E2LP-Praktikum 2017

Prof. Dr. Leonhard Reindl, Dr. Rui Zhang,
Dr. Fabian Höflinger (fabian.hoeflinger@imtek.uni-freiburg.de)
IMTEK, Albert-Ludwigs-Universität Freiburg, Freiburg Im Breisgau

Tutorial

Goal: This tutorial covers the installation and configuration of the software required to program the E2LP board. Additionally, a small VHDL example is created and programmed onto the FPGA.

Required Software

- Xilinx ISE WebPack (version 14.7):
  http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html
- E2LP software installer (this includes the required drivers): *E2LP_Installer.exe*

Installation

Please note: it is very important to install the E2LP software **after** the Xilinx software!

1. Install Xilinx ISE with the recommended settings. When asked, select *Get Free Vivado/ISE WebPack License*. On the Xilinx Website select *Vivado Design Suite (includes ISE): WebPACK License*. The free license will be sent to you via email.
2. Open Xilinx ISE. The license manager should open automatically. Load the license file received in the previous step and close the license manager.
3. Add the Xilinx binary folder to the windows *PATH* variable:
   - On Windows 7 open the control panel (Systemsteuerung)
   - Click on *System*, then in the left pane on *Advanced System Settings*.
   - Click on *Advanced*, then at the bottom onto *Environment Variables...*.
   - Find the variable *Path* in the System variables and double click it.
   - At the end of the Variable value, add a semicolon (;), then the path to the Xilinx binaries, i.e. *C:\xilinx\14.7\ ISE_DS\ISE\bin\nt*
   - Close all the open windows with *OK* and restart your computer.
4. Install the E2LP software. **Note:** this software has to be installed into a folder with full write rights (i.e. **not** the default location).
5. Connect the E2LP board to the computer and switch it on. The required drivers will be automatically installed. You should now be able to connect to the board in the *E2LP Config Utility*.

Creating a new project

- Open the Xilinx ISE software and click on *File* → *New Project*.
- Enter a name and location for the project. As Top-level source type select *HDL* and click on *Next*.
- Select the following values:

Family:	Spartan6
Device:	XC6SLX45
Package:	FGG676
Speed:	-3
Synthesis Tool:	XST
Simulator:	ISim
Preferred Language:	VHDL

- Click on *Next* and *Finish*. Congratulations, you are now ready to write code for the FPGA!

Blinking LED example

In this section a small VHDL example is implemented. The goal is to let one of the LEDs of the E2LP board blink with a frequency of 1 Hz.

- Create a new project as described on the previous section. Name it *tutorial*.
- In the design-panel on the left, right click on *xc6slx45-3fgg676* and select *Add Source...*.
- Chose *VHDL Module* and name the file *ledBlinker*. Click *Next* and *Finish*.
- Copy the following VHDL code into the newly created file (overwriting the original file content) or overwrite the whole file with the *ledBlinker.vhd* which comes with this tutorial.

```
-- LED blinking example

-- include the standard libraries
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

-- define the module
entity ledBlinker is port(
   clk_24 : in std_logic;     -- the 24 MHz clock of the FPGA
   led    : out std_logic     -- output of the led signal
);
end ledBlinker;

-- define the behavior of the module
architecture behavior of ledBlinker is
   -- internal storage of the output signal
   signal led_int : std_logic := '0';

   -- frequency = 1 Hz -> 2 signal changes per second
   -- incoming clock has 24 MHz -> need to count to 12,000,000
   -- -> 24 bit counter required
   signal counter : std_logic_vector(23 downto 0) := (others => '0');

begin
   process(clk_24) -- react to changes on the clock
   begin
      if rising_edge(clk_24) then  -- positive edge triggered logic
         if unsigned(counter) = 12000000 then
```

```
            counter <= (others => '0'); -- reset all bit to 0
            led_int <= not led_int;     -- change led state
         else
            -- increment the counter
            counter <= std_logic_vector(unsigned(counter) + 1);
         end if;
      end if;
   end process;
   -- connect internal storage to outgoing signal
   led <= led_int;
end behavior;
```

- Make sure you understand the above code!
- Click on the green *Implement Top Module* arrow to synthesize the VHDL code.
- Next, click on *Tools* → *PlanAhead* → *I/O Pin Planning - Pre-Synthesis*. Click on *Yes* when asked to create a new *ucf* file.
- Assign the following FPGA pins to the pins of the VHDL Design:

Name	Site
clk_24	M21
led	D24

  The complete list of pin assignments for the FPGA can be found in the file *E2LP-Deliverable-D3.2*.
- Close *PlanAhead* and save the changes to the file. Now the Design is ready to be transmitted to the FPGA. Resynthesize the VHDL (green arrow). Then, in the Design-Panel double-click on *Generate Programming File*.
- To transmit the programming file to the FPGA, use the *E2LP Config Utility*: Click on *Open BIT File* and select the file *ledblinker.bit*.