L-Università ta' Malta

DEPARTMENT OF COMMUNICATIONS AND COMPUTER ENGINEERING

Faculty of Information and Communication Technology

CCE1013—Computer Logic 1

Practical 1

Trevor Spiteri
trevor.spiteri@um.edu.mt
http://staff.um.edu.mt/trevor.spiteri

23 October, 2018

Combinational logic in Xilinx ISE

Objective

The objective of this part is to use the Xilinx ISE Design Suite to simulate simple combinational circuits.

Tasks

1. Open the Xilinx ISE environment, and create a new project using the menu item *File: New Project...*. Use the following options (other options can be left untouched):

    - Top-level source type: HDL
    - Family: Virtex 5
    - Device: XC5VLX50T
    - Package: FF1136
    - Synthesis tool: XST
    - Simulator: ISim
    - Preferred language: VHDL

TABLE 1: TRUTH TABLE.

A	B	C	F	G	H
0	0	0	1	0	1
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	0	1
1	1	0	0	0	0
1	1	1	0	1	0

2. Create a new source file using the menu item *Project: New Source...*. The source type should be *VHDL Module*. Set three input ports `A`, `B` and `C`, and three output ports `F`, `G` and `H`.

3. In the architecture body (near the bottom of the new file, between the last **`begin`** and **`end`** keywords), add VHDL code to set the outputs according to the truth table in Table 1. The line for `F` is given as an example below:

```
F <= (not A and not B and not C)
     or (not A and B and C)
     or (A and not B and C);
```

   Note that the example has three rows, one for each true value in the truth table for F. A similar expression with four rows can be determined for G, and a similar expression with five rows for H.

4. After VHDL entry, the design should be simulated and verified. This is called functional simulation, as it checks the functionality of the code. To do this, first generate a test bench.

   (a) Create a new file using the menu item *Project: New Source...*. This time, the source type should be *VHDL Test Bench*. The filename can be `tb_` followed by the filename of your first VHDL file. Associate the original VHDL file with the testbench.

   (b) Since this design contains combinational logic only, search for the parts of the test bench that are for the clock and delete them. (There are two such parts to be deleted.)

(c) Find the stimulus process and replace it with the following code:

```
a_proc: process
begin
    A <= '0';
    wait for 200 ns;
    A <= '1';
    wait for 200 ns;
end process;
```

(d) Write similar processes to stimulate `B` with 100 ns delays instead of 200 ns delays, and `C` with 50 ns delays.

5. In the top left panel, select the *Simulation* view, Make sure that the *Behavioral* simulation is selected. Select your test bench. Then, use the *Simulate Behavioral Model* option in the process panel at the bottom left.

6. In the simulation window, verify that the outputs follow the truth table in Table 1.

7. For the next test, we are going to introduce some artificial glitches. In the test bench, in the processes that stimulate the `B` and `C` inputs, modify the following four lines:

   (a) Change `B <= '0';` to: `B <= '0' after 3 ns;`

   (b) Change `B <= '1';` to: `B <= '1' after 3 ns;`

   (c) Change `C <= '0';` to: `C <= '0' after 5 ns;`

   (d) Change `C <= '1';` to: `C <= '1' after 5 ns;`

   Save the test bench file.

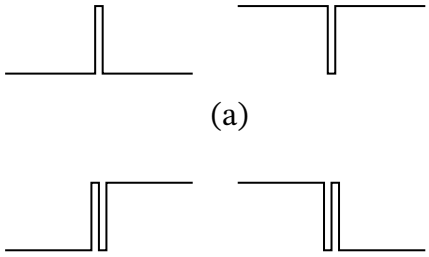
8. Simulate the design once again using *Simulate Behavioral Model*.

9. Observe that some glitches are introduced by this change. The glitches are called hazards. Figure 1 shows two kinds of hazards:

   Static hazards are those where the levels before and after the hazard are the same, e.g., the initial value is `'0'`, then there is a short `'1'`, and the final value is `'0'` again.

   Dynamic hazards are those where the levels before and after the hazard are different, e.g, the initial value is `'0'`, then there is a short `'1'` followed by a short `'0'`, and the final value is `'1'`.

   List the static and dynamic hazards introduced in the outputs. What do you think causes these hazards?



(a)

(b)

Figure 1: (a) Static hazards and (b) dynamic hazards.

Report

Your report should describe your work concisely. It should include the VHDL code you wrote (of course, this includes both the code of the design and of the test bench.) You should also show the waveform that verifies that the design works.