Cairo University, Faculty of Engineering, Computer Department
VLSI Design and Automation, CMP305, Fall 2024
Deadline: Week 13

Course Project

Overview

This project involves designing, testing, and synthesizing various circuits, focusing on understanding and analyzing their characteristics. It emphasizes mastery of RTL designs, testing, and the synthesis process. Note that students will be asked about the design concepts during discussions to assess understanding.

1 Basic RTLs

This section outlines the requirements and specifications for implementing the signed 32-bit designs and their testbenches.

1.1 Adders

Design the following 32-bit signed integer adders:

- Ripple Carry Adder
- Carry Look-Ahead Adder
- Carry Bypass Adder
- Carry Select Adder

1.2 Multipliers

Design the following 32-bit signed integer multipliers:

- Verilog ('*') version
- Multiplier Tree (combinational)
- Sequential Multiplier (shift & accumulate)
- Booth Algorithm

1.3 Testing

Adder Testbench Specifications

Implement a testbench to test the adders, covering the following cases:

- Overflow of positive numbers.
- Overflow of negative numbers.
- Addition of a positive and a negative number.
- Addition of two positive numbers.
- Addition of two negative numbers.
- Three additional random test cases.

Multiplier Testbench Specifications

Implement a testbench to test the multipliers, covering the following cases:

- Multiplication of a positive and a negative number.
- Multiplication of two positive numbers.
- Multiplication of two negative numbers.
- Multiplication of a negative and a positive number.
- Multiplication by zero.
- Multiplication by one.
- Two additional random test cases.

Common Testbench Requirements

The testbench for both adders and multipliers should:

- Print “TestCase#1: success” on success.
- Print “TestCase#1: failed with input X and Y and Output Z and overflow status N” (with blue text replaced by actual values) on failure.
- Report the total number of success and failure test cases at the end.

1.4 Synthesize using OpenLane

Synthesize the adders and multipliers using **OpenLane**, applying the following constraints:

- Virtual clock: 20ns.
- Input delay: 1ns.
- Load: 10.
- Output delay: 0.5ns.
- Utilization: 60%.
- Enable the usage of all library cells.

Generate a report including:

- Total Area
- Utilization
- Maximum Delay

1.5 Placement and Routing

Place and route the basic RTL designs using constraints similar to those used in synthesis. Follow these steps:

- Set the clock skew constraint to 0.2ns.
- Use only vertical strips for routing.

Generate a report including:

- Total Area
- Utilization
- Maximum Delay

- If any design suffers from negative slack, adjust the design to fit the constraints.
- Ensure that all designs operate correctly under the same modified constraints.
- Perform post-synthesis simulations using the testbenches to verify the synthesis process.
- Apply post-routing simulation using your previously made testbench. (include your sdf file)

2 Main RTL

This section focuses on designing a comprehensive ALU, combining the knowledge and results obtained from the synthesis and testing of the basic RTL components in Section 1.

2.1 ALU Design

Design an Arithmetic Logic Unit (ALU) with the following specifications:

- Includes input and output registers.
- Contains an operation selector to choose between addition and multiplication operations.
- Uses the most appropriate adder and multiplier (accroding to your findings in Section 1) to implement a 32-bit floating-point adder and multiplier (IEEE-standard).
- The reasoning behind your choise should be clearly stated during the discussion.

2.2 Testing and Simulation

Combine the testbenches outlined in Section 1.3 into a unified testbench to verify the ALU functionality. Ensure the testbench covers:

- Correct operation of the addition and multiplication paths.
- Thorough testing of edge cases for both integer and floating-point arithmetic.
- Reporting of results in the same format described in Section 1.3.

2.3 Synthesis and Placement

- Synthesize the ALU using **OpenLane**, applying the same constraints outlined in Section 1.4.
  - Clock: 20ns.
  - Input delay: 1ns.
  - Load: 10.
  - Output delay: 0.5ns.
  - Utilization: 60%.
  - Enable the usage of all library cells.
- Perform placement and routing for the ALU, using the constraints described in Section 1.5:
  - Set the clock skew constraint to 0.2ns.
  - Use only vertical strips for routing.
- Generate the final GDS file for your design.

2.4 Reporting

Generate reports during synthesis and placement that include:

- Total Area.
- Utilization.
- Maximum Delay.
- Clock.
- Total Power.

2.5 Post-Placement Simulation

- Perform post-synthesis and post-placement simulations using the unified testbench.
- Include the `sdf` file in post-routing simulations to validate timing.
- Ensure the correctness of all results and address any negative slack issues as outlined in Section 1.

3 Delivery

The final project deliverables include all files and reports generated during the synthesis, routing, and static timing analysis (STA) of both the **Basic RTLs** and **Main RTL**. Ensure the following items are prepared and submitted:

3.1 Deliverables

- **Source Files:** Include all source files written for the **Basic RTLs** and **Main RTL** during:
  - Verilog Files
  - Configuration Files
  - Helper Scripts
- **Generated Files:** Include all files generated for the **Basic RTLs** and **Main RTL** during:
  - Synthesis
  - Placement and Routing
  - Static Timing Analysis (STA)
- **Excel File:** Prepare an Excel file with two sheets:
  1. **Sheet 1: Basic Modules Comparison**
     Compare the synthesized basic modules (adders and multipliers) based on:
     - Area
     - Delay
     - Power
     - Any other parameters you deem relevant.
  2. **Sheet 2: Main RTL Reports**
     Include the reports for the Main RTL with details such as:
     - Area
     - Delay
     - Any additional parameters you find relevant.
- **Demonstration:** Be prepared to show your Main RTL module in action using **QuestaSim**.
- **Project Report:** Include team's information and showcase your work and justifications to help you through the discussion.

3.2 Delivery Date

The project deliverables are due during **Week 13**, starting on **21st December 2024**. Ensure all materials are complete and ready for submission by this deadline.

Best of Luck!