Oblig 1

Design Flow

IN3160 / IN4160

Version 2.0/01.02.2023

Note! Before you start, read this: "Mandatory assignments and other hand-ins at the Department of Informatics" at https://www.uio.no/english/studies/examinations/compulsory-activities/mn-ifi-mandatory.html **.**

The goal of our first lab exercises is learn how and to get practical experience in making digital designs. It is the first step in order to be able to "understand important principles for design and testing digital systems", and to "be able to perform simulation and synthesis of digital systems" See course description.

Vivado is used to implement a design, that have been tested, on the Zedboard. In Vivado we will assign which pins on the Zedboard are connected to the ports of out design and more.

For simulation and testing of our VHDL design file we will use GHDL as compiler and simulator, gtkwave to view the waveforms from the simulaton. To test the design a python testbench built with the cocotb framework will be used.

https://docs.cocotb.org/en/stable/index.html
https://ghdl.github.io/ghdl/

All the submitted VHDL files must follow the indenting guidelines as described in the cookbook chapter "Editing VHDL code".

Table 1. Attached files.

Filename	Description
`first.vhd`	VHDL source file
`tb_first.py`	Python test bench
`makefile`	Makefile to build and run the testbench

The file *makefile* must be modified to use your own path.

a)

Read chapter 1, *Design flow and tools,* in the cookbook, and follow the description in the cookbook chapter 2, *Getting started with Questa (Getting started with GHDL/gtkwave not available for beta)*. All necessary files can be obtained from the course website. Simulate the chip at the RTL level. Afterwards, the design shall be downloaded to the test board: Follow the instructions in chapter 3, "*Getting started with Vivado", 3.1- 3.3.2, 3.4-3.6.2*. (Note: 3.3.3 and 3.3.4 are not required for this task)

Before you program the chip, look at the *IO report* to check that the pin numbers are placed correctly. This means checking the pin assignment stated in the *ZedBoard Hardware User's Guide*: Zedboard Hardware User's Guide form Avnet

Check the *Timing summary report* to see whether the timing constraints have been satisfied.

Approval:

No special requirements.

b)

Change the VHDL code so that the counter becomes an up/down counter by adding an `up` signal. The `up` signal should be assigned to SW6 (pin H17). If `up='1'` the counter should count up, and if `up='0'` the counter should count down. In addition, we will add a `min_count` signal, which will give a pulse with a duration of 1 clock period when the counter goes to 0 on the way down. `min_count` shall be assigned to LD6. `max_count` will be modified so that it will give a pulse at the maximum value on the way up. Add the necessary modification to the XDC file to Vivado so that the `up` and `min_count` signals are assigned to the correct pin numbers. Modify the test bench so that we can verify that the counter is functioning correctly. A counting sequence from 0 to 15, and then reversal of the direction and counting down to 0, for example, is fine. Perform the simulation, verify the pin placement and program the chip when everything appears to be correct.

It is sufficient to check that the design behaves as indented by looking at the waveform with gtkwave. No need for an automated testbench with assertions etc.

Hint: it is possible to check the value of a signal by using doing:

```
if dut.portname.value = 0b1010. # True when the value of portname is binary 1010.
```

Approval:

Modified VHDL file, test bench, modified XDC file and IO report.