Document Type:	Lab report
Version:	1.0
Creation date:	April 23, 2019
Author:	Jan Burchard & Jan Kühn, Laboratory for Electrical Instrumentation, IMTEK
Contact:	Philip Klein (klein@telocate.de), Peter Härlen (peter.haerlen@students.uni-freiburg.de), Fabian Hoeflinger (fabian.hoeflinger@imtek.uni-freiburg.de)

Basic information about the laboratory exercise

Title: The blinking LED

Category: Basic VHDL

Topic: Brush-up of VHDL knowledge

Level of Difficulty: 2

Estimated duration time: 8 h

What will be learned

Controlling the blinking frequency of an LED in VHDL. Reading the state of a button.

Materials

E2LP development board, LED, buttons.

Environment setup

No special preparation required

Abstract & Motivation

The blinking LED example from the tutorial is expanded to allow for different blinking frequencies. The frequency can be selected with the buttons connected to the E2LP board and is displayed on some additional LEDs.

Theoretical part of the laboratory exercise (Theoretical background)

The blinking LED provides a first introduction to frequency dividers in VHDL, The buttons as the most simple input device allow for basic user input and are therefore the first step towards interaction with the FPGA.

In this exercise we will use a schematic approach to interconnect multiple VHDL modules. This is an easy way to visualize how multiple modules are interconnected. It is also possible to implement the interconnections in vhdl using *modules* and *port maps*.

Practical part of the laboratory exercise (experimental design and procedure)

Exercise 1 (2 Points)

Extend the code from the tutorial such that the blinking frequency of the led can be controlled with an additional 3-bit input *freq* between always off (0) and very fast blinking (7).
Create a *schematic symbol* of the VHDL module by double clicking *Create Schematic Symbol* in the Design-Bar on the left side of the screen (see Figure 1).

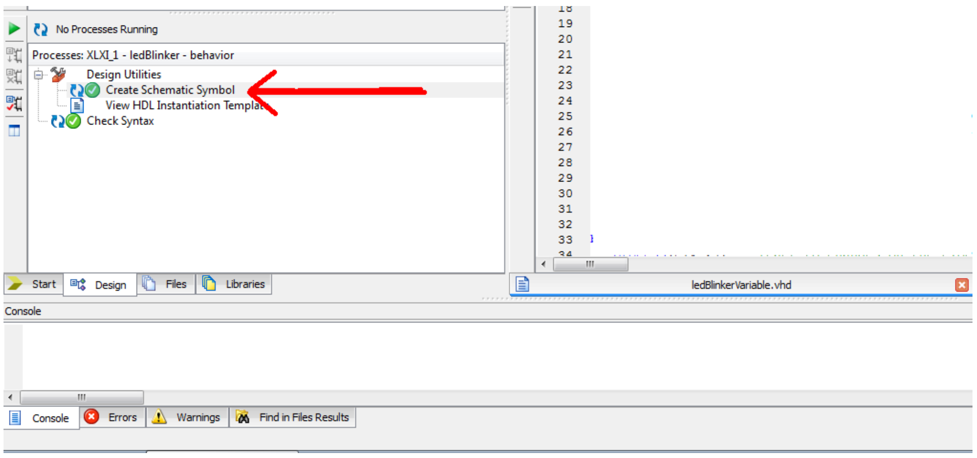

Figure 1: Creating a schematic symbol of a VHDL module.

Exercise 2 (1 Point)

Create a new schematic file by creating a *New Source* and selecting *Schematic*. Give this schematic the name *wholeDesign*. This schematic will be used to connect the VHDL modules together in a simple graphical way. Right click on the newly created file and select *Set as Top Module*.

Now open the new schematic. On the left side, select symbols. You should see the ledBlinker symbol in the list of symbols. Add it to the schematic (see Figure 2).

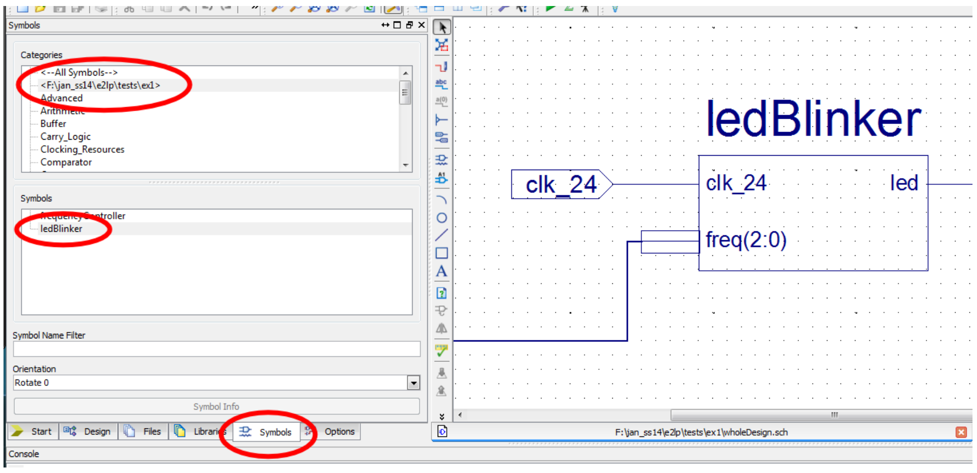

Figure 2: Adding a schematic symbol to a schematic file.

Create a second VHDL file *frequencyController*. For now, this module only requires a single 3-bit output *freq*. To test your implementation of exercise 1, simply output a fixed value 3-bit value with the frequency controller.

Create a *schematic symbol* of the new module and add it to the schematic file. Add a wire to connect the *frequencyController* to the *ledBlinker*. Also add an *I/O marker* to the input *clk_24*, the output *led* and to the connection between *frequencyController* and *ledBlinker* (see Figure 3). Rename the I/O markers in a useful manner. Assign these markers to FPGA pins in *PlanAhead* (similar to the tutorial). Assign *freqOut* to 3 adjacent LEDs.

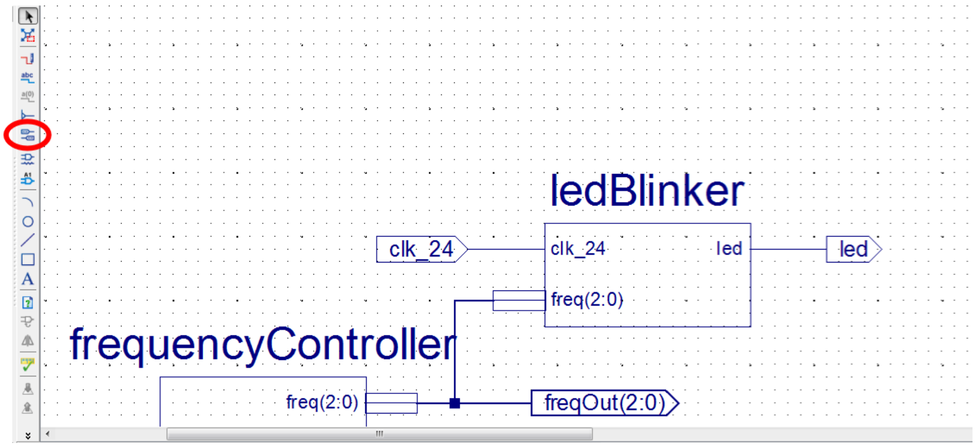

Figure 3: The completed schematic with 2 VHDL modules and I/O markers for any in- and outputs.

Compile the finished design and transmit it to the FPGA. Test you implementation of exercise 1 with different values for *freq*.

Exercise 3 (2 Points)

Extend the module *frequencyController* with two additional inputs *buttonUp* and *buttonDown*. Recreate the *schematic symbol* and connect the new inputs to *I/O markers*. Then, in *PlanAhead*, connect the markers to the FPGA pins for the up and the down button on the E2LP board.

Add additional logic to *frequencyController* such that the blinking frequency can be changed with the two buttons. Please note: the buttons are active low (they produce a logic '0' when pressed). Make sure that a single button press is only registered once. You may add an additional clock input in order to implement debouncing and edge detection.

The buttons on the E2LP Frontpanel are connected to the FPGA in the following order:

Button	FPGA Ball
bottom	AC24
left	AC23
center	AB24
right	AA24
top	AA23

Hand In

Please hand in project folder as a zip file. You do not need to provide an extra implementation of exercise 2 if you have completed exercise 3.

If you have any constructive criticism or suggestions about this exercise sheet, feel free to state theese in a textfile named *experience.txt* and add it to the zip file.