Floating-Point Unit Design using IEEE Standard 754 Project Proposal

Tyler Wilson   Michael Schena
Electrical and Computer Engineering Department
Utah State University

***Abstract* – Implementation of the IEEE Standard 754 for Floating Point Unit execution in a Verilog model using Synopsis. The implementation of this standard is verified by using a C++ golden model program.**

I. INTRODUCTION

The IEEE Standard 754 [1] describes Floating-Point Arithmetic. In this standard, more specifically it specifies the following:

1. Basic and extended floating-point number formats.
2. Add, subtract, multiply, divide, square root, remainder, and compare operations.
3. Conversions between integer and floating-point formats.
4. Conversions between different floating-point formats.
5. Conversions between basic format floating-point numbers and decimal strings.
6. Floating-point exceptions and their handling, including nonnumbers (NaNs).

II. OBJECTIVE

The objective of the project is develop a Floating-Point Unit (FPU) Design that meets the previously mentioned standard. This design will be implemented in Verilog using the Synopsis tool and verified that the results are correct by creating a C++ program.

III. BASIC METHODOLOGY

The following main components of the project are as follows:

1. Overall architectural design of the FPU.
2. Verilog/Synopsis design of the exponential representation of a FPU.
3. Verilog/Synopsis design of the mantissa to represent the decimal portion of the floating-point number.
4. Verilog/Synopsis design of the sign bit to represent the sign (positive or negative) of the floating-point number.
5. Placement, routing, timing, and simulation of the designed system.
6. C++ program to verify the results obtained from the simulation results.

These tasks will be divided amongst the two team members given that the different parts prove to be module.

<u>Tyler's Tasks</u>

- Develop the mantissa representation of a floating-point number in Verilog using the Synopsis tool.
- Develop a C++ program the verify the results of Verilog FPU design.

<u>Michael's Tasks</u>

- Design an overall architecture of the FPU.
- Develop the exponential representation of a floating-point number in Verilog using the Synopsis tool.

<u>Combined Tasks</u>

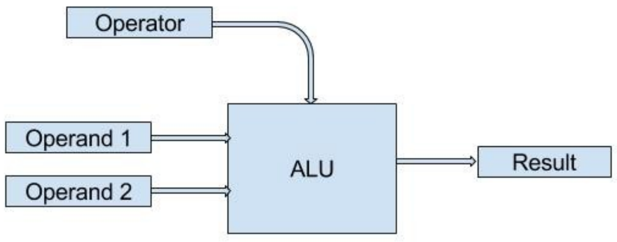
- Using Synopsis, run placement and routing of the architectural design.
- Verify the architectural design works and gather results.

IV. PROGRESS

<u>Golden Model</u>

The Golden Model has been developed in C++ to verify that the output from the Verilog simulation is correct. The Golden Model accurately does all of the necessary computations (+,-,*,/,√) and comparisons (>,<,=). The output of the C++ program displays the resulting float value after the operation occurs. In addition to this, the binary representation of the float value is displayed to the screen. The output from the Verilog simulation can then be compared to the results from the Golden model to verify accuracy.

Block Level Diagram



Description of the Project

A more complex design in Verilog is in the process of being completed in order to run simulations using the Cadence tools. Once the Verilog code is completed for each individual operator, then we can proceed to synthesize, perform placement, and lastly, routing to verify for correctness in operation.

V. REFERENCES

[1] IEEE Standard for Binary Floating-Point Arithmetic. New York: IEEE, 1985. Print.