Lab 2 – Behavioural Simulation

CS 2052 Computer Architecture

Dept. of Computer Science and Engineering, University of Moratuwa

Learning Outcomes

In this lab we will learn how to simulate a logic circuit using software. After completing the lab, you will be able to:

- design and develop a simple logic circuit using schematics
- verify its functionality via simulation
- verify its functionality on the development board

Introduction

A logic simulator allows us to observe the outputs of a circuit in response to all possible combinations of inputs before the circuit is implemented in hardware. Simulating a circuit is perhaps the best technique an engineer can use to ensure that all required features are present, and there are no unintended behaviours. For larger circuits, simulation is far cheaper and far less error prone than designing and testing a hardware prototype. If errors are observed in the simulator's output, the circuit can easily be corrected and re-simulated as often as necessary.

The Xilinx ISim simulator allows simulations to be run from the Xilinx Project Navigator. This process involves two steps. In step one, we create the desired logic circuit. In step two, the circuit is tested by providing a set of stimulus values that define all input logic.

Building the Circuit

In this lab, we will build a logic circuit that is capable of detecting prime numbers between 0 and 15. We will use four switches **A** (SW3), **B** (SW2), **C** (SW1), and **d** (SW0) to indicate a number (**A** is the most significant digit and **d** is the least significant digit) and an LED **X** (LD0) to indicate whether the given input is a prime number (X will be 1, if the number is prime) or not.

Step 1: Logic Equation

Use a suitable mechanism find the simplified Boolean representation of this circuit (i.e., $X = F(A, B, C, D) = \Sigma\ (2, 3, 5, 7, 11, 13)$).

Step 2: Building the Circuit

Create a new project in Xilinx ISE Design Suite. Name the project as **Lab 2**.

Tips Review the steps given in Lab 1, if you do not remember how to perform some of the tasks in building a circuit or what parameters to use.

Create the logic circuit using suitable components. Name the circuit as **Circuit 2**. Create a User Constraints File that defines how to connect the switches and LED to the correct pins. Name it as **PinAssignment**.

"Synthesize – XST" and make sure there are no errors in your design.

Behavioural Simulation Using ISim

Next, we will check the behaviour of our circuit by defining various input combinations. Though our circuit is defined using a schematic diagram, we need to write *test bench code* to indicate the input combinations (ISE does not support any other mechanism). While the test bench code can be written in either VHDL or Verilog hardware description languages, we will use VHDL in this lab. While learning VHDL is not required for our labs, it will be useful if you wish to work closely with hardware in the future.

Step 1: Selecting ISim as simulator

To select ISim as our project simulator, do the following:

In the Sources window (a.k.a. Hierarchy pane) of the Project Navigator Design panel, right-click the device line (**xc3s100e-4cp132**), and then select **Design Properties…** form the popup menu.

In the Design Properties dialogue box, set the **Simulator** field to **ISim (VHDL/Verilog)**. Click **OK** button.

Step 2: Generating Test Bench File

Click on the **Simulation** radio button on the Project Navigator Design panel (see Fig. 1). Then from the drop-down menu select **Behavioral** (see Fig. 1).

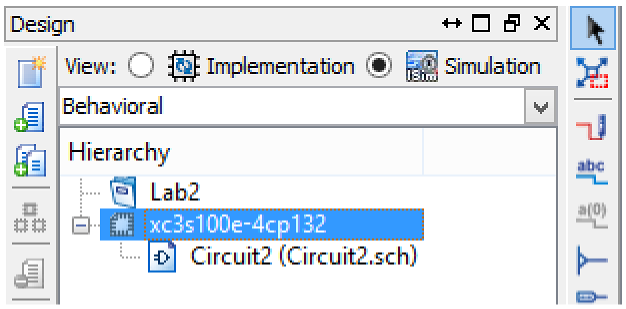


Figure 1 – Setting behavioural simulation mode.

To add a new test bench files to the project, right click on the target device **xc3s100e-4cp132** and select **New Source…** from the pop-up menu.

Add a **VHDL Test Bench** file and name is as **TB**. Click **Next**, **Next**, and **Finish** buttons in the given order.

You should see a new file named **TB.vhd** in the Hierarchy tree.

Step 3: Modify Test Bench File

If the **TB.vhd** file is still not open, open it by double clicking on it. Locate the following lines in the file.

```
-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
      WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***
```

Lines starting with "--" indicate comments.

Modify the user defined section in the TB.vhd as follows to try a several input combinations. Once we set a particular input combination, we retain that state for 1 ns before changing the inputs to another combination.

```
-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
           -- set initial values
           A <= '0';
           B <= '0';
           C <= '0';
           D <= '0';

           -- after 1 ns change inputs
           WAIT FOR 1 ns;
           A <= '0';
           B <= '1';
           C <= '0';
           D <= '0';

           --change again
           WAIT FOR 1 ns;
           A <= '1';
           B <= '0';
           C <= '0';
           D <= '1';

           WAIT FOR 1 ns;
           A <= '1';
           B <= '0';
           C <= '1';
           D <= '0';

      WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***
```

Save the file. Double click on **Behavioral Check Syntax** in the **Processes** window (expand the **ISim Simulator** tree if it is not already expanded). If there are no errors in your TB.vhd, you will see a message similar to the following:

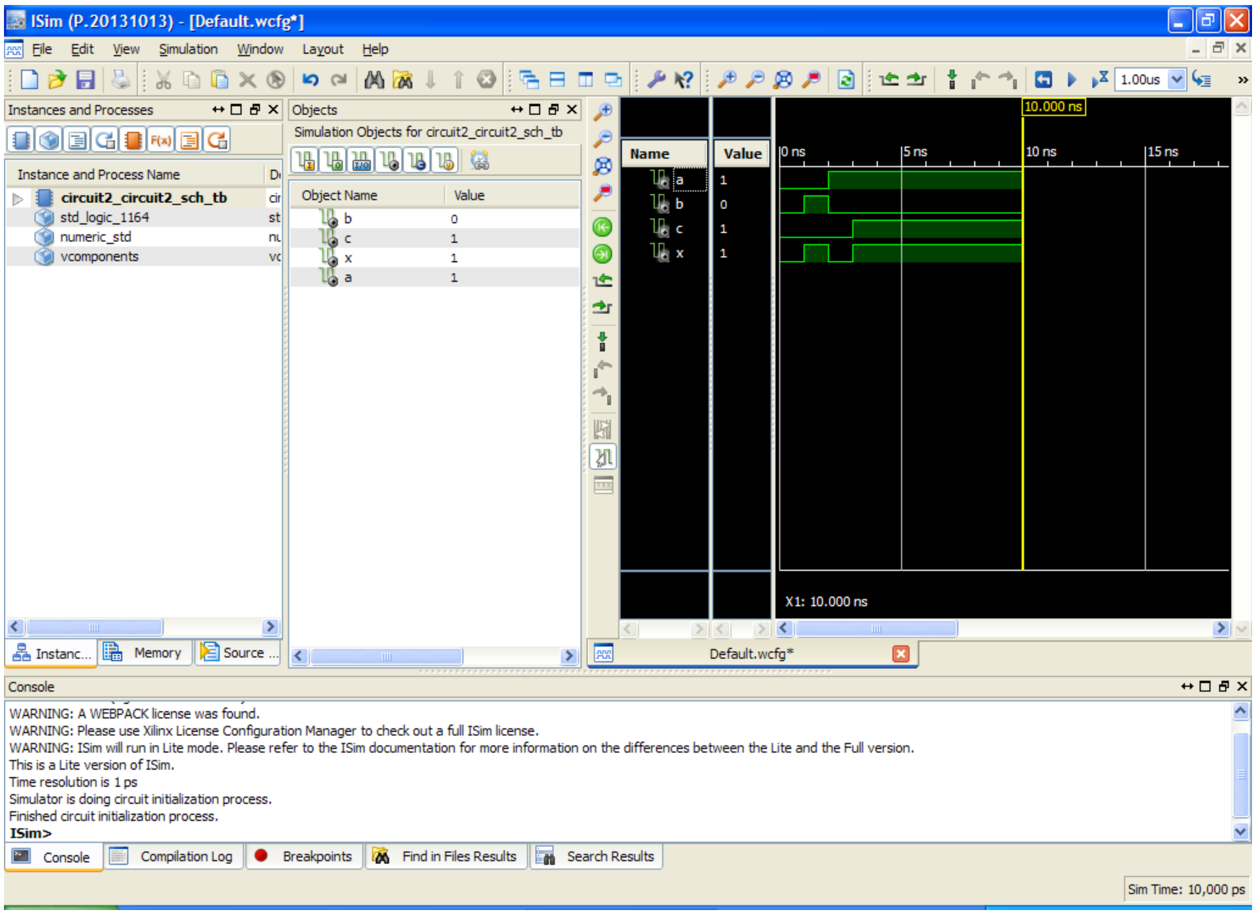
```
Process "Behavioral Check Syntax" completed successfully
```

If not, go back and double check the VHDL code.

Step 4: Simulating Circuit

Right click on **Simulate Behavioral Model** in the **Processes** window and then select **Process Properties…** from the popup menu. This will open up Process Properties dialogue box. Suppose we want to simulate our circuit for 20 ns. Therefore, set **Simulation Run Time** to 20 ns. Click **OK** button to close the dialogue box.

To start the behavioural simulation, double click on the **Simulate Behavioral Model** in the **Processes** window. ISim then creates the work directory, compiles the source files, loads the design, and performs simulation for the time specified. After a few seconds ISim window similar to Fig. 2 will open up.



Object Name	Value
b	0
c	1
x	1
a	1

```
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Figure 2 – ISim window.

ISim window has four components:

1. Source Files panel – use to select source files to be viewed
2. Objects panel – use to add different signals to the simulation
3. Simulation panel – use to observe the state of signals
4. Console panel – accept commands from the user

Use the Zoom buttons on the toolbar (especially **Zoom to Full View** button) to zoom the timing diagram. ISim by default simulates the circuit in 1 ps intervals. Hence, zooming is essential to clearly see the changes in inputs and outputs. You may also use **Go To Time 0** and **Go to Latest Time** buttons to navigate further. Note how the inputs and the corresponding output changes. Do the outputs confirm the correct functionality of your circuit? If not, double check your Boolean equation and schematic circuit.

It is not sufficient to validate your design only for a subset of the all possible input combinations. While we can extend the above VHDL test bench code to check for other combinations of inputs, next we will discuss a better way of setting all possible inputs.

We will vary the four inputs A, B, C, and D as alternating square waveforms. As A is the most significant bit, its period (8 ns) will be higher than B, C, and D. D being the least significant bit, will have a shorter period (1 ns). Modify the user defined section in the TB.vhd as follows:

```
-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
           --Make sure to set initial values for A, B, C,& D
           --Repeat signals to form waveforms
           ABC_loop: LOOP
                WAIT FOR 1 ns;
                D <= NOT D;  --invert D
                WAIT FOR 1 ns;
                D <= NOT D;
                C <= NOT C;
                WAIT FOR 1 ns;
                D <= NOT D;
                WAIT FOR 1 ns;
                D <= NOT D;
                C <= NOT C;
                B <= NOT B;
                WAIT FOR 1 ns;
                D <= NOT D;
           -- fill the rest to get proper sequence
                WAIT FOR 1 ns;
                D <= NOT D;
                C <= NOT C;
                B <= NOT B;
                A <= NOT A;
           END LOOP ABC_loop;
      WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***
```

Save the file. Rerun ISim and verify the functionality of your circuit.

Step 5: Test on BASYS2

Generate the programming file (i.e., bitstream) and load it to the BASYS2 board.

Change the switches (SW0 – SW3) on the BASYS 2 and verify the functionality of your circuit (check the output on LED LD0).

Demonstrate the circuit to the instructor and get the Lab Completion Log singed.

Step 6: Lab Report

You need to submit a report for this lab. Your report should include the following:

- Student name and index number. Do not attach a separate front page
- State the assigned lab task in a few sentences
- Steps showing how you derived the simplified Boolean representation
- Schematic circuit from ISE (label figures)
- Test bench code
- Timing diagram from ISim showing all possible inputs and outputs (label figures)
- Conclusions from the lab

Submit the lab report at the beginning of the next lab.

Tips If you do not have access to a printer while working on ISE, you may save/print schematics and timing diagrams as PDF files and later print them.

Bibliography

- Digilent Inc., "Xilinx ISE Simulator (ISim) with Verilog Test Fixture Tutorial", Feb. 26, 2010.
- Shawki Areibi, "Tutorial On Using Xilinx ISE Design Suite 13.2: Behavioral Simulation of a "Half Adder" Circuit", Jan. 2014.
- Xilinx, "ISim User Guide", July 25, 2012.

Prepared By

- Dilum Bandara, PhD – Feb 04, 2014.
- Updated on Aug 31, 2017