Logic Design
CMP101
Fall 2024

Cairo University
Faculty of Engineering
Computer Engineering Department

Semester Project Description Document

Design and Implementation of an Arithmetic Unit on DE1-SoC FPGA

1. Introduction

This project aims to design and implement an arithmetic unit capable of evaluating mathematical expressions involving five single-digit signed numbers and four operations. The design emphasizes practical digital logic concepts and the integration of Verilog HDL with FPGA hardware.

The project uses switches for input, seven-segment displays for intermediate and final results, and LED indicators for sign, zero, and division-by-zero flags. It also incorporates an optional bonus feature to evaluate the expression with operator precedence after an extra clock cycle.

2. Objectives

- To design an arithmetic unit in Verilog that can:
    1. Accept single-digit signed numbers and operations sequentially via switches.
    2. Display the operands sequentially on a seven-segment display.
    3. Display intermediate results after every operation.
    4. Output the final result of the expression after five inputs on seven-segment displays.
- To implement and test the design on the DE1-SoC FPGA board.
- To include division-by-zero detection and handle errors gracefully.
- **Bonus:** Evaluate the expression with operator precedence after an extra clock cycle.

3. Design Requirements

Functional Requirements

1. Input Mechanism:
    - Signed numbers (0–3) and operations (+, -, *, /) are input sequentially using switches.
    - Each clock cycle, we input a single number and an operator, except for the last cycle in which we input a number only. (as we have 5 numbers and only 4 operators)
    - With each clock cycle the inputted number is displayed on one of the seven-segment displays.
2. Intermediate Results:
    - Intermediate results are displayed on the seven-segment displays (dedicated to displaying the final results and intermediate results).
3. Final Output:
    - After the last clock cycle, the cumulative result is displayed on the seven-segment displays dedicated to the result. (same as those of intermediate results)
4. Error Handling:
    - Division by zero results in 0 being displayed (on the seven-segment displays), and a dedicated div-by-zero LED is lit.
    - Cumulative (final) results with values equal to zero results in displaying 0 on the seven-segments as well, and a dedicated Zero LED is lit.
    - Cumulative (final) results with negative values results in a dedicated Sign LED being lit.
5. Bonus Feature:
    - After the last clock cycle, the expression is reevaluated with precedence rules (+ and - after *, /) and displayed.

Hardware Requirements

- DE1-SoC FPGA Development Board:
  - Switches: For number and operation input.
  - Seven-Segment Displays: To display numbers and results.
  - LEDs: To indicate errors (e.g., division by zero).
- Clock Input:
  - One push button will simulate the clock. Number and operator dedicated switches need to be adjusted before pressing the clock button to capture the values on clock edge.

Constraints

- Operations are evaluated accumulatively (left to right, no precedence) for the first result.
- A maximum of five inputs numbers and four operators is allowed.
- Built-in operators should **NOT** be used, build your own functions using logic gates.
- In division, we are only interested in the quotient. Ignore the reminder.

4. Implementation Details

The design will be implemented in Verilog using a modular approach. Each functionality (input capture, arithmetic operations, display control) will be implemented as separate modules and interconnected.

Mapping to FPGA Hardware

FPGA Component	Usage
Switches (SW[1:0])	Input number
Switches (SW[2]	Sign bit (+ve is 1)
Switches (SW[5:4])	Operator (you design the 4 combinations)
Leftmost 7-Segment Displays	Show intermediate and final results.
Rightmost 7-Segment Display	Show the current number.
LEDs (LEDR[0])	Indicate division by zero flag.
LEDs (LEDR[1])	Indicate sign flag.
LEDs (LEDR[2])	Indicate zero flag.
Push Button (KEY[0])	Simulated clock input.

5. Simulation and Testing

For Simulation: Simulate the design in a Verilog simulation tool (e.g., Quartus or ModelSim) to validate:

1. Correct input handling and sequencing.
2. Accurate intermediate and final results.
3. Error and flags handling.
4. Bonus feature for precedence evaluation.

For Hardware Testing: Load the design onto the DE1-SoC board and test:

1. Input handling via switches.
2. Correct display of results on seven-segment displays.
3. LED indication for flags.

6. Bonus Feature

After the last clock cycle, evaluate the expression with operator precedence (multiplication and division first, then addition and subtraction). Update the display to show this result on the seven-segment displays.

7. Deliverables

1. Verilog source code files.
2. Simulation results (waveforms and outputs).

3. Demonstration on the DE1-SoC board.
4. Final project report including:
    - Design overview.
    - Module descriptions.
    - Test cases and results.
    - Challenges and solutions.

8. Team Formation

Each team consists of 3 – 4 students.