COAL

Report:

Stage 1 - Design of the Processor Architecture

1. PROCESSOR ARCHITECTURE

Block Diagram:

The block diagram illustrates the components of the calculator processor and their interactions. It includes:

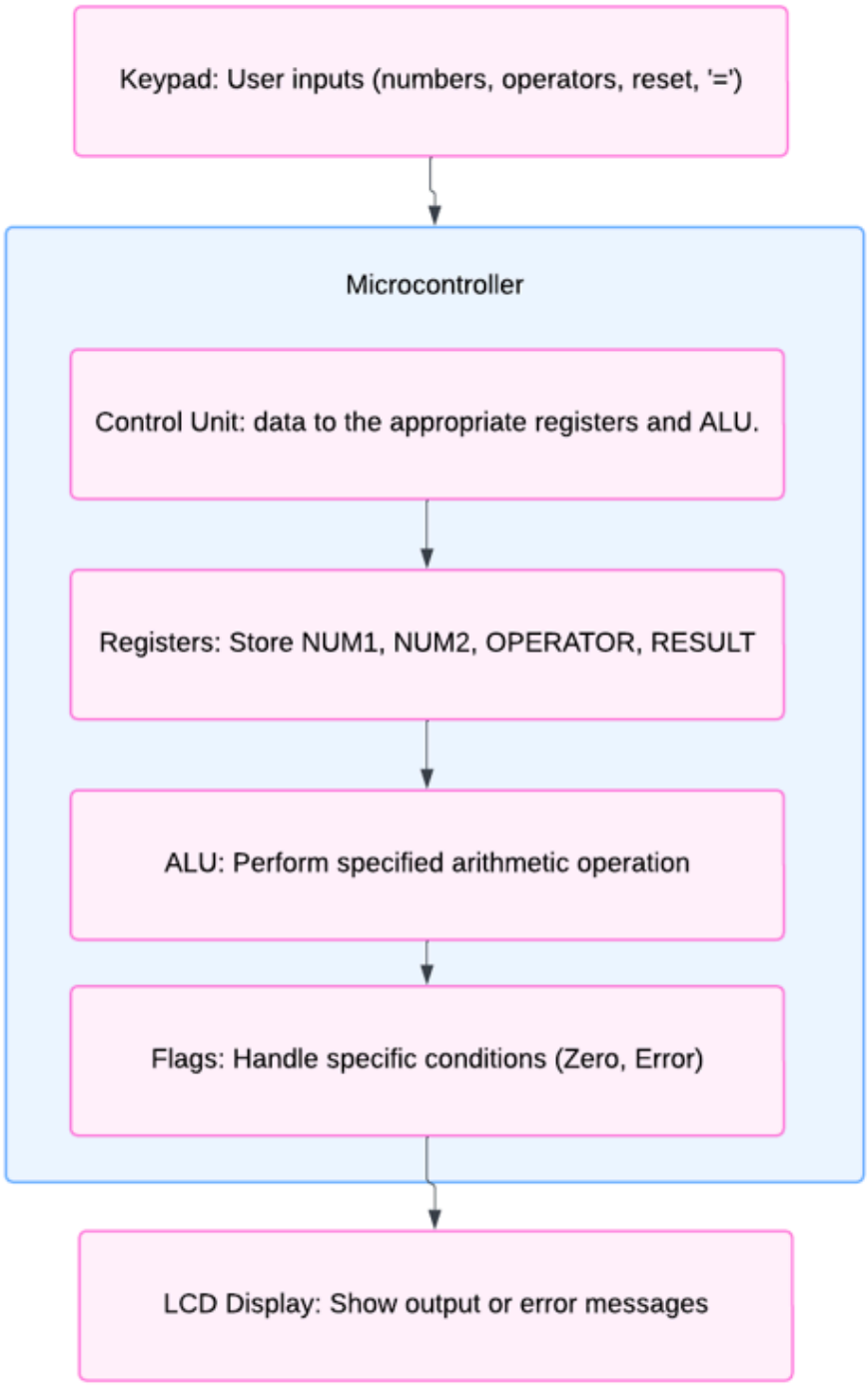
1. Arithmetic Logic Unit (ALU): Handles arithmetic and logical operations such as addition, subtraction, multiplication, division, AND, OR, XOR, and NOT.

2. Control Unit (CU): Decodes instructions, manages control signals, and ensures synchronization between components.

3. Registers: Temporary storage for operands (NUM1, NUM2), operator (OPERATOR), and the result (RESULT).

4. Memory: Stores program instructions and data for programmable functionality.

5. I/O Interface: Interfaces with input devices (keypad) and output devices (LCD display).


Keypad: User inputs (numbers, operators, reset, '=')
Microcontroller
Control Unit: data to the appropriate registers and ALU.
Registers: Store NUM1, NUM2, OPERATOR, RESULT
ALU: Perform specified arithmetic operation
Flags: Handle specific conditions (Zero, Error)
LCD Display: Show output or error messages

2. HARDWARE DESCRIPTION

LOGIC GATES AND COMPONENTS

Arithmetic Logic Unit (ALU):

- Constructed using basic logic gates (AND, OR, XOR, NOT) for logical operations.
- Arithmetic operations (addition, subtraction, multiplication, division) implemented using combinational circuits such as adders and subtractors.

- Control Unit (CU):

- Decodes instructions from the keypad and generates control signals for the ALU and registers.
- Ensures the correct sequence of operations and monitors flags (e.g., zero, carry, error).

Registers:

Stores intermediate values:

- NUM1: First operand entered by the user.
- NUM2: Second operand.
- OPERATOR: Holds the selected operation (+, -, *, /, etc.).
- RESULT: Stores the computed result before display.

Memory:

- Supports program storage for programmable designs or direct logic for a fixed-function calculator.

3. INTERFACE BETWEEN PROCESSOR AND I/O DEVICES

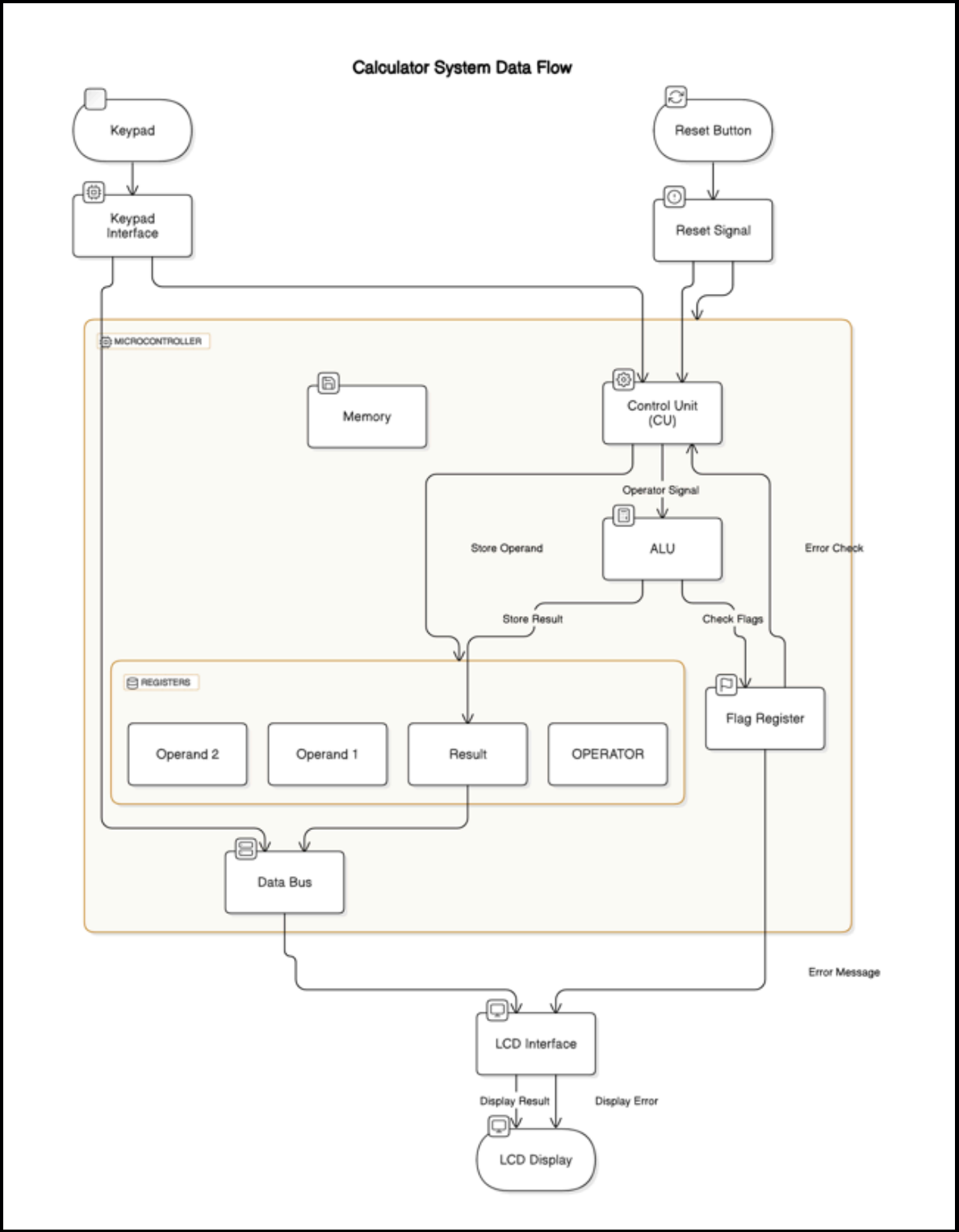
Input Devices:
Keypad:

- User inputs numbers, operations, and commands.
- Includes a reset button to clear data and restart the operation.

Output Devices:
LCD Display:

- Shows intermediate and final results.
- Displays error messages for invalid operations, such as division by zero or pressing = at the start.

DATA PATH DIAGRAM



SUPPORTED INSTRUCTIONS

The following instructions are supported by the processor:

1. **Load:** Load data from memory or input to registers.

2. **Store:** Store data from registers to memory.

3. **Add:** Perform addition of two operands.

4. **Sub:** Perform subtraction of two operands.

5. **Multiply:** Perform multiplication of two operands.

6. **Divide:** Perform division of two operands.

7. **AND:** Perform bitwise AND between two operands.

8. **OR:** Perform bitwise OR between two operands.

9. **XOR:** Perform bitwise XOR between two operands.

10. **NOT:** Perform bitwise NOT of a single operand.

11. **Exit:** Reset the system or terminate the program.

CONCLUSION

This design forms the basis for implementing a basic calculator with a microcontroller, using a minimal set of instructions and efficient hardware design. The block diagram, data path diagrams, and list of instructions ensure clarity in understanding the processor architecture and its interaction with external devices. The next stage will involve developing the assembly language code and simulating the functionality using Proteus.

Prepared by
Mahnoor Ikram (232115)
Abeera Mehtab (232087)
Eman Mansoor (232149)
Mahnoor (232083)