Digital Systems Electronics Laboratory 01

Group 13

s295391 Giorgio Zoccatelli
s294422 Lorenzo Iemmulo
s295567 Vittorio Macripò

Due Date:	March 21, 2024
Delivery date:	March 20, 2024
Instructor:	Professor Guido Masera

Politecnico di Torino
Accademic Year 2023/24

Contents

1 Introduction

The aim of this laboratory is to get familiar with the FPGA DE1-SoC by implementing three simple examples. In the first example we want to control the ten LEDs on the board by using ten switches while in the second and third examples we want to implement two different types of multiplexer: a four bit wide two-to-one multiplexer and a three bit wide five-to-one multiplexer.

2 Controlling the LEDs

2.1 Design Entry

First of all we want do describe the steps used to reach the design entry reported in *led.vhd*. We have to open the software Quartus Prime, select **New Project Wizard**, choose the directory for the file and choose the file name which has to match the top entity name in the design file. Then we skip until the voice **Family, Design & Board Settings** and we select the device **5CSEMA5F31C6** which is the FPGA used on Altera's DE1-SoC. We can now proceede with the following path to open a new clean sheet where we can write the actual design entry: **File** → **New** → **VHDL File**. Another important step to succesfully compile the design entry is to select a proper pin assignemt based on the FPGA we have to use to correctly refer to the pin we want to use, so we have to go to **Assignments** → **Import Assignments** and we select the file reporting the pin assignment characterized by the *.qsf* extension (in our case the file *DE1_SoC.qsf* was given). After the completion of the design entry we can compile the file to find eventual errors following the path **Processing** → **Start Compilation**.

2.2 Functional Simulation

The Functional Simulation consists in a testbench of the device under test. Therefore we choose some input configurations to check in the next step if the design implemented works as we expect. Basically we have to create a new VHDL file following the exact steps of section 2.1. We can

```
SW_tb <= "1010101010";

WAIT FOR 20 ns;

SW_tb <= "1111100000";

WAIT FOR 20 ns;

SW_tb <= "0000011111";
```

Figure 1: Chosen configurations

notice looking at Figure 1 that the three configurations should show these results:

1. Alternating lit LEDs

2. Half of the LEDs lit half off

3. Complementary configuration of configuration number 2

2.3 Synthesis

The last step of our simulation is to emulate the behavior of the configuration in our testbench by using Modelsim. After the opening of the software we will follow this path: **File** → **New** → **Project** then under the voice **Project Name** we will match the top entity name of our project and we will choose a new directory for the simulation under the voice **Project location** (we can also rename the **Default library name**). A new screen will show up asking us which files we want to import in our simulation. We will select the voice **Add existing file** two times to select both the design entry and the testbench (in this exercise we are refering to *led.vhd* and *led_tb.vhd*). It is then necessary to follow this path to assure that the design entry will be compiled before the testbench: **Compile** → **Compile Order** → **Auto Generate** (we can also manually select the wanted order).Now we can start the actual simulation by clicking on **Simulate**→ **Start Simulation** and under the directory of our Default library name we select the testbench. We can notice at a first glance that there are no items in the window panel we want to analyze so we have to go under the voice **Add** → **To wave** → **All items in design** and then **Simulate** → **Run all**. We highlight in Figure 2 the transition between the first chosen

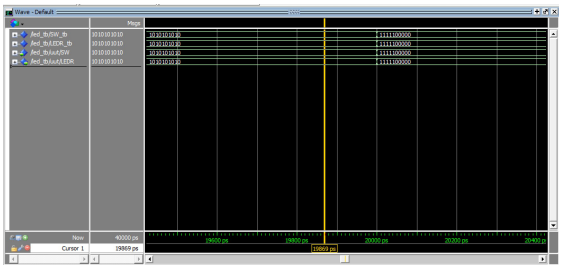

Figure 2: Wave analyzer

configuration and the second chosen configuration discussed in section 2.2.

3 2-to-1 Multiplexer

3.1 Design Entry

3.2 Functional Simulation

3.3 Synthesis

4 5-to-1 Multiplexer

4.1 Design Entry

4.2 Functional Simulation

4.3 Synthesis