10 Simulating Designs with Vivado

This appendix will walk you through the process of simulating HDL code (such as Verilog HDL) using Xilinx Vivado's built in simulator.

A Introduction

It is expected that you have gone through the Vivado Quick Start guide and know how to create a Vivado project and go through the synthesis and implementation steps. This tutorial merely covers how to simulate your design in Vivado. There are three types of simulations used to test a design: (1) Behavioral simulation, (2) Functional simulation, and (3) Timing simulation. To check the logical correctness of a design, behavioral and functional simulations are used. Behavioral simulations are typically done on HDL prior to synthesis/implementation to verify the design behavior. Whereas functional and timing simulations can be used at the hardware level after synthesis and implementation. Functional simulations verify that the functionality is preserved after synthesis, and timing simulations provide information about the timing properties of the design and FPGA.

B Steps

First, follow the steps in the Vivado Quick Start guide. to get a project set up and have a verilog source file for upcount.v. For more details on the Vivado simulator, refer to the in-depth tutorial.

1. We will simulate an example of an Up Counter. The Verilog code used for this simulation is on Listing 1 in the Vivado Quick Start guide.

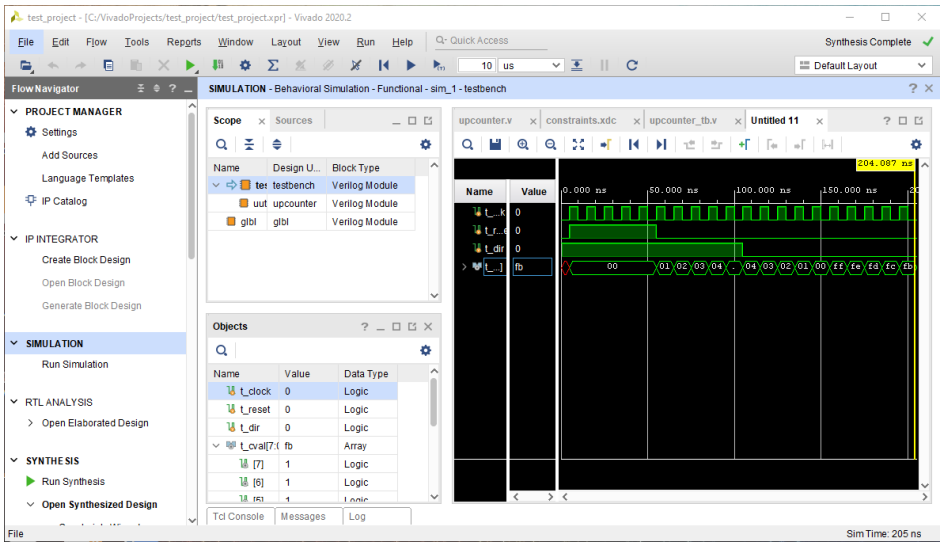

Figure 1: Run behavioral simulation, view waveforms and verify functionality.

2. In the "Sources" window, right-click on "Simulation Sources" and add a new source. Proceed through the prompt to create a new verilog file named "upcount_tb.v". The source for this testbench is provided in listing 2.
3. On the left panel, click on "Run Simulation" then select "Run Behavioral Simulation". The full simulation from the testbench should be ran. To view the waveforms of the outputs, open

the "Untitled X" tab that was generated from the simulation. The output should look like what is shown in Figure 1.

4. Within this interface, you can visually inspect the waveforms for any input and output signals in the design. If you have any errors, make sure that the testbench file is the top module in the simulation sources.

Listing 1: Test-bench for Up Counter

```
module testbench;

        // Inputs
        reg t_clock, t_reset, t_dir;

        // Outputs
        wire [7:0] t_cval;

        // Instantiate the Unit Under Test (UUT)
        counter uut (
                .clock(t_clock),
                .reset(t_reset),
                .dir(t_dir),
                .cval(t_cval)
        );

        initial begin
                // Initialize Inputs
                t_clock = 1'b0;
                t_reset = 1'b0;
                t_dir = 1'b1;
                $monitor($time, "\t%h", t_cval);
        end
                // Add stimulus here
        always begin
                #5 t_clock = ~t_clock;
        end

        initial begin
                #5 t_reset = 1'b1;
                #5 t_reset = 1'b0;
                #50 t_dir = 1'b0;
                #100 $finish;
        end
endmodule
```

References Cited