ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

Hardware Implementation of Algorithms

1. Introduction to ISE Webpack. Combinational logic.

Lead Partner: Warsaw University of Technology

Authors: Lukasz Mik

University of Applied Sciences in Tarnow

Erasmus+

GRANT NUMBER: 2020-1-PL01-KA226-HE-095653

Declaration

This laboratory instruction has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

Funding Disclaimer

This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

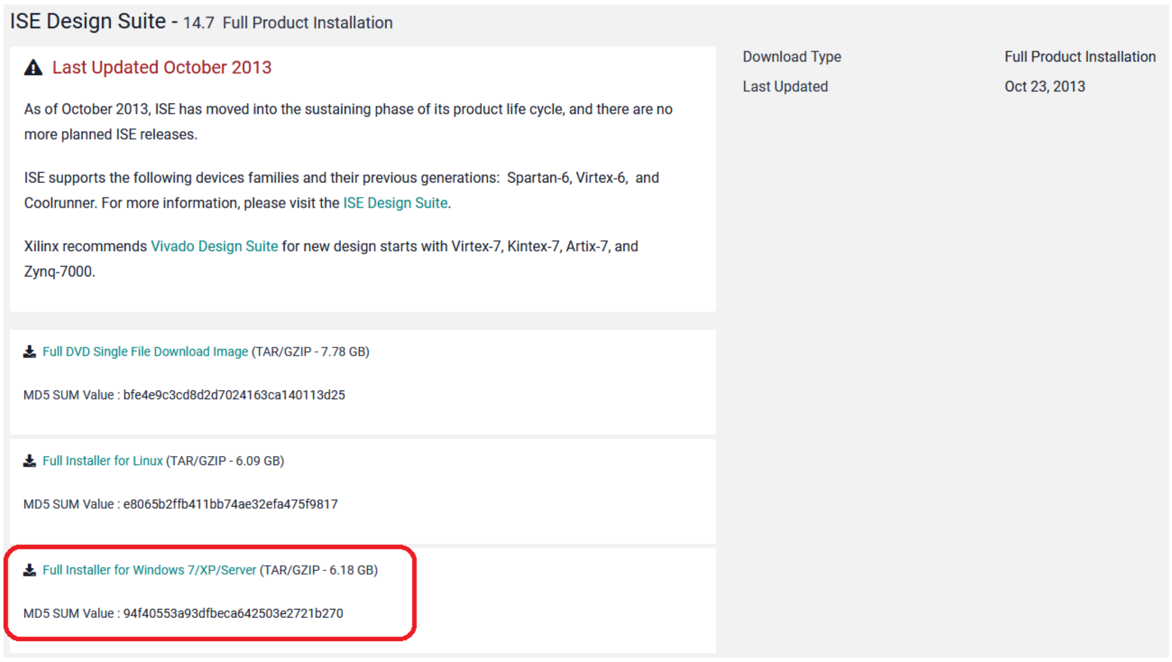
I. Installation of ISE WebPack software

ISE Webpack is a free Xilinx tool that enables the design of digital systems from synthesis and simulation, through implementation, device fitting and finally JTAG programming. The newest version is available on Xilinx website and is adapted for Linux and Windows operational systems.

STEP 1: Downloading installation files from Xilinx website.

Go to the website: https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html and click the link called *Download ISE WebPACK software for Windows and Linux*. If you have Windows 10 Pro or Enterprise version, you can download the latest version of ISE Design Suite 14.7 with integrated ISE Webpack software, which must be run from the virtual machine. It is a less convenient version for users, therefore I suggest staying with ISE Design Suite 14.7 for older Windows7 / XP / Server systems. You can use it in Windows 10 after making minor adjustments to the application installation directory. The procedure is described on the manufacturer's website at https://support.xilinx.com/s/article/62380?language=en_US and discussed in this manual.

You must have an account on the manufacturer's website to download the file. If you have already created an account, you can log in, otherwise you have to create one by entering your university e-mail address and select the function performed at the university (in the *Job Function* drop-down list, select *Student*). From the manufacturer's website, download the 6.18 GB TAR / GZIP file to which the link is shown in the figure below.



ISE Design Suite - 14.7 Full Product Installation

Last Updated October 2013

As of October 2013, ISE has moved into the sustaining phase of its product life cycle, and there are no more planned ISE releases.

ISE supports the following devices families and their previous generations: Spartan-6, Virtex-6, and Coolrunner. For more information, please visit the ISE Design Suite.

Xilinx recommends Vivado Design Suite for new design starts with Virtex-7, Kintex-7, Artix-7, and Zynq-7000.

Download Type	Full Product Installation
Last Updated	Oct 23, 2013

Full DVD Single File Download Image (TAR/GZIP - 7.78 GB)

MD5 SUM Value : bfe4e9c3cd8d2d7024163ca140113d25

Full Installer for Linux (TAR/GZIP - 6.09 GB)

MD5 SUM Value : e8065b2ffb411bb74ae32efa475f9817

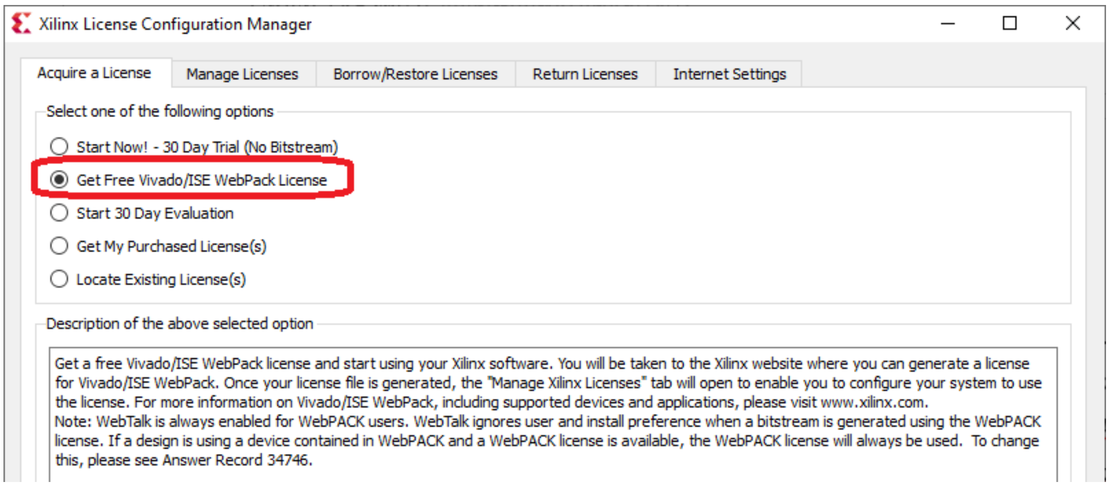
Full Installer for Windows 7/XP/Server (TAR/GZIP - 6.18 GB)

MD5 SUM Value : 94f40553a93dfbeca642503e2721b270

STEP 2: Installing the ISE Webpack software 14.7.

After unpacking the downloaded archive, run the file named xsetup.exe in the Windows 7 compatibility mode. During installation, select ISE Webpack from the list of products to be installed. In the installation options, you can also uncheck *Install WinCap for Ethernet Hardware Co-simulation* and *Install Cable Drivers* if you are not going to use the Xilinx JTAG programmer. The Numato Elbert V2 evaluation board will be used for the classes, for which the manufacturer provides both the drivers and the FPGA configuration program.

At the end of the installation process, the license manager program will be launched and you will need to obtain the license file after logging into your Xilinx account.



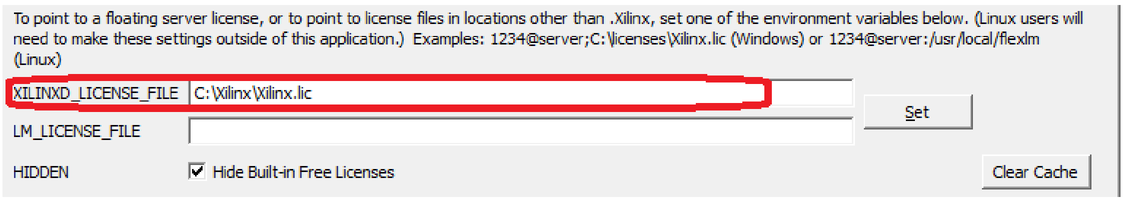
Select one of the following options

- [ ] Start Now! - 30 Day Trial (No Bitstream)
- [✓] Get Free Vivado/ISE WebPack License
- [ ] Start 30 Day Evaluation
- [ ] Get My Purchased License(s)
- [ ] Locate Existing License(s)

Description of the above selected option

Get a free Vivado/ISE WebPack license and start using your Xilinx software. You will be taken to the Xilinx website where you can generate a license for Vivado/ISE WebPack. Once your license file is generated, the "Manage Xilinx Licenses" tab will open to enable you to configure your system to use the license. For more information on Vivado/ISE WebPack, including supported devices and applications, please visit www.xilinx.com.
Note: WebTalk is always enabled for WebPACK users. WebTalk ignores user and install preference when a bitstream is generated using the WebPACK license. If a design is using a device contained in WebPACK and a WebPACK license is available, the WebPACK license will always be used. To change this, please see Answer Record 34746.

The license file called Xilinx.lic should be downloaded from your account and uploaded to the folder where the software was installed. You can also manually set the path to the license file in the license manager.


To point to a floating server license, or to point to license files in locations other than .Xilinx, set one of the environment variables below. (Linux users will need to make these settings outside of this application.) Examples: 1234@server;C:\licenses\Xilinx.lic (Windows) or 1234@server:/usr/local/flexlm (Linux)

XILINXD_LICENSE_FILE	C:\Xilinx\Xilinx.lic
LM_LICENSE_FILE	
HIDDEN	[✓] Hide Built-in Free Licenses

In case of any problems with obtaining a license, the teacher should be informed about this fact. After the installation is complete, apply the fixes described in the note Xilinx AR# 62380:

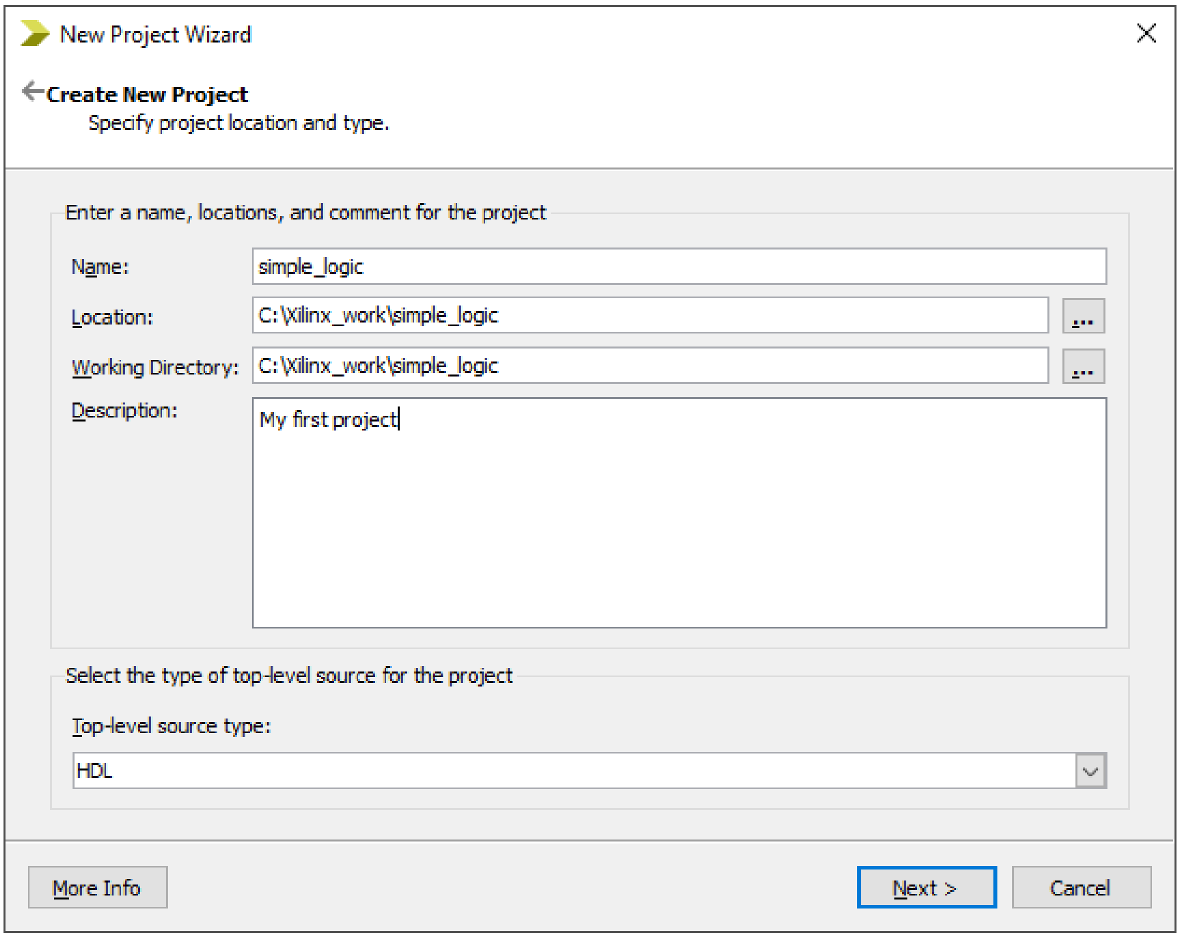
1) Go to *<install_path>\Xilinx\14.7\ISE_DS\ISE\lib\nt64\*
2) Rename the *libPortability.dll* file to *libPortability.dll.orig*
3) Rename the *libPortabilityNOSH.dll* file to *libPortability.dll*

Repeat steps 1 to 3 for the same files in the folder: <install_path>\Xilinx\14.7\ISE_DS\common\lib\nt64\

II. Creating a new project in the ISE WebPack program

Start ISE Design Suite 14.7 from the desktop shortcut or from the Start menu. After starting, the main program window will appear. A new project can be created using the New Project… button in the start window or by selecting File → New Project…

Before creating a project, you should create a working folder, e.g. *Xilinx_work*, where our projects will be saved. In the project creation window, enter the name of the project, e.g. *simple_logic*, and select the previously created working folder, where it will be saved.



Enter a name, locations, and comment for the project

Name: simple_logic

Location: C:\Xilinx_work\simple_logic

Working Directory: C:\Xilinx_work\simple_logic

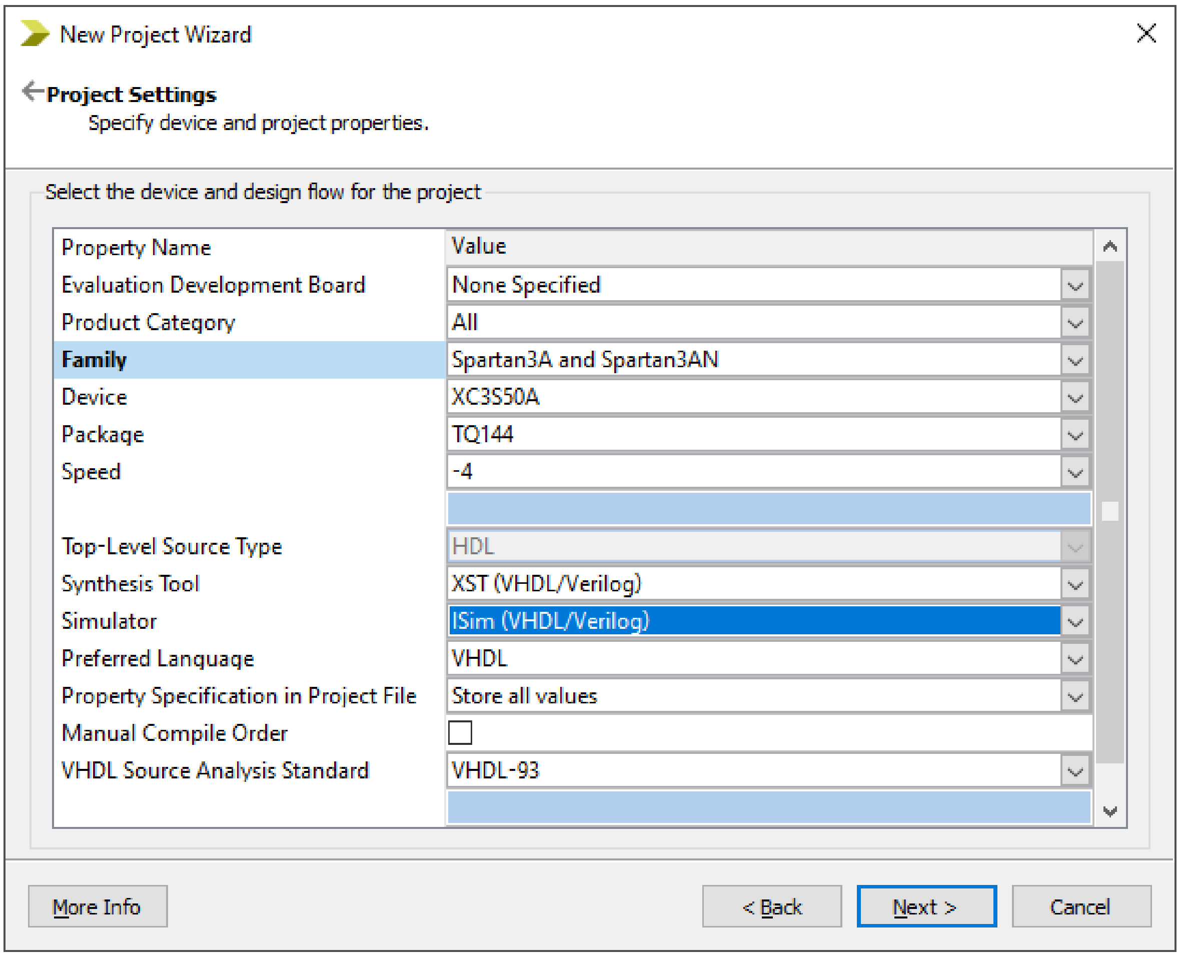
Description: My first project

Select the type of top-level source for the project

Top-level source type: HDL

We also need to set the source type which is the highest in the project hierarchy → HDL.

In the next project settings window, enter the parameters of the target system and the type of hardware description language. We choose parameters based on the markings of the FPGA chip located on the Numato Elbert V2 board. They are presented in the next picture. We choose VHDL (version VHDL-93) as the preferred language.

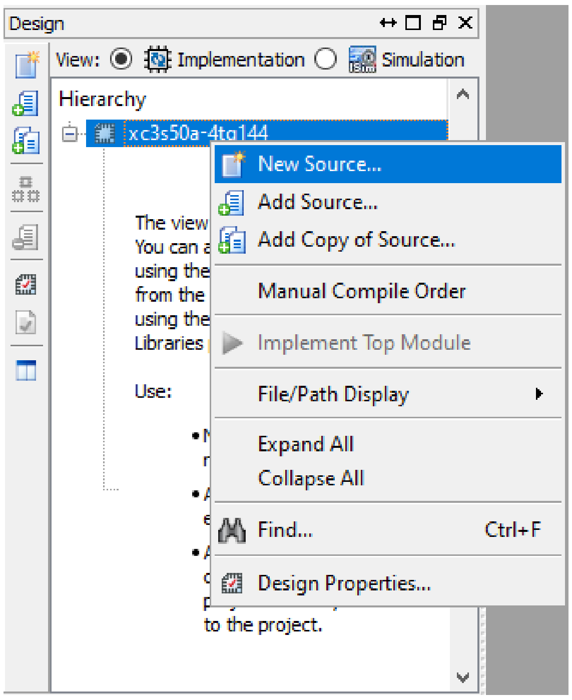

New Project Wizard

Project Settings
Specify device and project properties.

Select the device and design flow for the project

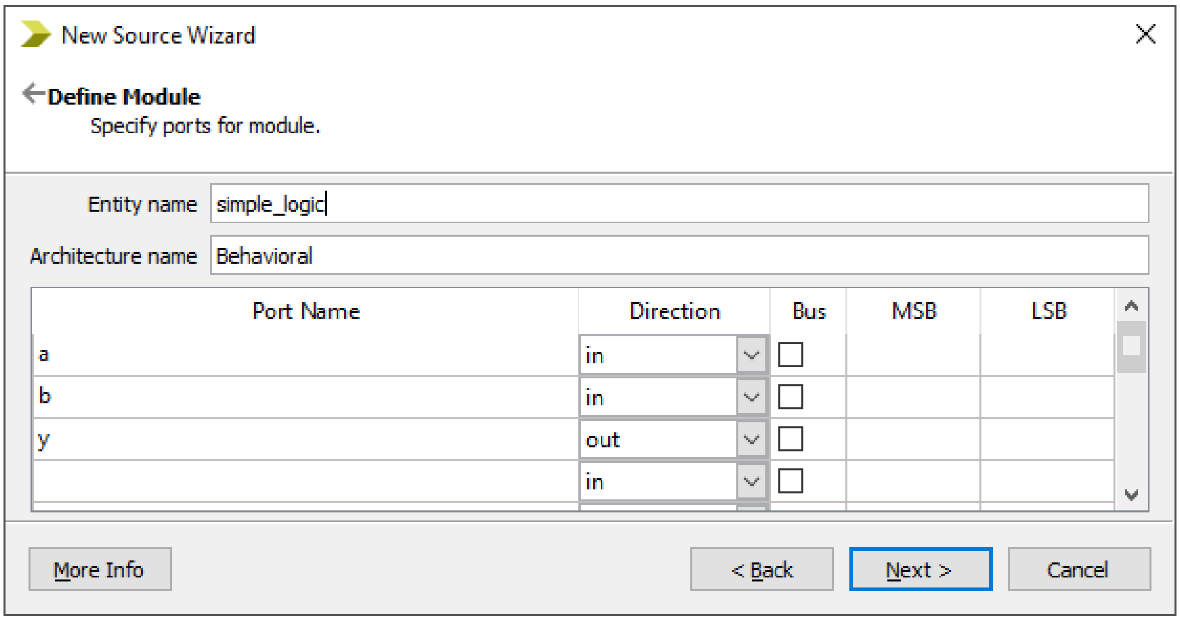
Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3A and Spartan3AN
Device	XC3S50A
Package	TQ144
Speed	-4
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	☐
VHDL Source Analysis Standard	VHDL-93

After creating the project, we add the appropriate sources to it.



First, we add a *VHDL Module* file named *simple_logic*.

In the next window that appears, set up 2 input ports named **a** and **b** and one output port named **y**.


New Source Wizard

Define Module
Specify ports for module.

Entity name: simple_logic
Architecture name: Behavioral

Port Name	Direction	Bus	MSB	LSB
a	in	☐		
b	in	☐		
y	out	☐		
	in	☐		

After adding the file to the project, an editing window will open. Note that the source file has been assigned the **.vhd* extension. We will now complete the description of the architecture so that the project can perform a simple 2-bit logical operation.

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
-- library UNISIM;
-- use UNISIM.VComponents.all;

entity simple_logic is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           y : out  STD_LOGIC);
end simple_logic;

architecture Behavioral of simple_logic is

begin

y <= a and b;       -- funkcja logiczna AND

end Behavioral;
```

You have to remember that the signals are assigned using the operator "<="

In the next step, create a source file of the *Implementation Constraints File* type, which will contain the assignment of ports from the design unit to the appropriate pins of the FPGA chip. We also enter *simple_logic* as the file name. The file will be automatically assigned the **.ucf* extension.

For Numato Elbert V2 kit, all pins are described in the elbertv2.ucf file, available at:

https://productdata.numato.com/assets/downloads/fpga/elbertv2/elbertv2.ucf

We only copy 3 lines from this file to our UCF file:

```
NET "DPSwitch[0]" LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
NET "DPSwitch[1]" LOC = P69 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
NET "LED[0]"      LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
```

We modify the names of the signals to the following:

```
NET "a"  LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
NET "b"  LOC = P69 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
NET "y"  LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
```

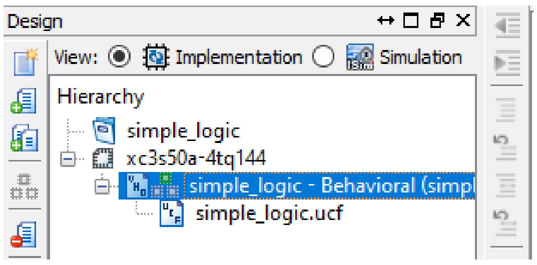
The above configuration connects ports a, b and y with the corresponding pins P70, P69 and P46. P70 and P69 pins have internal pull-up (PULLUP) enabled, voltage standard set as LVCMOS33 (IOSTANDARD parameter), maximum current at 12mA (DRIVE parameter) and slowly rising and falling edges of the signal (SLEW parameter).

Pin P46 has the LVCMOS33 voltage standard set, free slopes of the signal and a maximum current of 12mA.

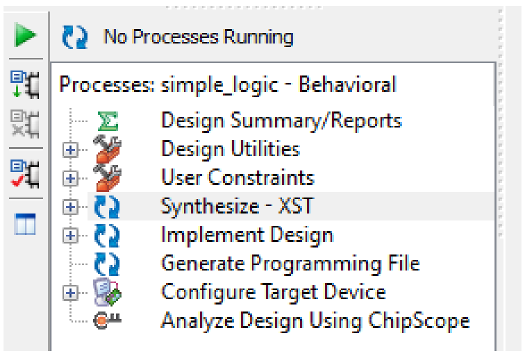
After saving all the files in the project, we move on to the next stage – logical synthesis.

III. Logic synthesis.

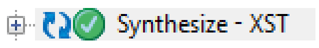
After selecting the source file simple_logic.vhd in the design view window:



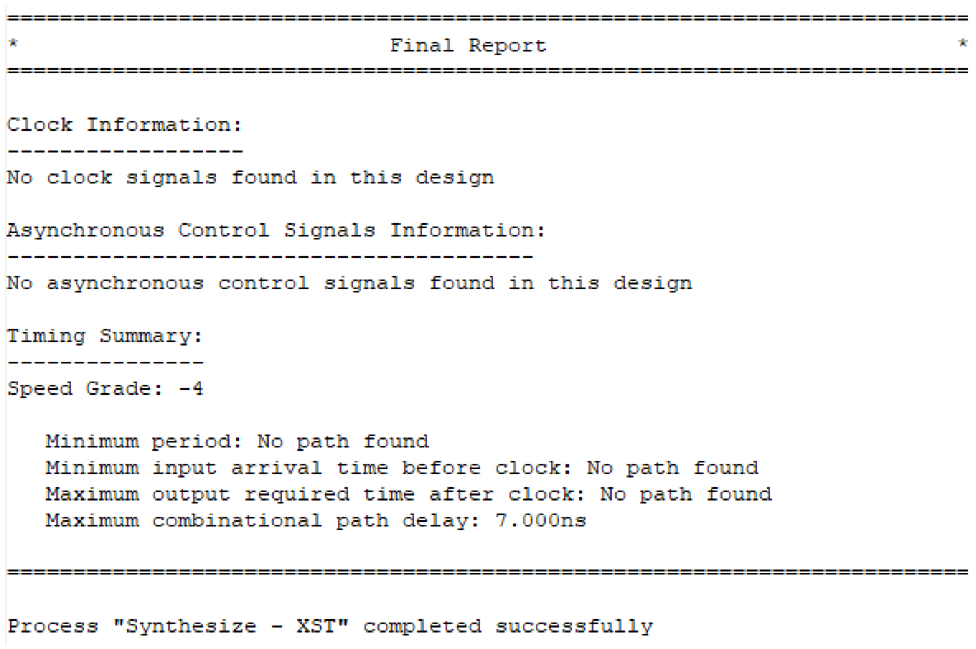
in the lower process window, there will be options that are responsible for the consecutive stages of the project compilation.



First, we run the *Synthesize - XST* option, which will run the syntax check and the logical synthesis process.

If the syntax of the VHDL code does not contain errors and the logical synthesis was successful, the sign of the correct completion of this stage will appear in the process window: 

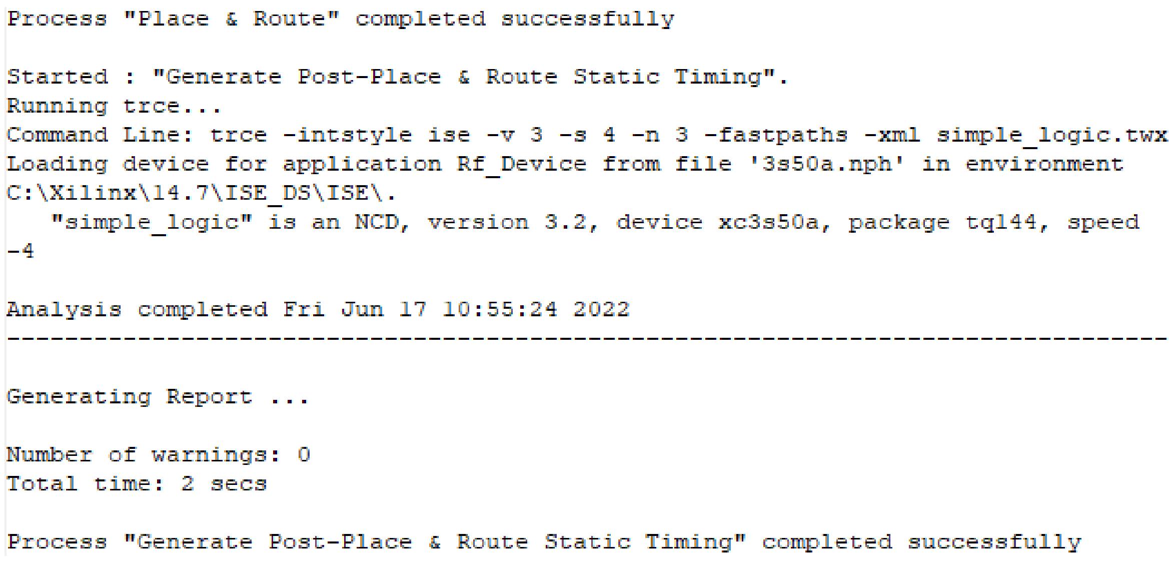
while in the console window a report will be generated on the successful completion of the synthesis process.



```
=========================================================================
*                            Final Report                               *
=========================================================================

Clock Information:
------------------
No clock signals found in this design

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 7.000ns

=========================================================================

Process "Synthesize - XST" completed successfully
```

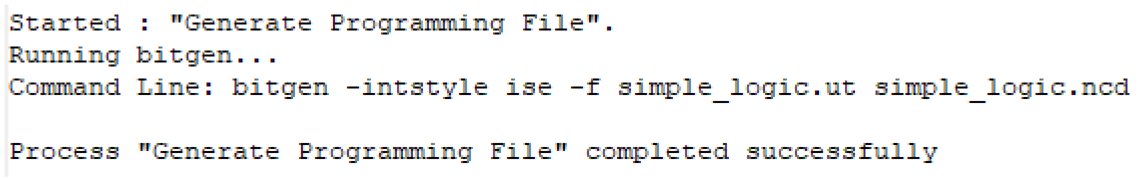
IV. Implementation of the project in the target chip. Chip configuration using the bitstream file.

The next stage of project compilation is starting the implementation process in the target system. The *Implement Design* option in the process window is used for this. After the successful implementation, you will see a sign that this stage has been correctly completed: ✓ Implement Design and in the console window a report on successful completion of the implementation process.



```
Process "Place & Route" completed successfully

Started : "Generate Post-Place & Route Static Timing".
Running trce...
Command Line: trce -intstyle ise -v 3 -s 4 -n 3 -fastpaths -xml simple_logic.twx
Loading device for application Rf_Device from file '3s50a.nph' in environment
C:\Xilinx\14.7\ISE_DS\ISE\.
   "simple_logic" is an NCD, version 3.2, device xc3s50a, package tq144, speed
-4

Analysis completed Fri Jun 17 10:55:24 2022
--------------------------------------------------------------------------------

Generating Report ...

Number of warnings: 0
Total time: 2 secs

Process "Generate Post-Place & Route Static Timing" completed successfully
```

The last step before programming the target system is generating the configuration file by running the *Generate Programming File* option. If the file is generated correctly, the report shown in the picture below will appear in the console window.



```
Started : "Generate Programming File".
Running bitgen...
Command Line: bitgen -intstyle ise -f simple_logic.ut simple_logic.ncd

Process "Generate Programming File" completed successfully
```

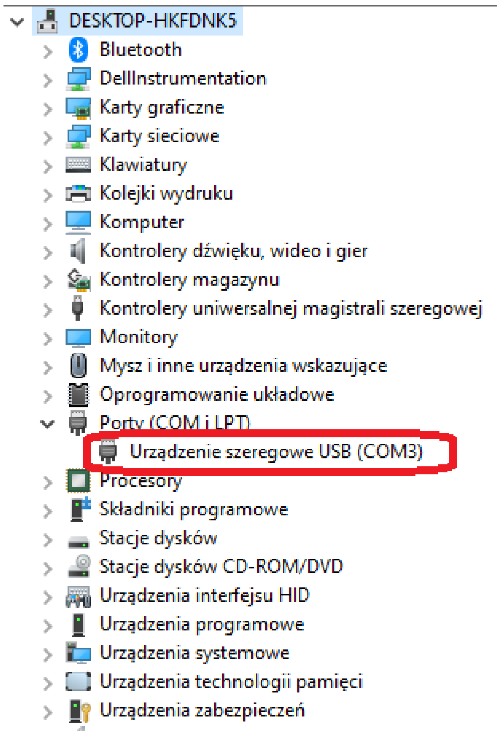
Before programming the target device, download the drivers from the Numato Lab website: https://numato.com/wp-content/uploads/2021/06/numatocdcdriver.zip

Drivers should be unpacked to any folder and select them if the system does not find the appropriate drivers.

To program the target chip (FPGA configuration) you need to use the program provided by the manufacturer:

https://productdata.numato.com/assets/downloads/fpga/elbertv2/ElbertV2Config.exe

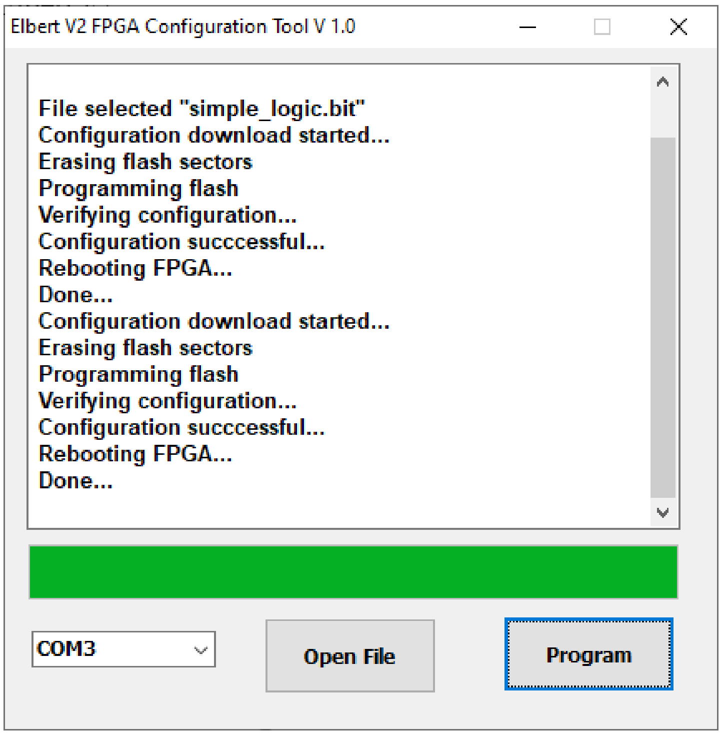
After starting the program, indicate the number of the COM port in the system assigned to the Elbert V2 set.



```
DESKTOP-HKFDNK5
  Bluetooth
  DellInstrumentation
  Karty graficzne
  Karty sieciowe
  Klawiatury
  Kolejki wydruku
  Komputer
  Kontrolery dźwięku, wideo i gier
  Kontrolery magazynu
  Kontrolery uniwersalnej magistrali szeregowej
  Monitory
  Mysz i inne urządzenia wskazujące
  Oprogramowanie układowe
  Porty (COM i LPT)
    Urządzenie szeregowe USB (COM3)
  Procesory
  Składniki programowe
  Stacje dysków
  Stacje dysków CD-ROM/DVD
  Urządzenia interfejsu HID
  Urządzenia programowe
  Urządzenia systemowe
  Urządzenia technologii pamięci
  Urządzenia zabezpieczeń
```

Then we open the **.bit* file in the project folder and program the target system.

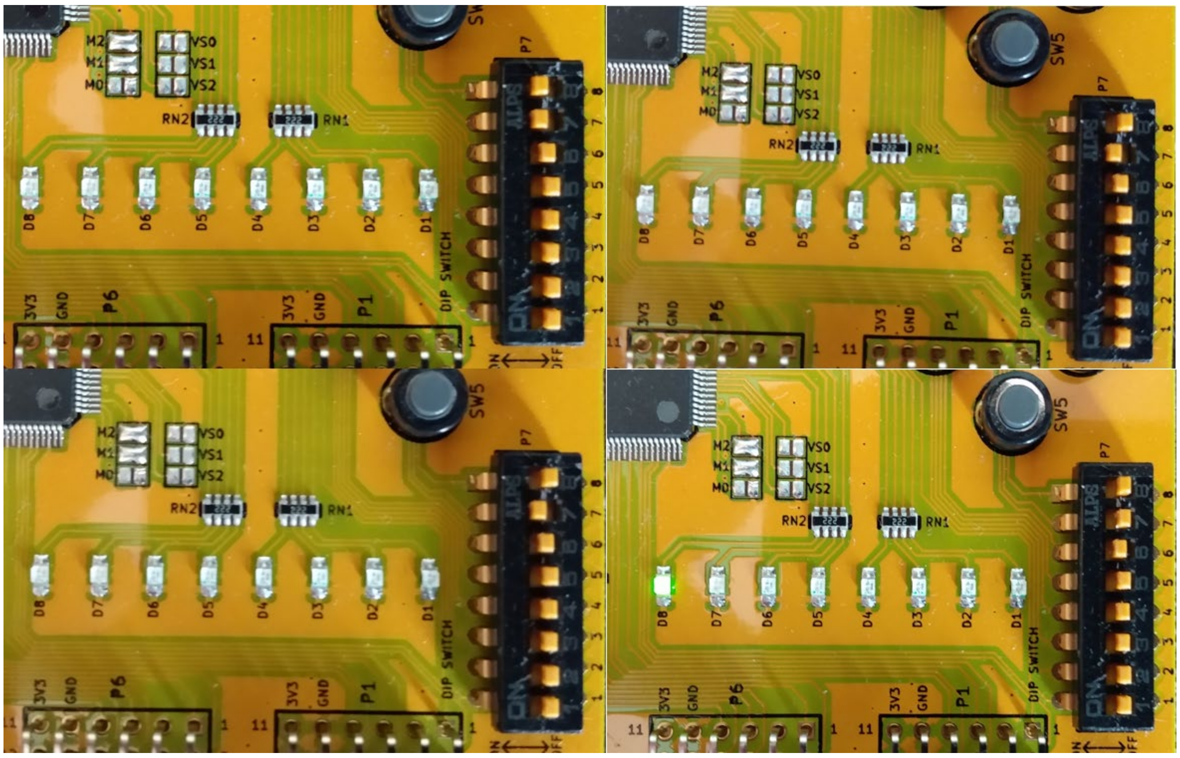
After the correct configuration of the FPGA chip, the D9 diode on the evaluation board (assigned to the DONE pin of the FPGA chip) will light up. The message *Done...* will appear in the program window.



Elbert V2 FPGA Configuration Tool V 1.0

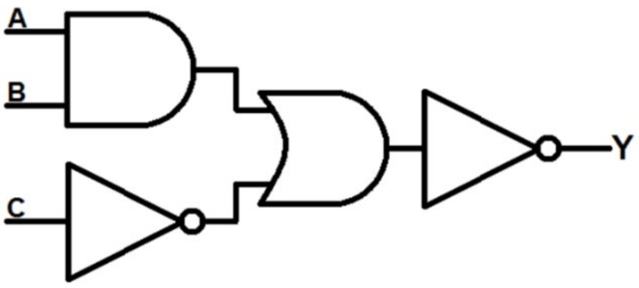
```
File selected "simple_logic.bit"
Configuration download started...
Erasing flash sectors
Programming flash
Verifying configuration...
Configuration succcessful...
Rebooting FPGA...
Done...
Configuration download started...
Erasing flash sectors
Programming flash
Verifying configuration...
Configuration succcessful...
Rebooting FPGA...
Done...
```

The result of the project is presented in the following photos. The change of states on inputs **a** and **b** takes place by means of slide switches no. 7 and 8. The state of output **y** is signaled by the D8 diode.



TASK:

Create a project in VHDL and assign to it the appropriate elements on the Elbert V2 board. The diagram of the combination circuit to be implemented is presented in the figure below.



References

- *User manual for Elbert V2 - Spartan 3A FPGA Development Board.*
  https://numato.com/docs/elbert-v2-spartan-3a-fpga-development-board/

- *62380 - ISE Install - Installing and Running ISE 10.1 or 14.7 on a Windows 8.1 or Windows 10 machine.*
  https://support.xilinx.com/s/article/62380?language=en_US