CPU ALU

Digital Logic Design Final Project - Semester 4031

Overview

The goal of this project is for students to design a basic Arithmetic Logic Unit (ALU) for a CPU using VHDL. The ALU will be capable of performing essential arithmetic operations (addition, subtraction) and logical operations (AND, OR, NOT), along with some additional functions like shift and rotate operations. This ALU will serve as a foundational component in understanding CPU architecture.

Functional Requirements

The ALU should be able to perform the following operations based on an opcode input:

- Basic arithmetic:
  - Addition: Add two input numbers and produce a result.
  - Subtraction: Subtract one input from the other and produce a result.
- Logical:
  - AND: Perform a bitwise AND operation on two inputs.
  - OR: Perform a bitwise OR operation on two inputs.
  - NOT: Perform a bitwise NOT operation on one input.
- Shift:
  - Logical left shift: Shift the input bits to the left.
  - Logical right shift: Shift the input bits to the right.
  - Arithmetic right shift: Shift the input bits to the right, retaining the sign bit for signed operations.
- Rotate:
  - Rotate left: Rotate the bits of the input to the left.
  - Rotate right: Rotate the bits of the input to the right.
- Other Operations:
  - Comparison: Output 1 if input A is greater than input B; otherwise, output 0.
  - Equality check: Output 1 if input A is equal to input B; otherwise, output 0.

Specifications

Inputs

- Operand A: The first input operand (e.g., 8-bit).
- Operand B: The second input operand (e.g., 8-bit).
- Opcode: A control signal that selects the operation to perform.
- Clock (optional): If you want to integrate sequential elements like pipelining or registers.
- Reset: To initialize or clear the ALU state.

Outputs

- Result: The output result after the selected operation is performed.
- Zero flag: A flag that is set when the result is zero.
- Carry/Borrow flag: A flag indicating a carry (for addition) or a borrow (for subtraction).
- Overflow flag: A flag indicating an overflow error for signed arithmetic operations.
- Comparison flags (optional): Flags for greater-than or equal-to comparisons.

Implementation Details

Operation selection

Implement a control mechanism based on the opcode to select one of the operations. The ALU should decode the opcode and execute the corresponding operation.

Combinational logic

Implement logic for each operation separately (e.g., addition, AND, shifts) as combinational logic.

Flags and status indicators

Set flags based on the results of arithmetic and comparison operations:

- The Zero Flag should be set if the result is zero.
- The Carry/Borrow Flag should indicate a carry for addition and a borrow for subtraction.
- The Overflow Flag should indicate an overflow for signed addition/subtraction.
- Comparison Flags can be set based on the outputs of the comparison and equality checks.

Modules

- Arithmetic unit: Handles addition, subtraction, and any overflow/carry/borrow flags.
- Logical unit: Performs AND, OR, NOT operations.
- Shift and rotate unit: Performs shifts and rotations as specified.
- Control unit: Decodes the opcode and routes signals to the correct module based on the operation selected.

Extra features

Sequential pipelining: If desired, implement simple pipelining or a clock signal to test basic sequential ALU behavior.

Other operations and signals You're free to add additional input and output signals or operations to your design to extend or fine-tune its functionality.

Deliverables

- VHDL code: Source code for the ALU modules, including the control logic for operation selection.
- Testbenches: VHDL testbenches that verify each ALU operation individually and in combination.
- Simulation results: Simulation waveforms or logs demonstrating the ALU's functionality, including flag outputs and correct handling of operations.
- Documentation containing the following:
  - Design Document: Detailed description of each module and the ALU control logic.
  - Testing Report: Summary of the tests conducted and their results.
  - User Guide: Brief guide explaining how to use the ALU, interpret its output, and understand its flags.

Evaluation Criteria

- Design completeness (30%): Adherence to all project requirements and functionalities.
- VHDL code quality (20%): Code organization, readability, modularity, and adherence to good coding practices.
- Testing and validation (20%): Quality of testbenches and evidence of successful testing.
- Documentation (20%): Clarity and completeness of design and testing documentation.
- Efficiency and innovation (10%): Creative and optimized solutions, and any additional interesting features (e.g., pipelining, sequential elements).

Submission Guidelines

Until Sat 1st Feb 2025, 23:59

E-mail: mkoushan@macaw.me

SUBJECT: DLD4031_FP

BODY: *STUDENT_NO*