Lab 2 - Converting Fixed-Point to Floating-Point and Vice Versa

Prelab

You will need to write your own testbench and submit it on ilearn prior to coming to the lab (check online for due dates). Your testbench should demonstrate that you have read through the lab specifications and understand the goal of this lab. You will need to consider the boundary cases. You do not need to begin designing yet, but this testbench will be helpful during the lab while you are designing.

Introduction

This lab consists of 2 parts. In the first part, you will develop a Verilog simulation to convert a **fixed-point** binary number to a single-precision **floating-point** number. In the second part, you will extend your program to support the opposite i.e. from floating point to fixed point.

- A floating-point tutorial can be found here.
- Your code *MUST* handle 0 input and negative inputs.

Part 1 - Fixed-Point to Floating-Point

You will receive two inputs representing the fixed point value. The first value will be the number of bits to the right of the binary point. The second value will be a 32-bit fixed-point number (read as a 2's complement integer). You can assume the values given will not be out of range. You will create a module using the template provided (`fixedToFloat.v`) in the zip file and implement the fixed to floating-point operation. Your Verilog simulation should output the floating-point number.

Fixed-Point as an Integer:

Assume we want to represent 8.25 as an integer in 32.7 format (7 bits to the right of the binary point).

- 8.25 in binary is 1000 . 01
- Extend to 7 bits to the right of the binary point
  1000 . 010 0000
- Extending to 32 total bits, maintaining the binary point:
  0000 0000 0000 0000 0000 0100 0 . 010 0000
  - To hex: 0xh0000 0420
  - To an integer: 1056 (Ignoring the radix point)

Fixed-Point to Single-Precision Floating-Point Conversion

Input : 7 , 1056 (i.e. 0xh07, 0xh00000420)
Output : 8.25 as single-precision FFP (i.e 0_10000010_0000_1000_0000_0000_0000_000 or 0x41040000)

Part 2 - Floating-Point to Fixed-Point

You will receive two inputs, the first value will be the desired number of bits to the right of the binary point on the output. The second value will be a 32-bit floating-point number. You can assume the values given will not be out of range. You will create a module using the template proved (`floatToFixed.v`) in the zip file and implement the floating-point to fixed-point operation. Your Verilog simulation should output the fixed-point number.

Sample Input:

7 (i.e 0xh07)
8.25 as single precision FFP
(i.e 0_10000010_0000_1000_0000_0000_0000_000 = 0xh41040000)

Sample Output:

1056 (i.e 0xh00000420)

Mandatory functionality

- User inputs will be specified in the testbench
- Floating-point outputs must be formatted as floating-point values
- Fixed point outputs can be formatted as integer values

Part 3

You will now create the top-level module for this project using the template provided (`fixedFloatConversion.v`). This top-level module should support both fixed to float and float to fixed-point conversion. Which module you use is configured by the `opcode` input (1 for `floatToFixed` and 0 for `fixedToFloat`).

Submission:

Each student **must** turn in one zip file to Gradescope. The contents of which should be:

- A README file with your name and email address, and any incomplete or incorrect functionality
- All Verilog file(s) used in this lab (implementation and test benches).

If your file does not synthesize or simulate properly, you will receive a 0 on the lab.