Lab 09

Part C – Two-High-Bit Challenge x16

Description

In this lab module, you will work with a more generalized solution to the two-high-bit challenge from Labs 05B, 06C, and 07A.

Procedure

Project Creation

1. Download `full_comparator.vhd`, `half_comparator.vhd`, `comparator_6.vhd`, `half_adder.vhd`, and `lab09c.vhd` from Canvas and place them in a new folder titled `lab09c`.
2. Open Vivado and create a new project titled appropriately.
   - Note that there is no constraint file for this lab module, so you can skip that step in the project creation process.
   - For your convenience, the board identifier is `xc7a35tcpg236-1`.
     - You can also find the board with these options: `General Purpose`, `Artix-7`, `cpg236`, `-1`.

Hardware Review and Simulation

3. Open `lab09c.vhd` in the editor, and follow along with the instructor to review the hardware description of it and the other files.
4. `Open Elaborated Design`, review the schematic alongside the instructor, and take screenshots for your report.
5. Setup the behavioral simulation.
   - In Vivado, from the left side pane, click `Run Simulation`, then `Run Behavioral Simulation`.
   - Minimize the two panes on the left with `Scope` and `Objects` tabs to make the wave view bigger.
   - Optionally, minimize the bottom pane (`Tcl Console/Messages/Log`).
   - Optionally, drag the handle next to `Value` to make the columns larger.

6. Create a clock on the `clk` signal:
    - Right-click the signal `clk`.
    - Select the `Force Clock...` option.
    - Set the `Leading edge value` to `1` and the `Trailing edge value` to `0`.
    - Keep the `Duty cycle (%)` at 50%.
    - Set the `Period` to `10ns` instead of `100ns`.
7. Now, load the data by forcing `load` to `1`:
    - Right-click the signal `load`.
    - Select `Force Constant...`.
    - Set a `Force value` of `1`.
8. Force `shift` to `0` similar to how you did for `load`.
9. Force `input1` to a 32-bit number with at least 16 high bits.
    - For example, you can use `11100001111110001111001010001000` in binary, which is `E1F8F288` in hexadecimal.
        - If you choose to enter a binary value, you will need to change the `Value radix` to `Binary` in the `Force Constant` menu.
10. Force `input2` to 16, which is `10` in hexadecimal (the default radix).
11. Now, run the simulation for `10 ns` by clicking the play button with `(T)` underneath, or pressing `Shift+F2`.
    - While holding `Ctrl`, you can use the mouse scroll wheel to zoom in and out in the waveform view.
    - While holding `Shift`, you can use the mouse scroll wheel to navigate horizontally along the waveform view.
    - If you accidentally close the waveform view, go to `Window` in the toolbar, and click `Waveform`.
12. Now, we'll start shifting the data until the value of the `shift_reg` signal is `0`:
    - Force `load` to `0`.
    - Force `shift` to `1`.
    - Run the simulation repeatedly until the value of `shift_reg` is all zeroes.
        - If you chose a value with a number of high bits greater than the value of `input2`, then `output` will be high when you are done.
13. At this point, take screenshots of your simulation results for your report.

Deliverables

- Include as part of your **informal report**:
  - A screenshot of the Elaborated Design schematic (Step 4)
  - One or more screenshots of your simulation results (Step 13)

Outcomes

- Practice working with VHDL.
- Practice using Vivado for hardware simulation.
- Understand VHDL components, instantiation, and structural connections.
- Understand shift registers and accumulators.