COMPUTER ARCHITECTURE ASSIGNMENT 8

Name: Yogita Mundankar
Rollno. B22CS068

32-Bit Processor Implementation

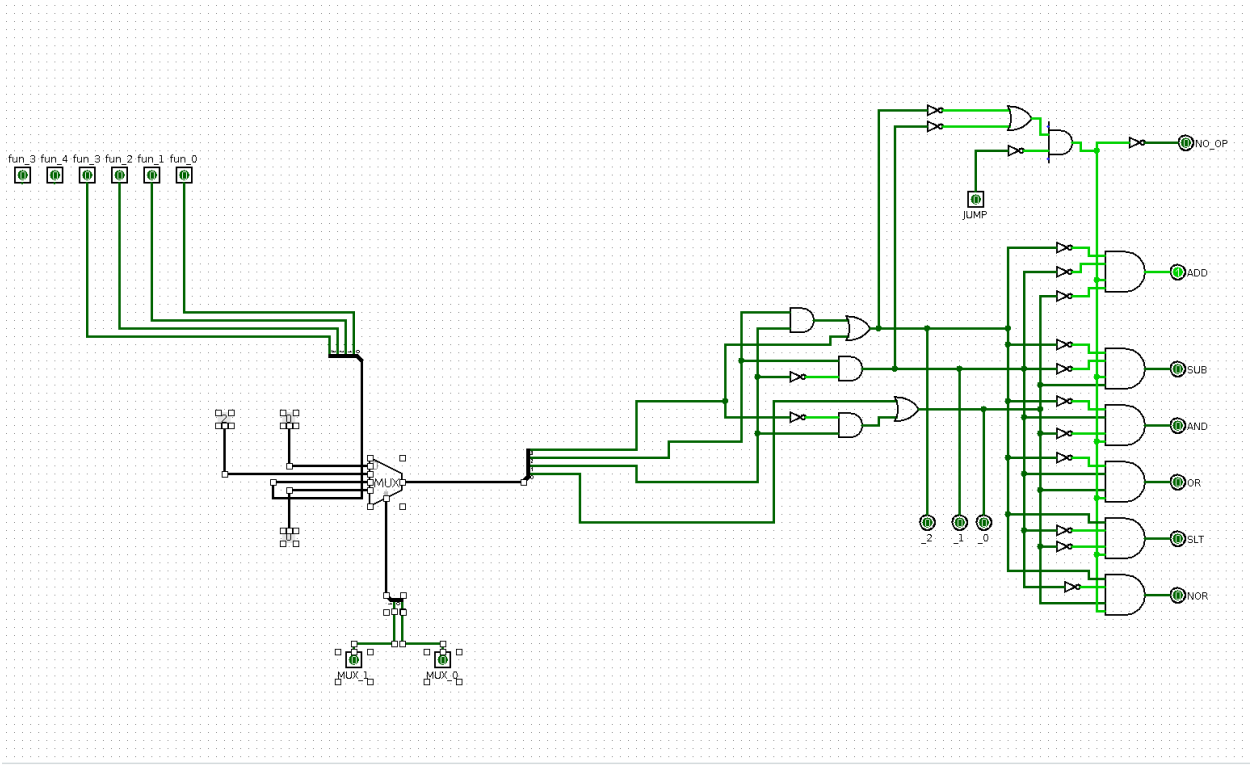
Objective

To design and implement a 32-bit processor capable of executing R-type instructions. The processor integrates:

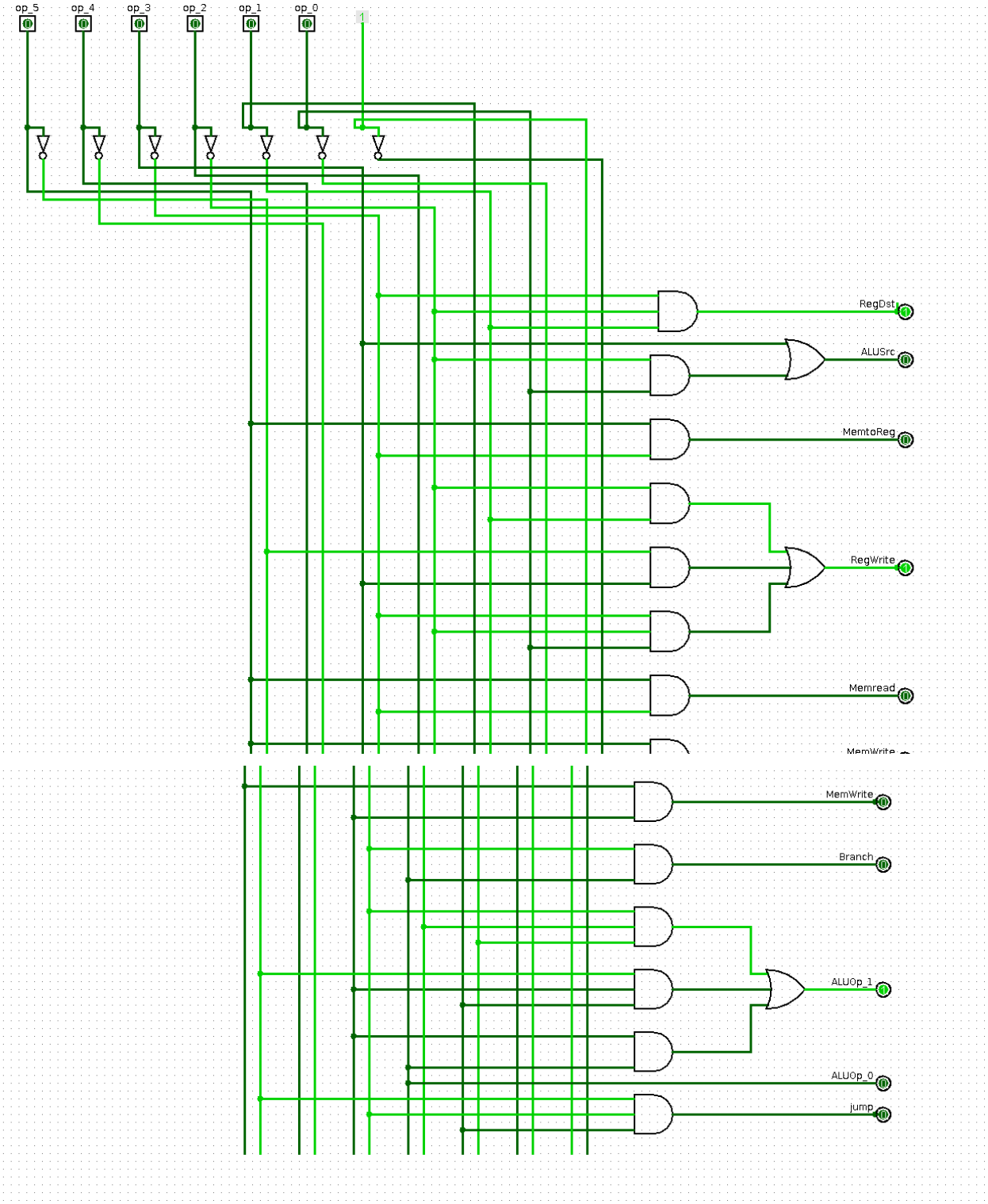
- A 32-bit ALU
- A 32-register file
- A control unit

This project aims to create a functional system in Logisim, ensuring modularity, efficient data flow, and accurate execution of R-type instructions with proper control signals.

ALU- Control Unit:



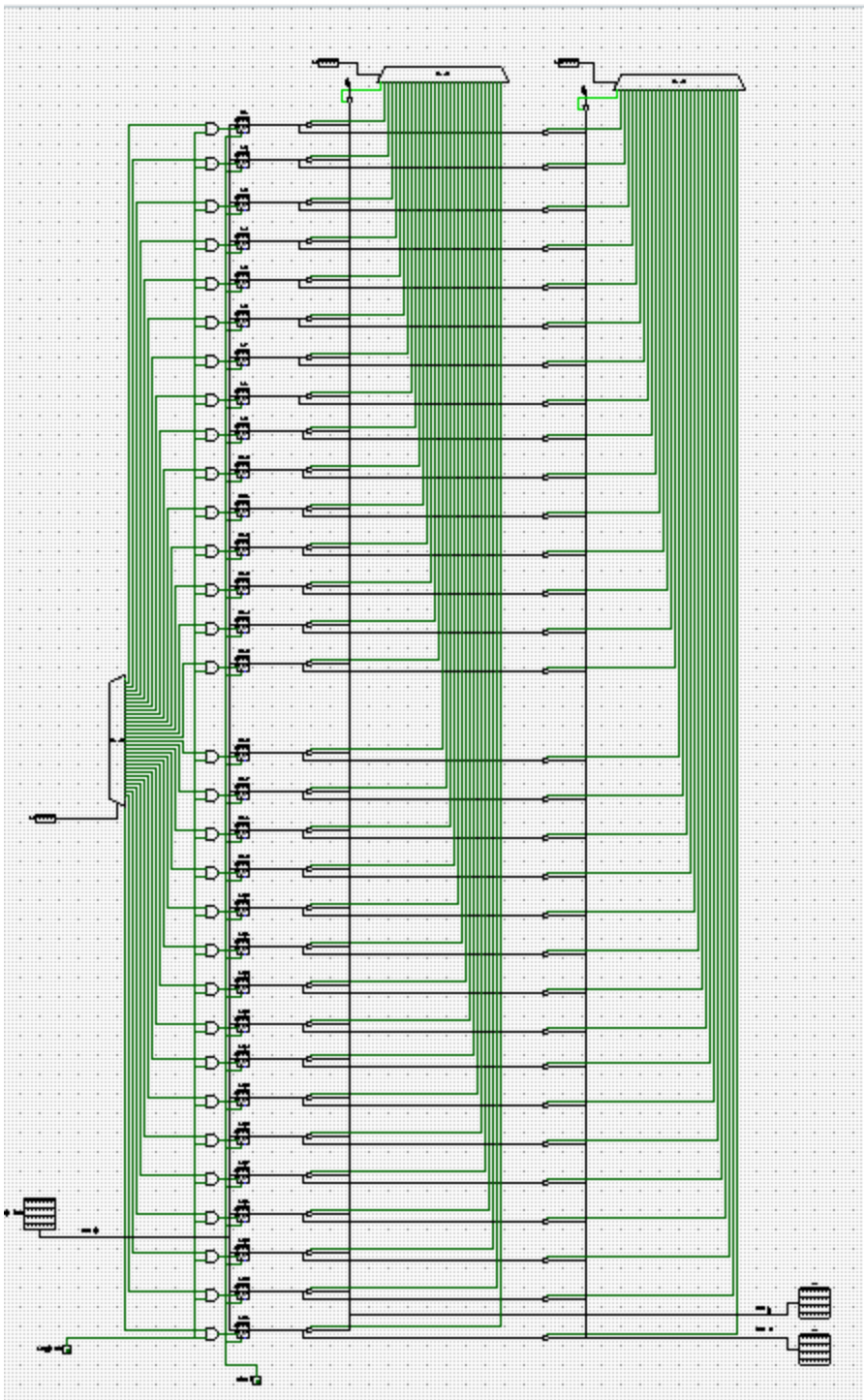
MIPS Control Unit:



Register Set

Key Features:

1. **Centralized Component**:
   Acts as the core for temporary data storage and facilitates data exchange between the ALU and the control unit.
2. **Buses**:
   - **Bus W**: Used to write data into the registers.
   - **Bus A & Bus B**: Used to read data for the ALU or other components.
3. **Structure**:
   - **Control Lines**: Manage read and write operations.
   - **Decoders**: Select specific registers based on the input control signals.
   - **Multiplexers & Buffers**: Ensure efficient data routing and reduce delays.
4. **Operations**:
   - **Write Operation**: Data from Bus W is written to a specific register, enabled by write signals.
   - **Read Operation**: Data from the registers is retrieved and sent through Bus A and Bus B using read enable signals.



ALU Design

Inputs/Outputs:

Inputs:
Two 32-bit data inputs (A and B).
Outputs:
A 32-bit result and an overflow flag.

Supported Operations:

Arithmetic:
Addition (ADD)
Subtraction (SUB)
Logical:
AND
OR
Shift:
Arithmetic Shift Right

Operation Selection:

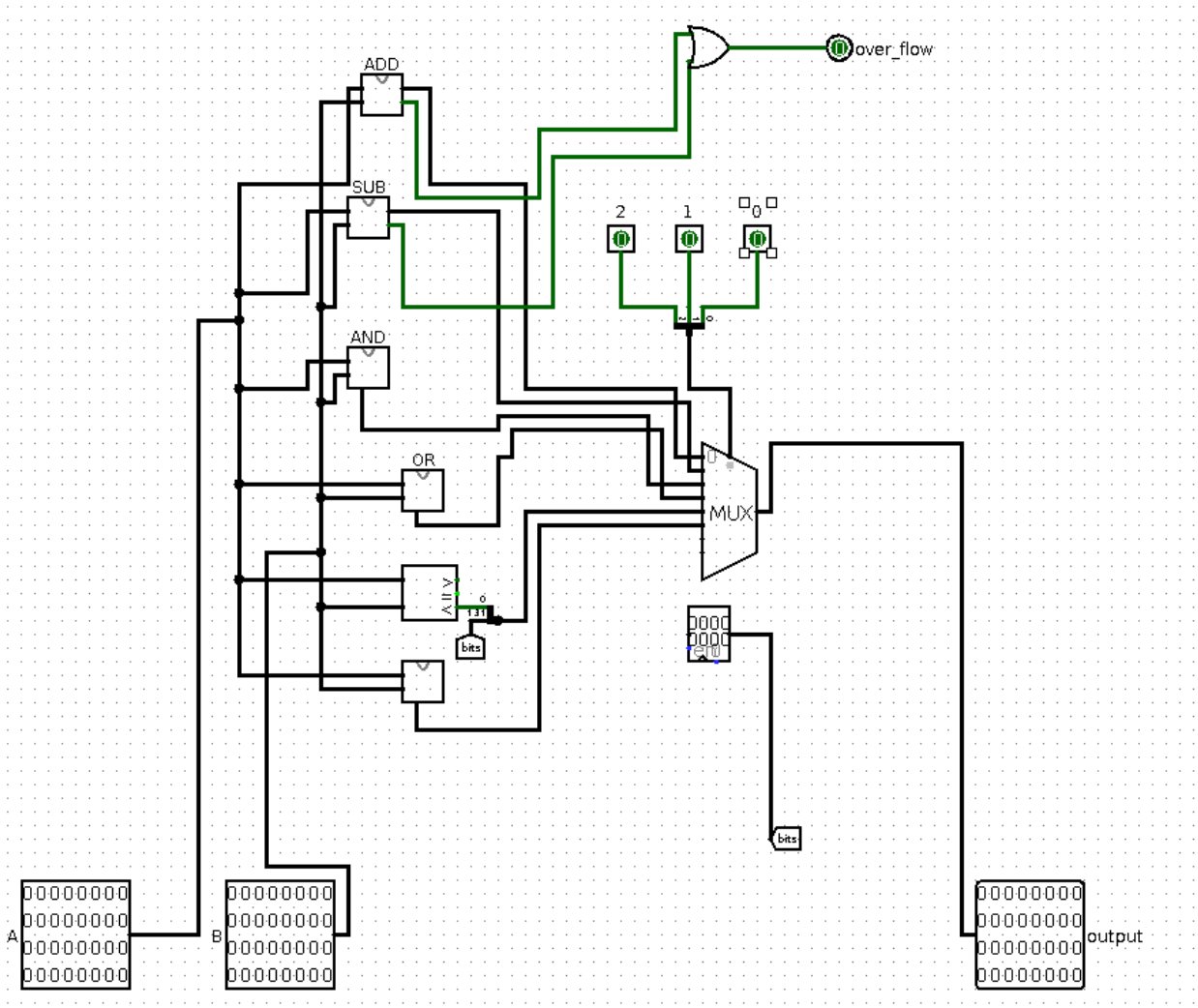
A Multiplexer (MUX) is used to select the result of the operation based on control signals.

Overflow Detection:

The ALU includes logic to detect overflow during arithmetic operations and sets an overflow flag accordingly.

Flow:

Inputs (A, B) are processed in parallel by the ALU.
The MUX selects the desired output based on control signals.
If overflow is detected, it is indicated through the overflow flag.



MIPS Integration

Register File (32 Registers)

- **Inputs**:
  - **5-bit Address Inputs**: `rs`, `rt`, `rd` (specify the registers).
  - **32-bit Write Data Input**: Data from the ALU result.
  - **RegWrite Control Signal**: Enables writing to the specified register.
- **Outputs**:
  - **Read Data 1**: Data from register specified by `rs`.
  - **Read Data 2**: Data from register specified by `rt`.
- **Clock Signal**: Ensures synchronized write operations.

32-bit ALU

- **Inputs**:

    - Two 32-bit data inputs retrieved from the register file (`Read Data 1` and `Read Data 2`).
- **Output**:
    - Performs the required computation and produces a 32-bit result, along with any necessary flags.