**Digital Logic Design Lab Project Report**

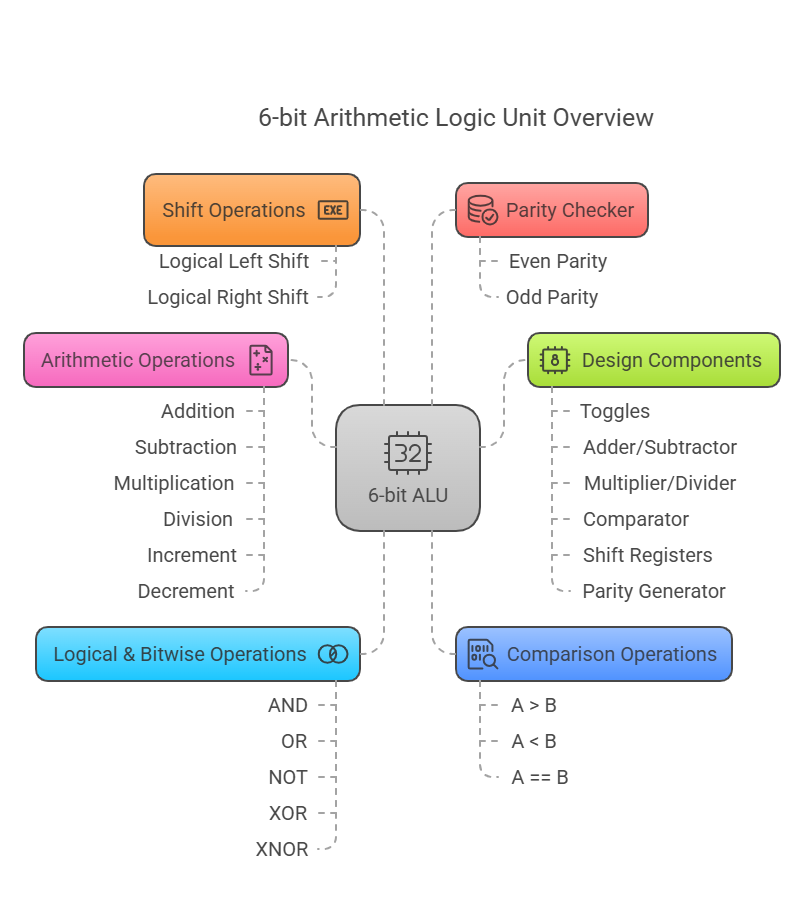


**Project Title : 6-Bit Arithmetic Logic Unit (ALU)**

**Group Members:**

1. Syed Muhammad Sufyan | 24K-0806
2. Faizan Basheer | 24K-0571
3. Abdur Rehman Khan | 24K-0767 **Dated:** 13-May-2025

**Course Instructor : Miss Fareeha Jabeen**

****

**1. Abstract**

This project presents the design and implementation of a 6-bit Arithmetic Logic

Unit (ALU) capable of performing 16 distinct arithmetic, logical, and shift operations. The ALU was built hierarchically, starting from basic logic gates to functional subcircuits, and finally integrated into a complete system controlled by 4-bit opcodes. The design demonstrates fundamental computer architecture principles and serves as an educational model for understanding CPU operations.

**2. Introduction**

**2.1 Main Objective :**

The primary goal was to **design and implement a functional 6-bit Arithmetic Logic Unit (ALU)** that performs:

1. **Arithmetic operations** (addition, subtraction, multiplication, division)
2. **Logical operations** (AND, OR, NOT, XOR)
3. **Shift operations** (left/right)
4. **Comparison & utility functions** (increment, decrement, parity check)

**Key Purpose:**

1. Demonstrate how **basic logic gates** combine to form complex computing units.
2. Simulate the **core computational component** of a CPU.
3. Provide hands-on experience in **modular digital circuit design** (from gates → subcircuits → full ALU).

**Learning Outcomes:**

✔ Understanding of **binary arithmetic & Boolean logic** in hardware  
✔ Experience in **hierarchical circuit design**  
✔ Practical knowledge of **ALU operation selection & control**

This project serves as a **foundation for computer architecture** and microprocessor design.

**2.2 Background:**

ALUs are fundamental components of CPUs, handling all arithmetic and logic computations. This project mimics commercial ALU designs at a simplified scale to illustrate:

* Binary number representation
* Combinational logic design
* Operation multiplexing

**2.3 Tools Used:**

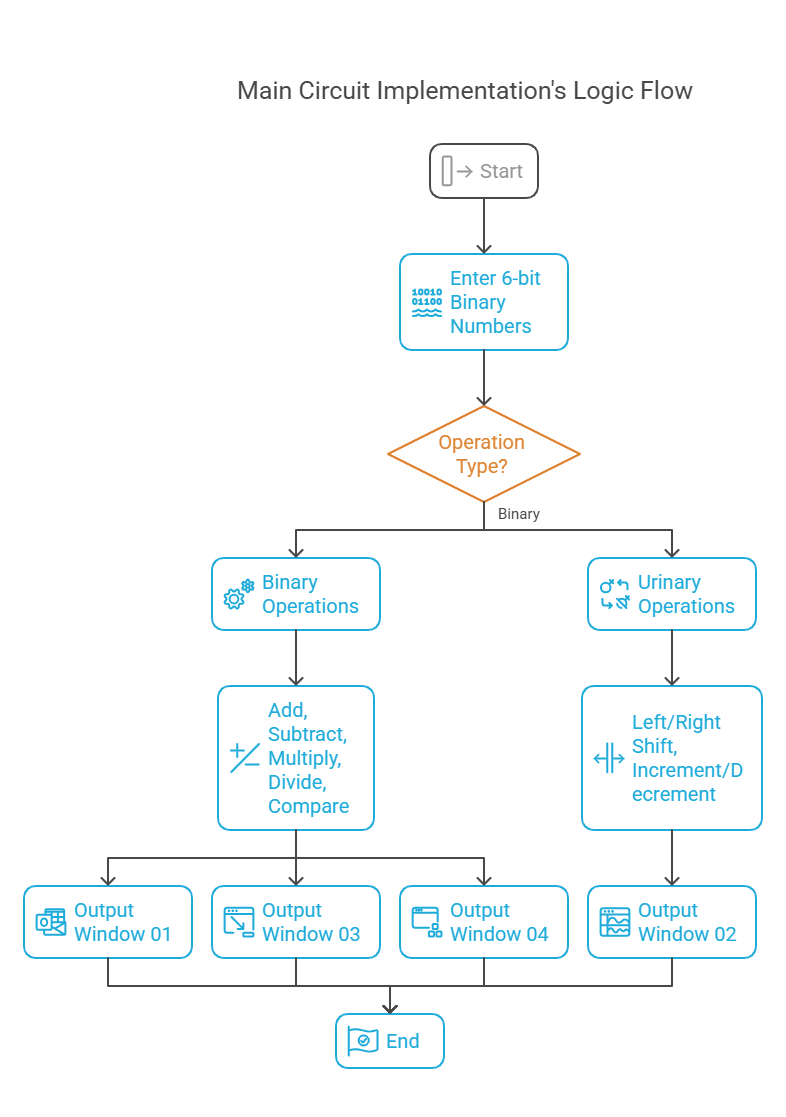
1. Software **Logisim-win** (Version 2.7.1)
2. Components:

All the Logisim built-in tools are used in development

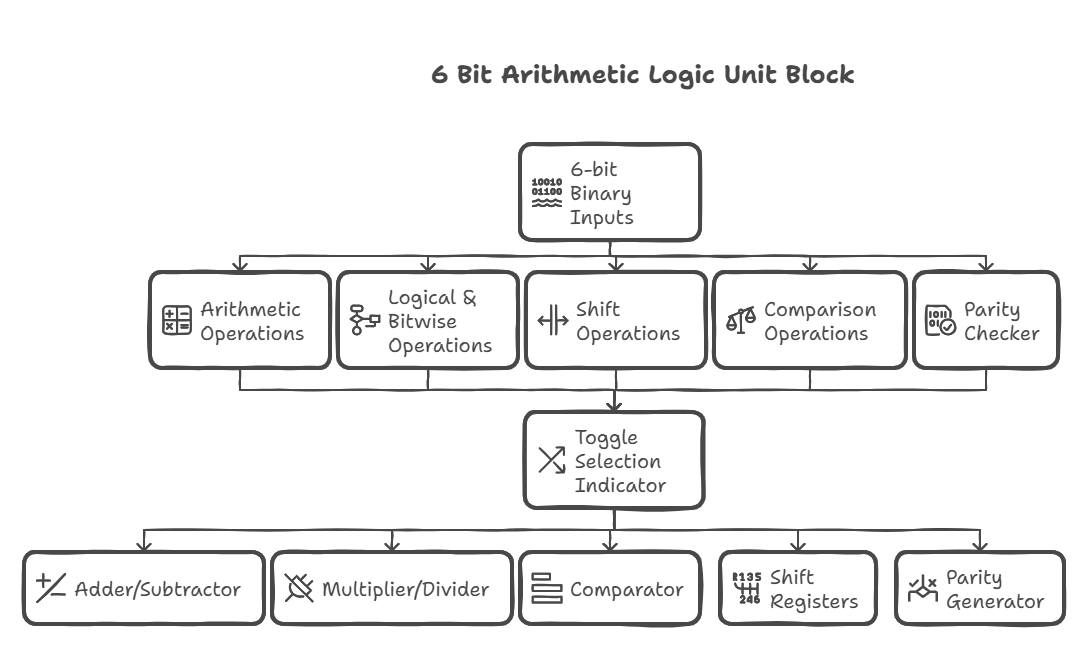
of this ALU, no external components are involved !

**3. Design Methodology**

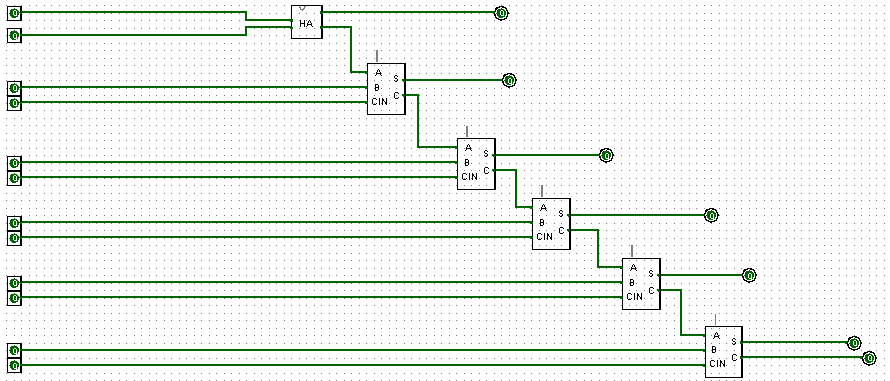
**3.1 Top-Down Approach :**

****

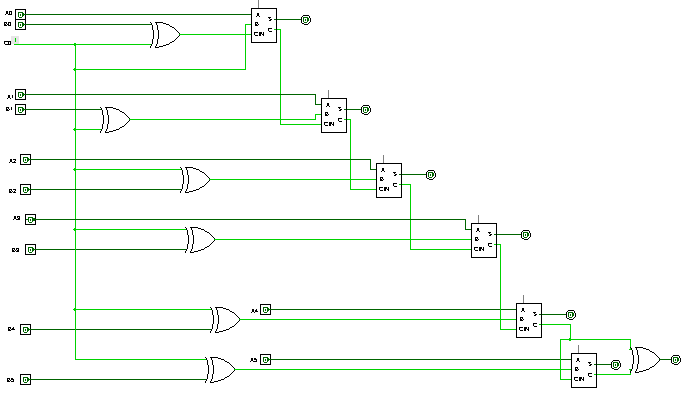
**3.2 Core Components:**

** *Block Diagram of ALU***

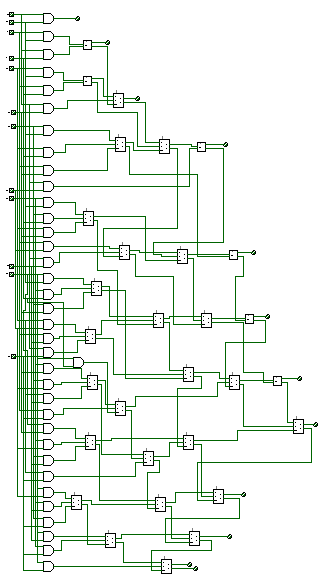
**A. Arithmetic Unit**

****

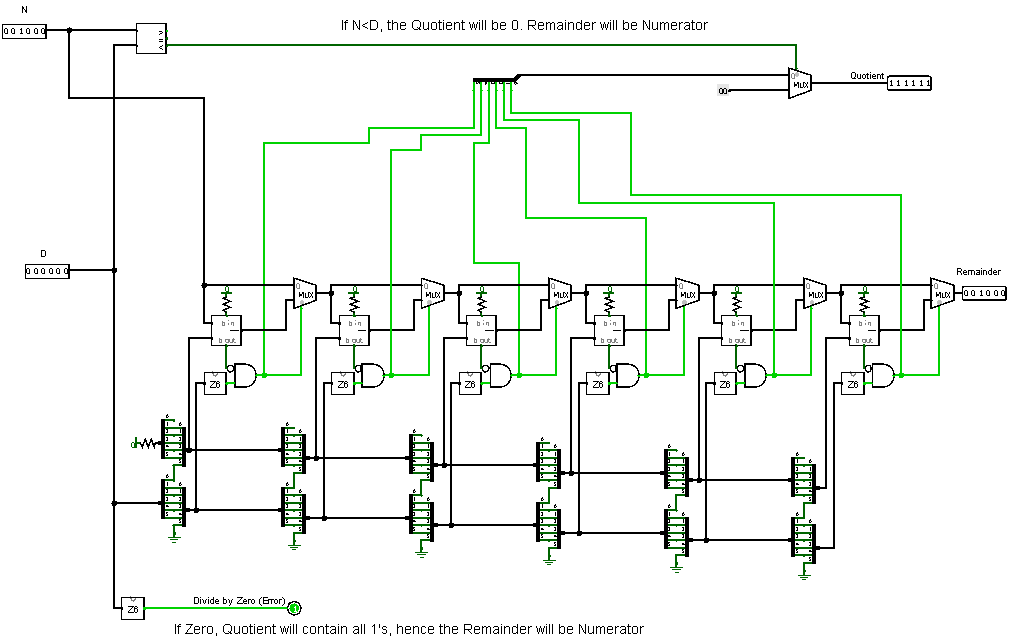
***6-Bit Adder using Full Adder as Subcircuit***

****

***6-Bit Subtractor using Full Adder as Sub-circuit***

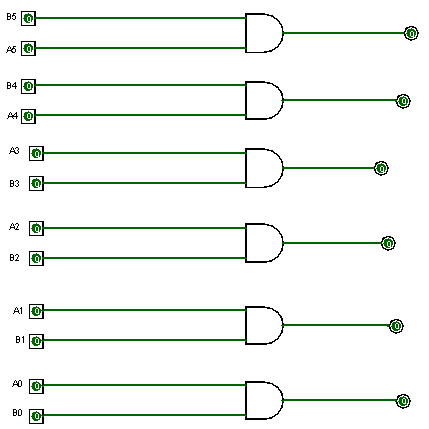
**

***6-Bit Multiplier using Full Adder as Subcircuit***

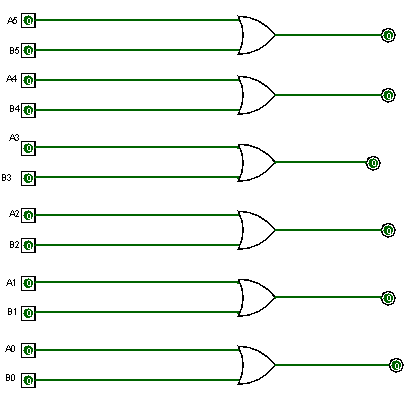
****

***6-Bit Divider using 6-Bit Subtractor as Subcircuit***

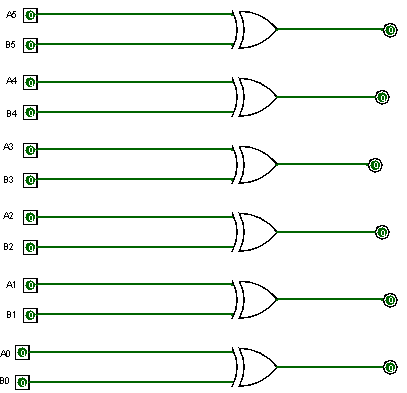
**B. Logic Unit :** Bitwise operations using gate networks



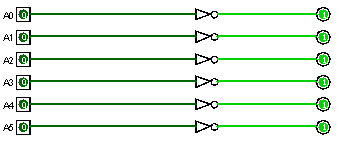
***Bitwise AND***

******

***Bitwise OR***

******

***Bitwise XOR***

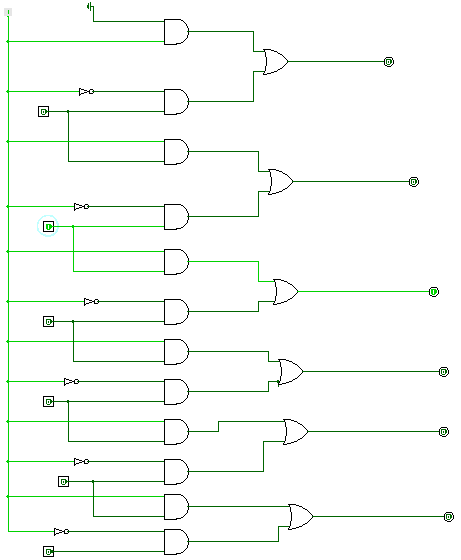


***Bitwise NOT***

**C. Shift Unit:**

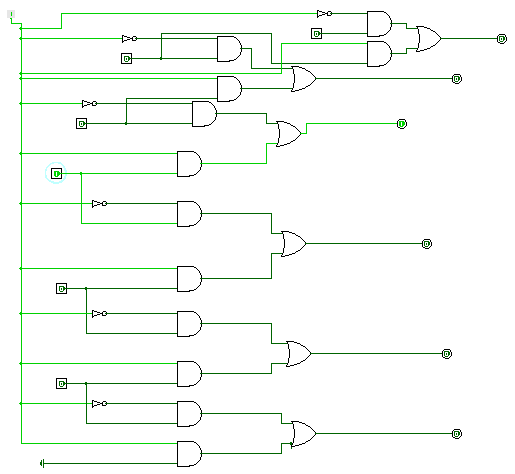
1. **Left Shift:**

Bits A[i-n] → RESULT[i] (for `i ≥ n`), else 0.



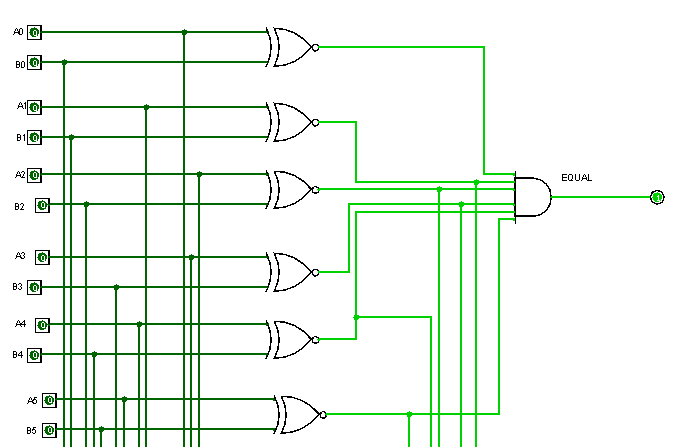
1. **Right Shift:**

Bits `A[i+n] → RESULT[i]` (for `i ≤ 5-n`), else 0.

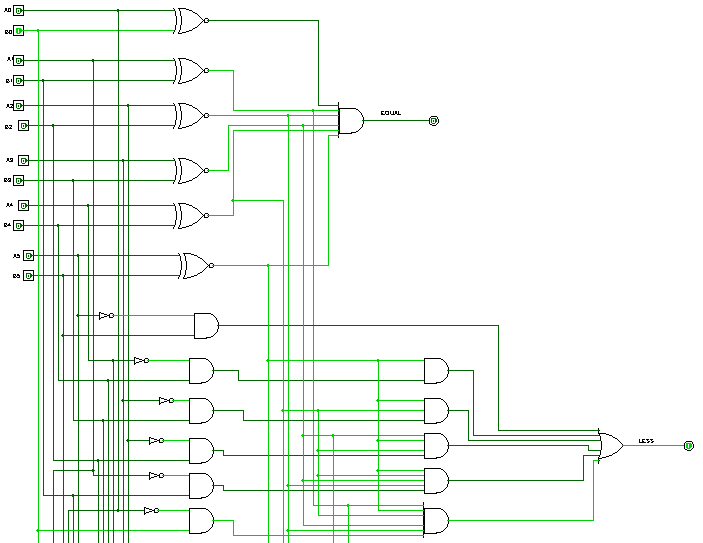


**D. Comparison Unit:** Compares two 6-bit inputs (A and B) and outputs:

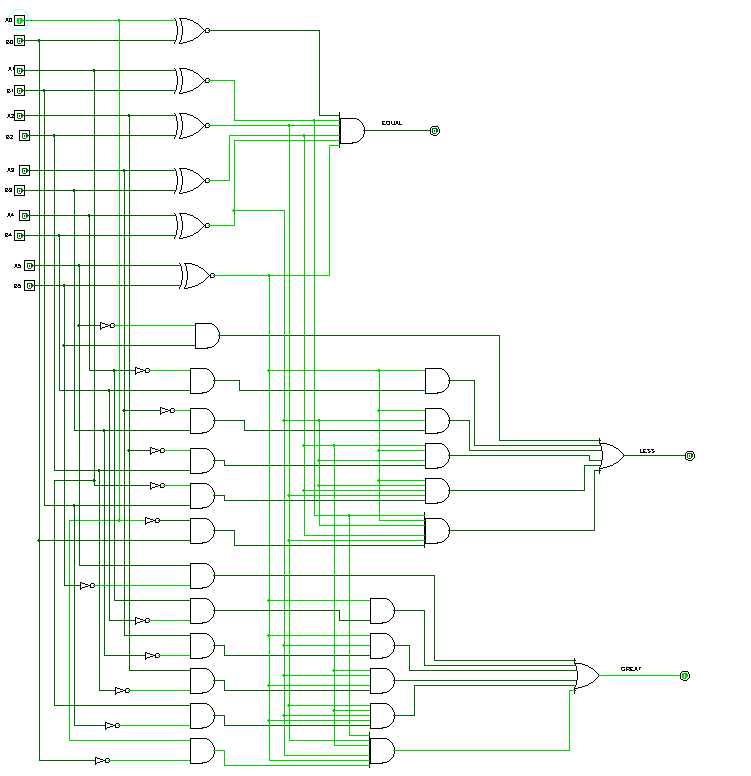
* A == B



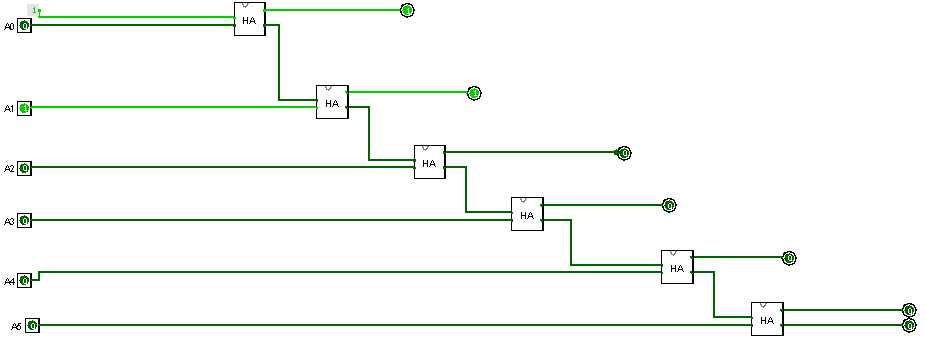
* A < B



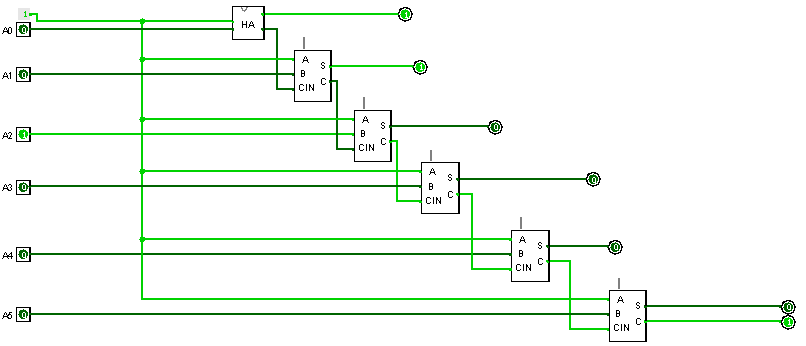
* A > B



**E. Increment/Decrement Unit:** Computes `A + 1`(increment) and `A - 1`(decrement)

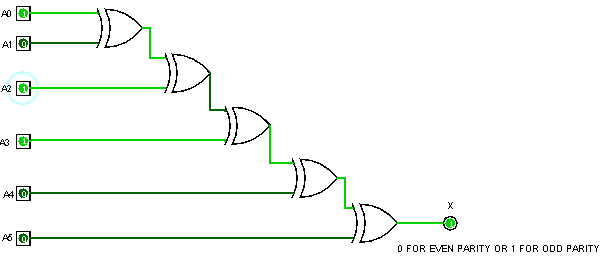
**1. Increment Circuit:** 

**2. Decrement Circuit:**

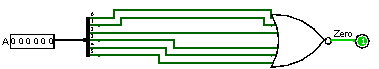


**F. Status Flag Unit:**

**1. Parity Check:** Determines if the number of `1`s in the input is even/odd.



**2. Zero Detect**: Checks if all the input bits are ‘0’.



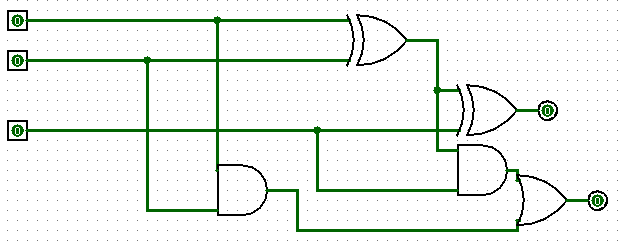
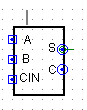
***Generously used in 6-Bit Divider Circuit***

**3.3 Gate-Level Building Blocks:**

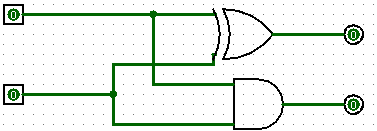
Designed Half Adders and Full Adders using Basic Logic Gates (AND,OR,NOT)

and used them in the complex circuit designs like 6-Bit Adder:

* Adder
* Subtractor
* Multiplier
* Divider

***Full Adder Circuit***

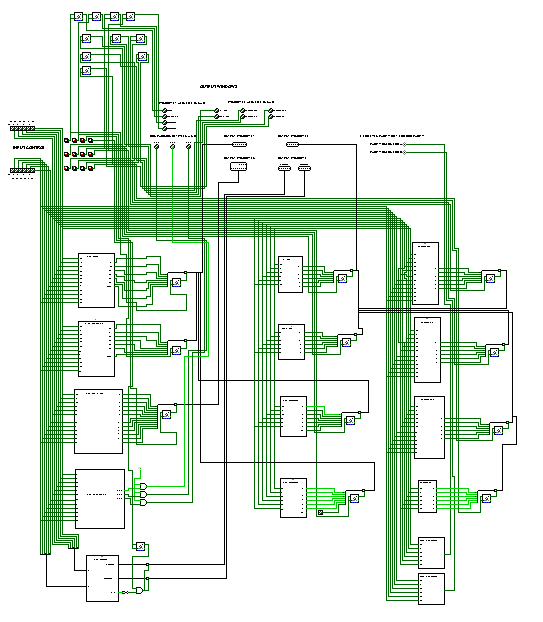
 

***Half Adder Circuit***

**3.4 Toggle Button Mechanism:**

Each operation in the ALU is controlled using a **toggle button**, a press sets it **ON**, and a second press turns it **OFF**. This simplifies user interaction and mimics instruction selection in real CPUs. To guide users, **output indicators** display which operation is active, ensuring clarity when multiple results are visible.

**3.4 All in One (Main Circuit) | The Satellite View**

****

**The main circuit integrates all submodules, operation decoders, data buses, selectors, and output visualizers—into a unified ALU simulation controlled via Toggle Buttons**

***"Behold the Beast: A 6-Bit ALU So Powerful, It Defies Screenshots!"***

**"This circuit is so dense, it has its own gravitational pull."**

**4. Conclusion: More Than an ALU**

What began as logic gates became a journey, a fusion of logic and learning, struggle and satisfaction. Our 6-bit ALU isn’t just circuitry; it’s a symbol of structured thinking, perseverance, and creative engineering.

**Technical Wins**  
We built a modular, robust system, 16 operations, 120+ gates, countless iterations. Each subcircuit told a story: carry chains, shift registers, comparators, all woven into a functioning digital brain.

**The Human Side**  
We grew in resilience. Simulations failed, yet we tried again. Designs clashed with logic, yet we adapted. Every connection was a decision. Every fix, a step forward.

**Looking Ahead**  
Beyond the bits and operations, we’ve built a foundation for future systems—and for ourselves as engineers. Thank you to everyone who shared this journey. May every bug bring growth, and every 0 and 1 hold meaning.

***With every bit flipped and every gate conquered, this ALU bows out. Sayonara, and onward to the next challenge.***