**Digital Logic Design Project Report:**

**4 Bit Arithmetic Logic Unit (ALU)**

Submitted by:

Zeeshan Haider

DE-39-EE Syndicate: B



**NUST**

**COLLEGE OF ELECTRICAL AND**

**MECHANICAL ENGINEERING**

**ABSTRACT:**

The project was to design and implement an ALU (Arithmetic Logic Unit).

The ALU takes two 4-bit binary number as input and a 4-bit binary number as selection calls to implement 16 functions on the two input numbers. The project included the design and implementation of following 16 functions:

* Display A (4-bit Number)
* Display B (4-bit Number)
* Add A and B (A + B)
* Subtract B from A (A - B)
* Right circular shift of A by 1 bit
* Left circular shift of B by 1 bit
* Negate A (0 - A)
* Negate B (0 – B)
* Increment A (4-bit Number)
* Increment B (4-bit Number)
* Decrement A (4-bit Number)
* Decrement B (4-bit Number)
* A AND B (bit by bit)
* A OR B (bit by bit)
* A XOR B (bit by bit)
* Compare A and B (Select maximum)

**INTRODUCTION:**

The project consists of two major parts:

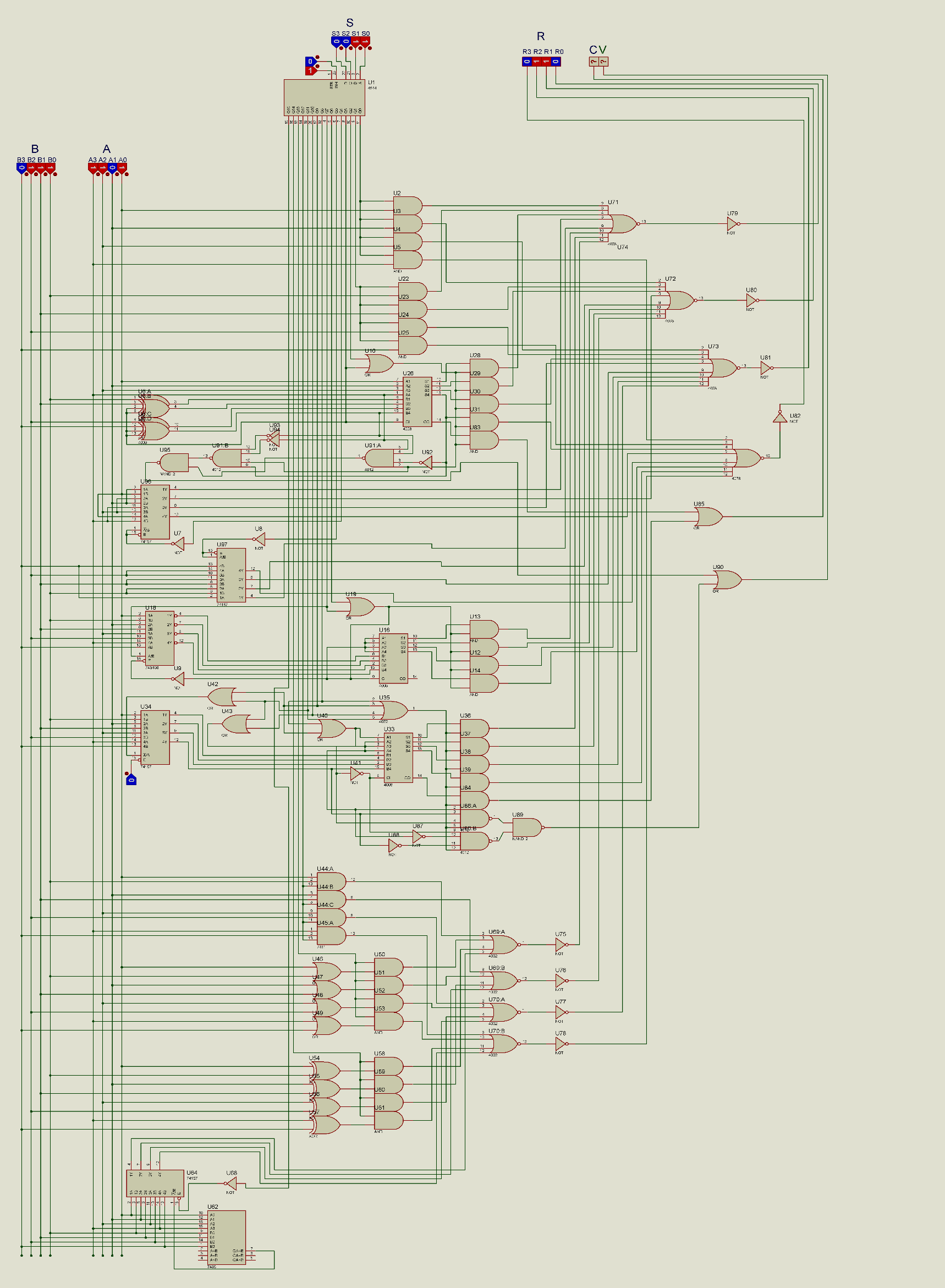
1. Design

2. Implementation

**DESIGN:**

The design of ALU was made using Proteus. The design is quite simple as we have used a 4x16 decoder which helps us give selection option in 16 different functions and simple AND, OR, XOR and other gates listed below to implement each function.

* Decoder (4514)
* OR (7432)
* 3 Input AND (7411)
* NOT (7404)
* XOR (7486)
* 4- Bit Binary Full Adder (7483)
* AND (7408)
* Quad 2x1 Multiplexer (74157)
* 8 Input NOR (4078)
* 4 Input NOR (4002)
* 2 Input NAND (7400)
* 4 Input NAND (7420)
* 4 Input OR (4072)
* Comparator (7485)
* Logic toggle
* Logic Probe

****

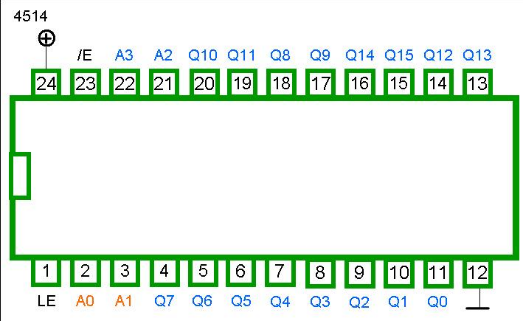
**IMPLEMENTATION:**

The design was implemented on breadboard. The implementation included sixteen Arithmetic functions in total. We made sure to keep all the function an sequential order and easy to understand pattern. Each function is designed in the efficient possible implementation provided with the least cost IC available in market. All of them were successfully implemented. The design is quite efficient, using only a small number of gates and less cost.

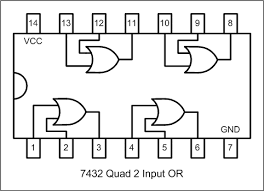
The breadboard design is similar to the proteus design. The only addition is the extra wires to provide IC with VCC and ground. The logic toggle in Proteus is implemented using a dip switch and logic probe is shown using a simple LED.

The gates and the ICs used in the project are as follows:

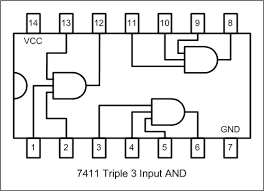
* Decoder (4514)



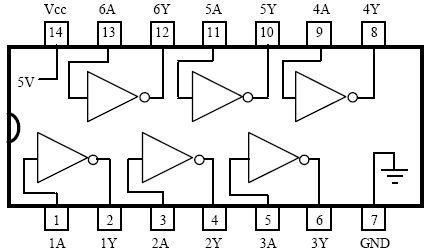
* OR (7432)



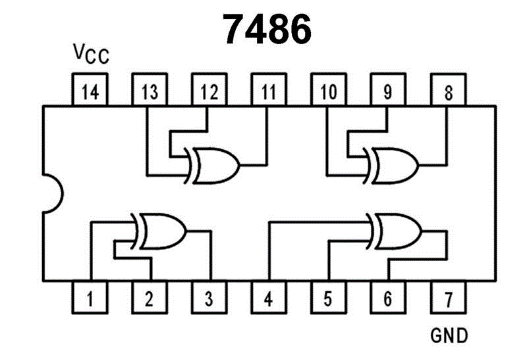
* 3 Input AND (7411)



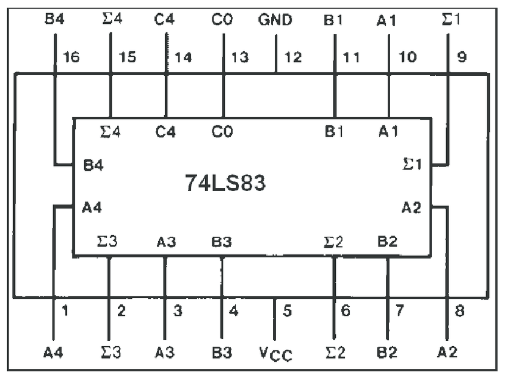
* NOT (7404)



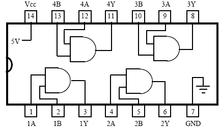
* XOR (7486)



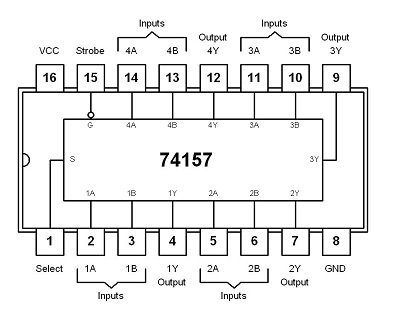
* 4- Bit Binary Full Adder (7483)



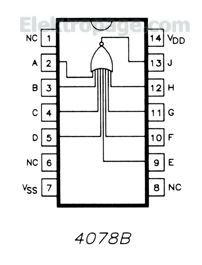
* AND (7408)



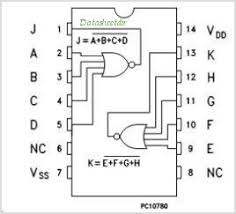
* Quad 2x1 Multiplexer (74157)



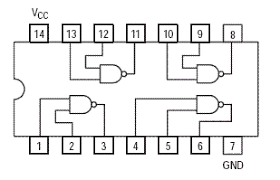
* 8 Input NOR (4078)



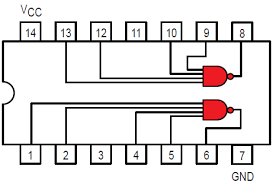
* 4 Input NOR (4002)



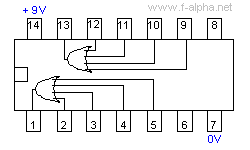
* 2 Input NAND (7400)



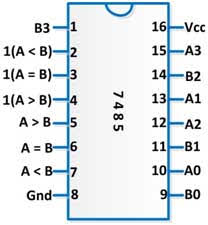
* 4 Input NAND (7420)



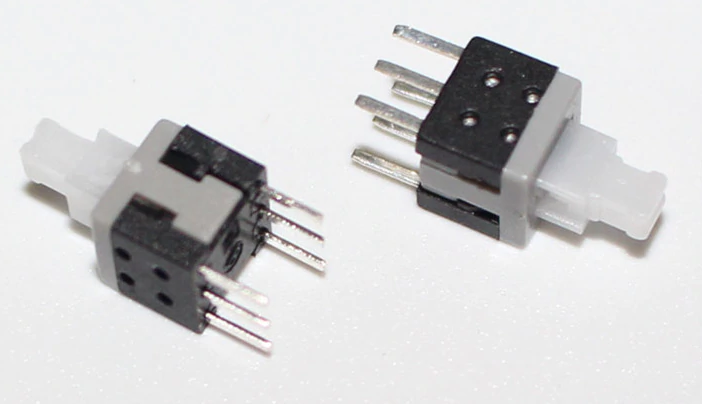
* 4 Input OR (4072)



* Comparator (7485)



* Push Button Switches



**Conclusion / Objectives Achieved:**

The circuit designed by us is giving us the desired results. The implemented circuit works as simulated in Proteus. Moreover, the designed circuit is quite efficient. The practical implementation is quite tough and time taking but the correct approach made it worth the effort. Using 4x16 decoder as selection mechanism and implementing short circuit for each function was a natural choice in this case. Ending each function’s output with an AND gate and connecting the AND gate to decoder output acted as an case sensitive switch and enabled us get the required output without disturbance from any other function.