

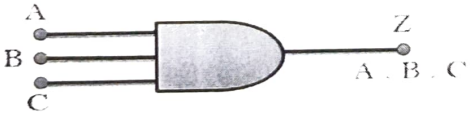
Input	Input	Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



Projectiot123.com



Triple 3-input AND GATE
TRUTH TABLE



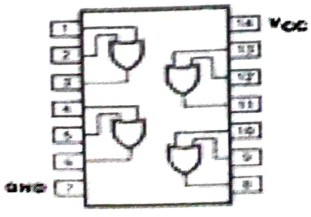
7411 TRIPLE 3-Input AND GATE

3 Input AND Gate Truth Table

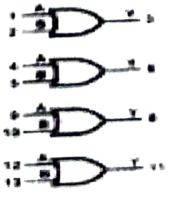
Inputs			Outputs
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

7432 • 74S32 • 74LS32 • 74F32
QUAD 2-INPUT OR GATE

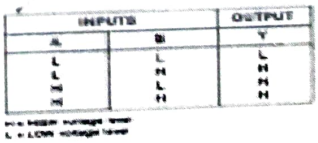
PIN ASSIGNMENT



LOGIC DIAGRAM

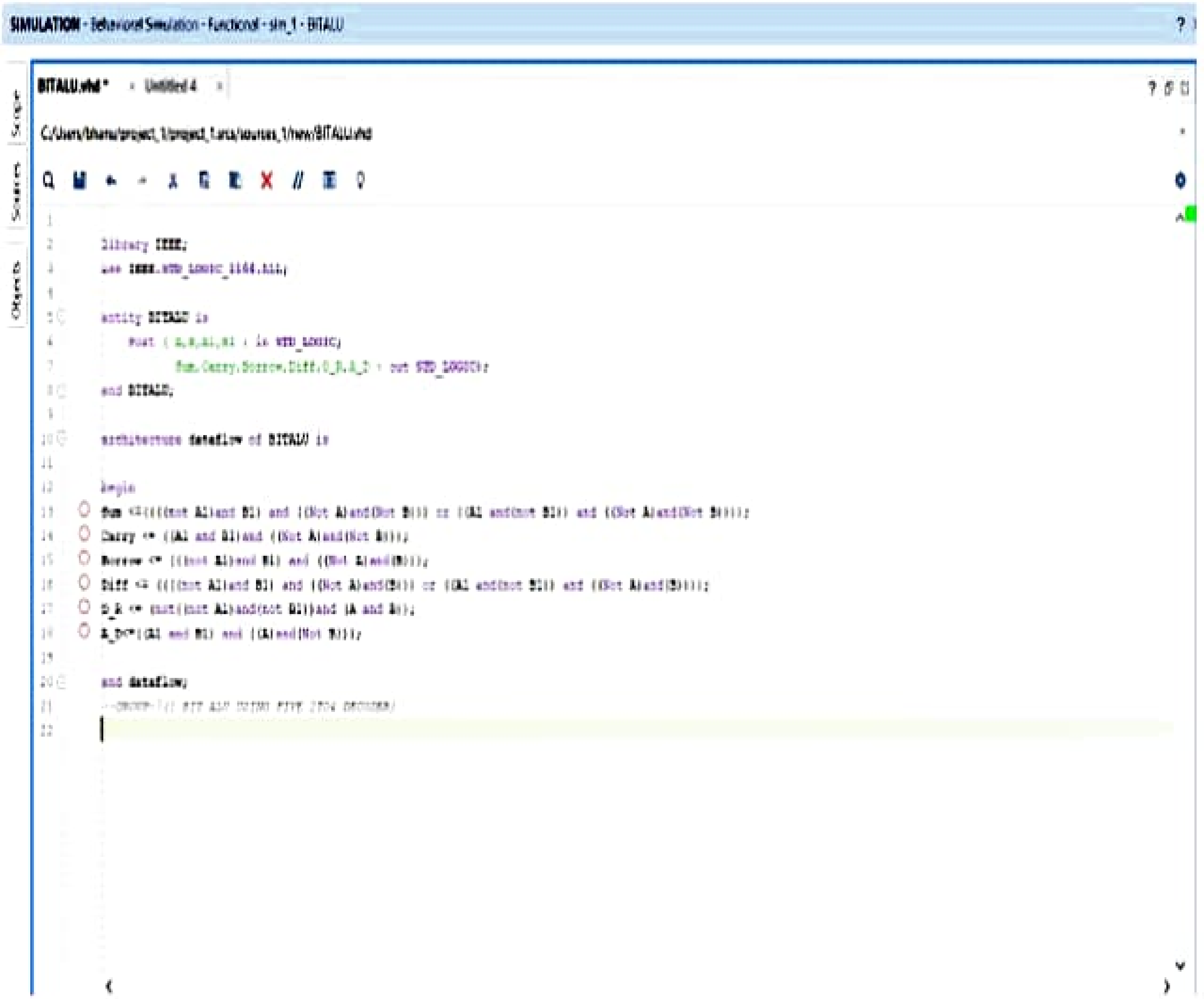


TRUTH TABLE

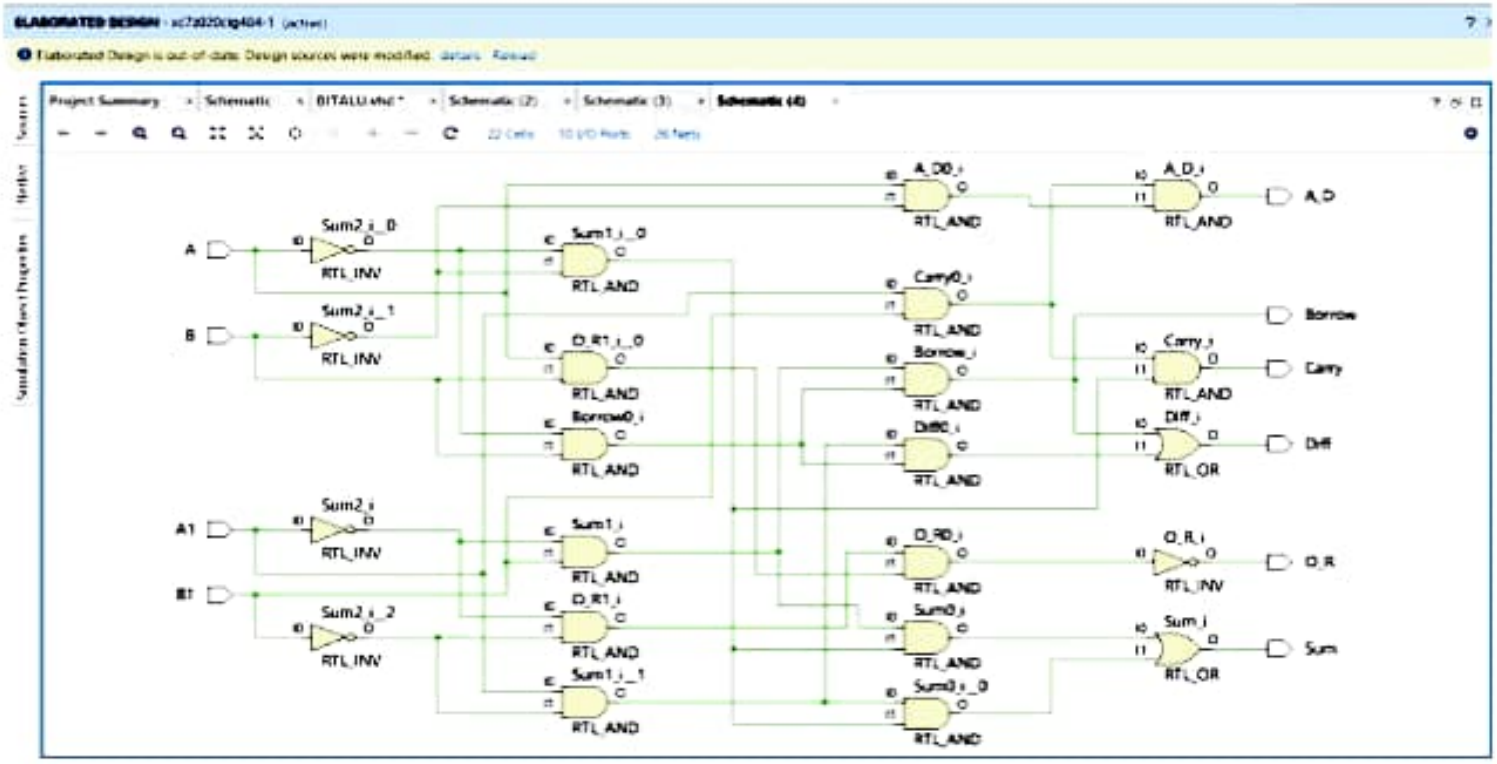


1-BIT ALU [VIVADO Implementation]:-

VHDL code:-



Schematic Diagram:



Behavioural Simulation :