README.md

# 1. cvičenie

## Preparation tasks

### Table with connection of 16 slide switches and 16 LEDs

| LED | Connection | Switch | Connection |
|---|---|---|---|
| LED0 | H17 | SW0 | J15 |
| LED1 | K15 | SW1 | L16 |
| LED2 | J13 | SW2 | M13 |
| LED3 | N14 | SW3 | R15 |
| LED4 | R18 | SW4 | R17 |
| LED5 | V17 | SW5 | T18 |
| LED6 | U17 | SW6 | U18 |
| LED7 | U16 | SW7 | R13 |
| LED8 | V16 | SW8 | T8 |
| LED9 | T15 | SW9 | U8 |
| LED10 | U14 | SW10 | R16 |
| LED11 | T16 | SW11 | T13 |
| LED12 | V15 | SW12 | H6 |
| LED13 | V14 | SW13 | U12 |
| LED14 | V12 | SW14 | U11 |
| LED15 | V11 | SW15 | V10 |

# 2. cvičenie

# Two-bit wide 4-to-1 multiplexer

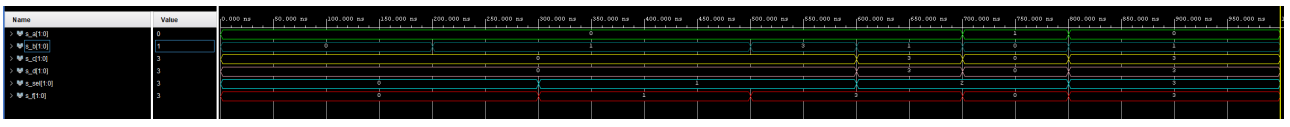
## VHDL architecture ( `mux_2bit_4to1.vhd` )

```
architecture Behavioral of mux_2bit_4to1 is
begin
    f_o <=  a_i when (sel_i = "00") else
            b_i when (sel_i = "01") else
            c_i when (sel_i = "10") else
            d_i;



end architecture Behavioral;
```

## VHDL stimulus process ( `tb_mux_2bit_4to1.vhd` )

```
p_stimulus : process
    begin
        -- Report a note at the begining of stimulus process
        report "Stimulus process started" severity note;



        -- First test values
        s_d <= "00"; s_c <= "00"; s_b <= "00"; s_a <= "00"; s_sel <= "00"; wait for 100 ns;

        s_a <= "00"; wait for 100 ns;
        s_b <= "01"; wait for 100 ns;

        s_sel <= "01"; wait for 100 ns;
        s_c <= "00"; wait for 100 ns;
        s_b <= "11"; wait for 100 ns;

        s_d <= "11";  s_c <= "11"; s_b <= "01"; s_a <= "00";
        s_sel <= "10"; wait for 100 ns;

        s_d <= "00";  s_c <= "00"; s_b <= "00"; s_a <= "01";
        s_sel <= "10"; wait for 100 ns;

        s_d <= "11";  s_c <= "11"; s_b <= "01"; s_a <= "00";
        s_sel <= "11"; wait for 100 ns;

        -- WRITE OTHER TESTS HERE




        -- Report a note at the end of stimulus process
        report "Stimulus process finished" severity note;
        wait;
    end process p_stimulus;

end architecture testbench;
```

## Screenshot with waveforms



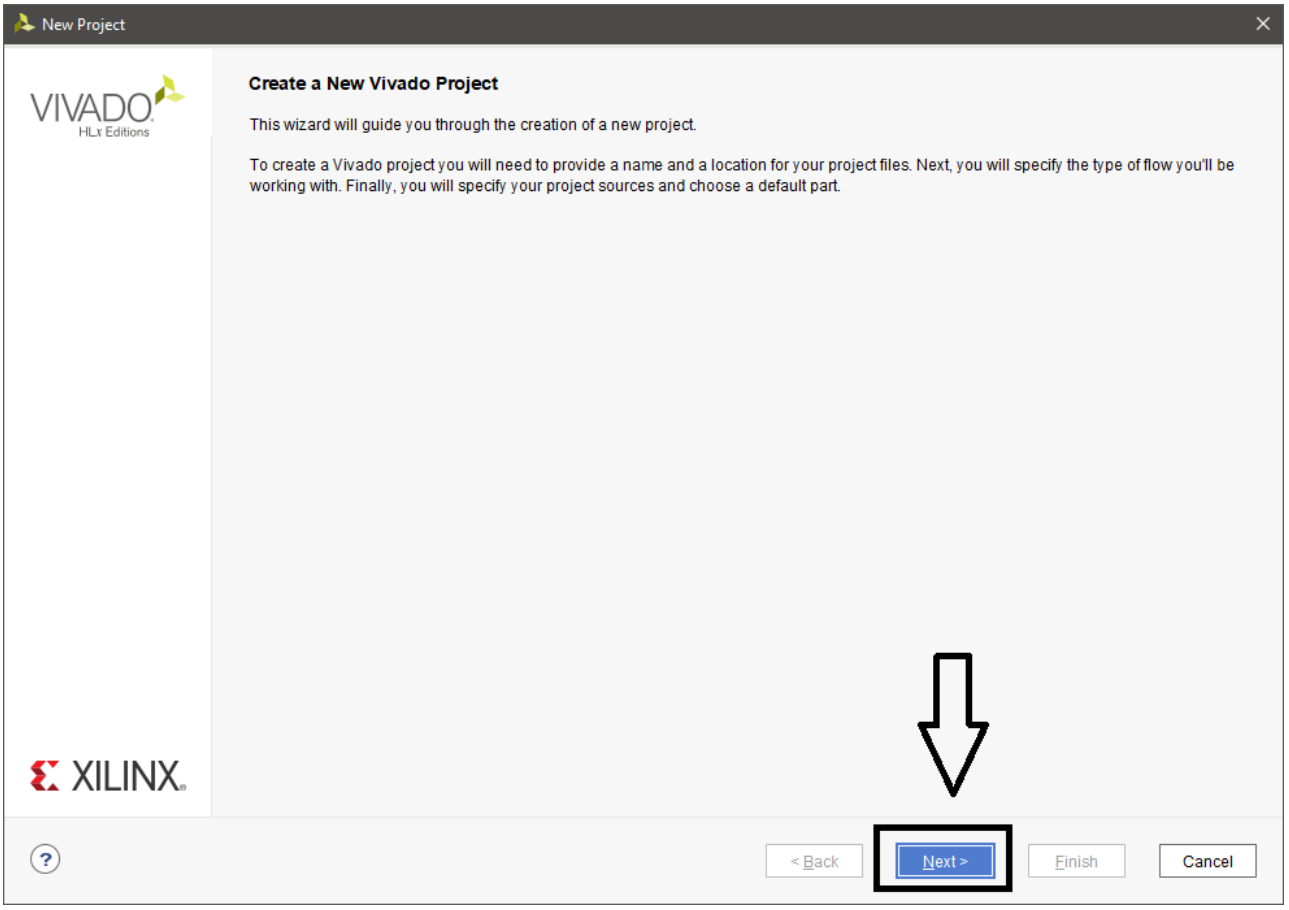
# 3. cvičenie

## A Vivado tutorial

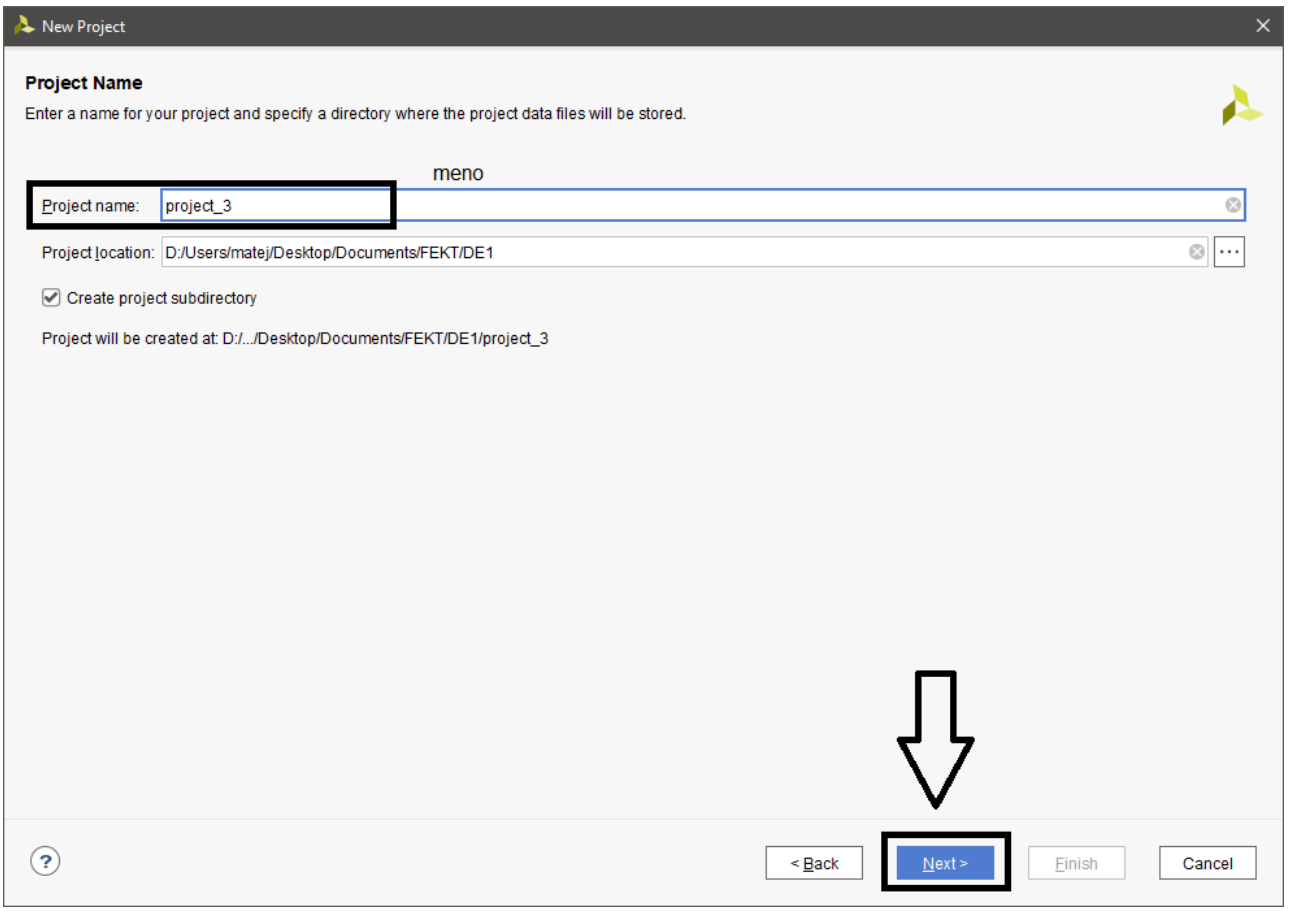
### Tutorial

#### 1. krok

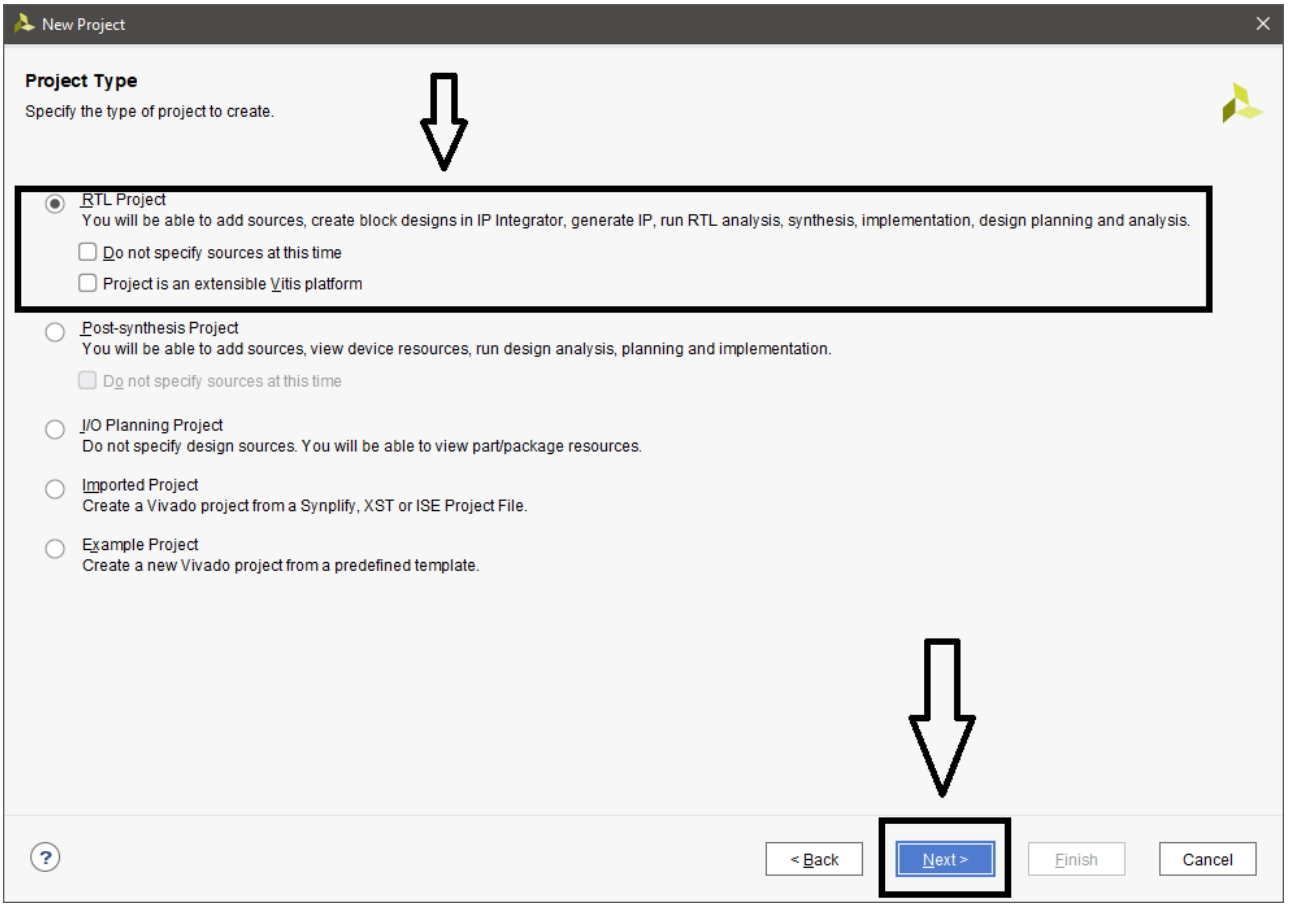

#### 2. krok



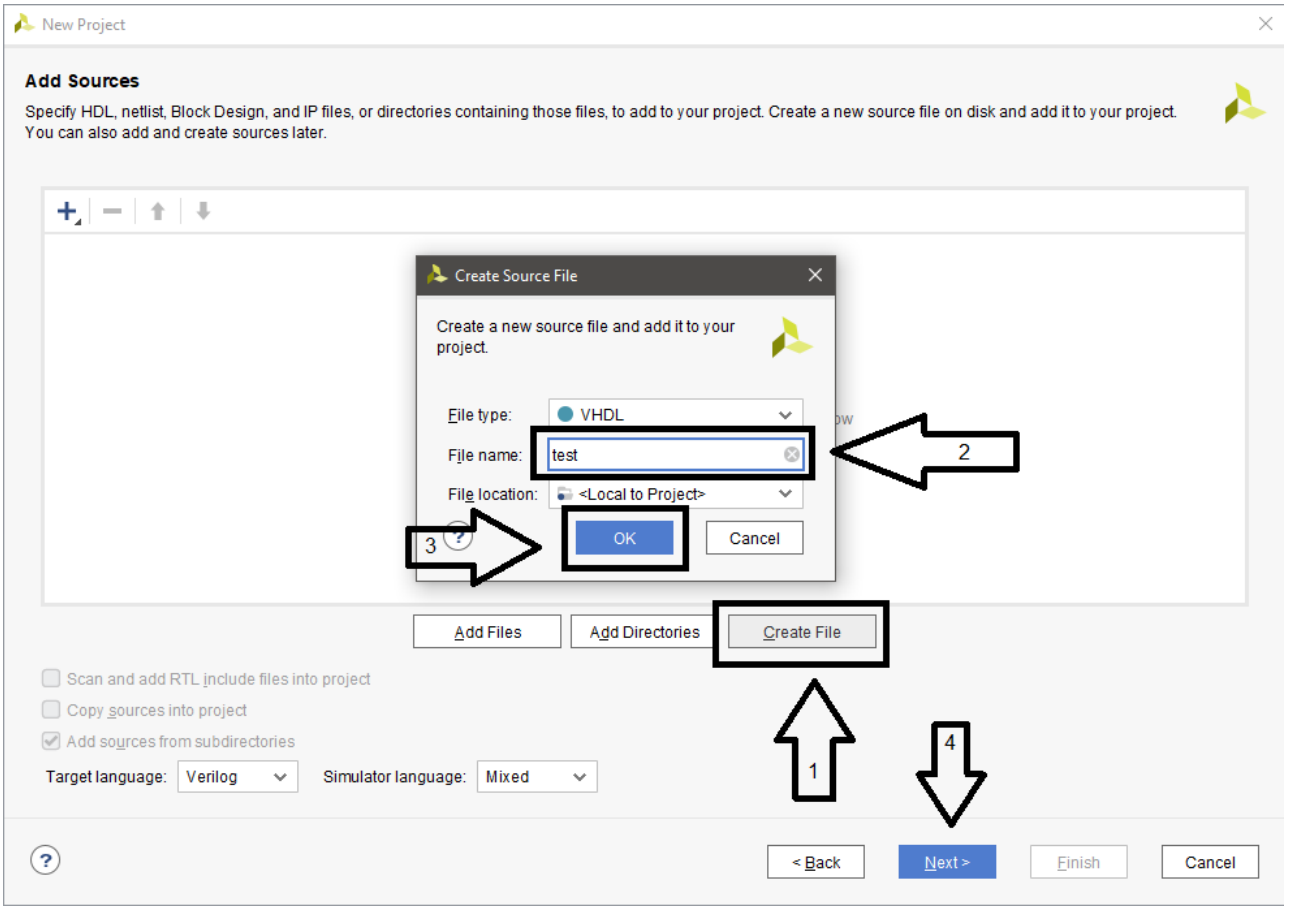

#### 3. krok


New Project
Project Name
Enter a name for your project and specify a directory where the project data files will be stored.
meno
Project name: project_3
Project location: D:/Users/matej/Desktop/Documents/FEKT/DE1
Create project subdirectory
Project will be created at: D:/.../Desktop/Documents/FEKT/DE1/project_3
< Back
Next >
Finish
Cancel

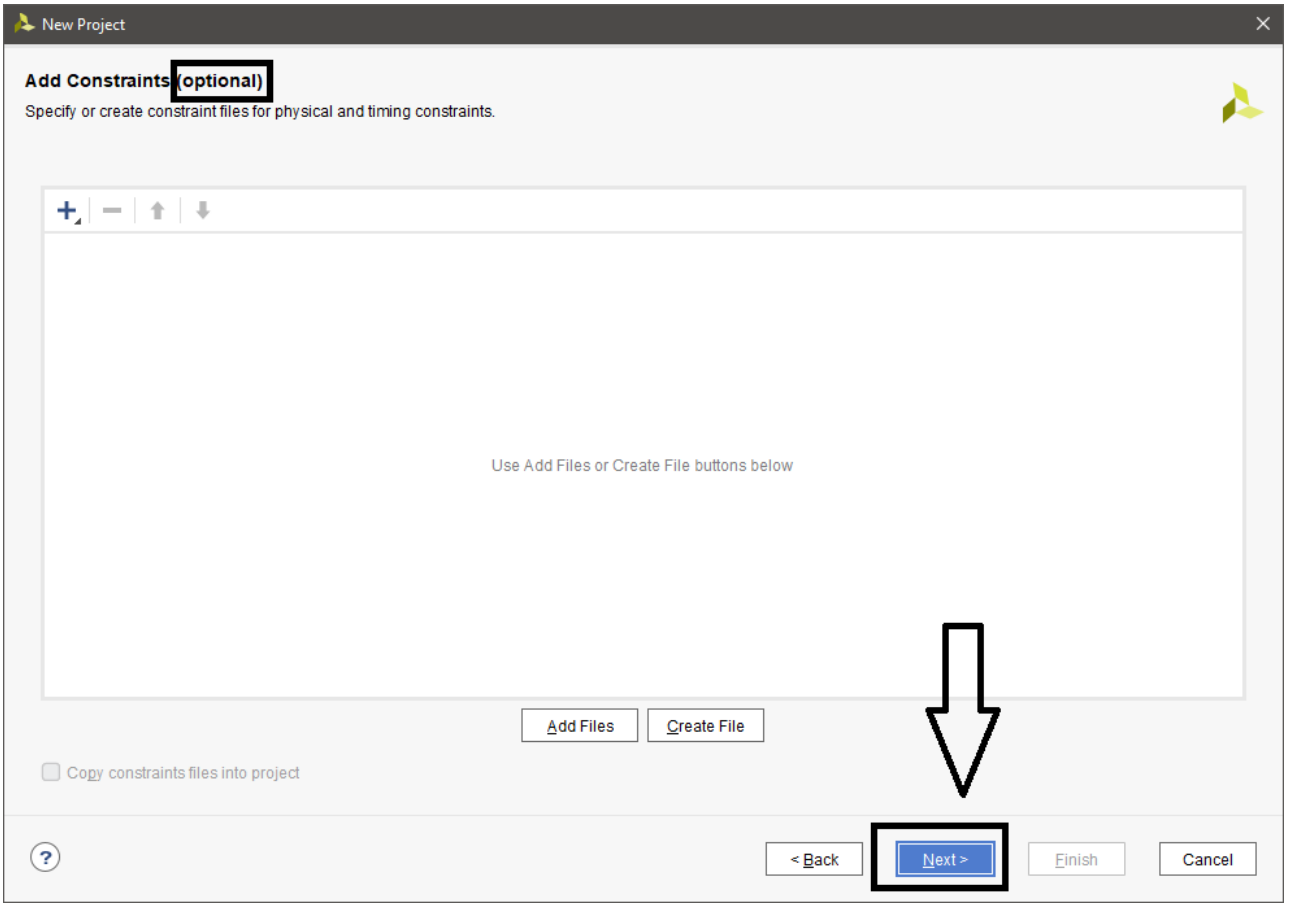
4. krok


New Project
Project Type
Specify the type of project to create.
RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
Do not specify sources at this time
Project is an extensible Vitis platform
Post-synthesis Project
You will be able to add sources, view device resources, run design analysis, planning and implementation.
Do not specify sources at this time
I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.
Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.
Example Project
Create a new Vivado project from a predefined template.
< Back
Next >
Finish
Cancel

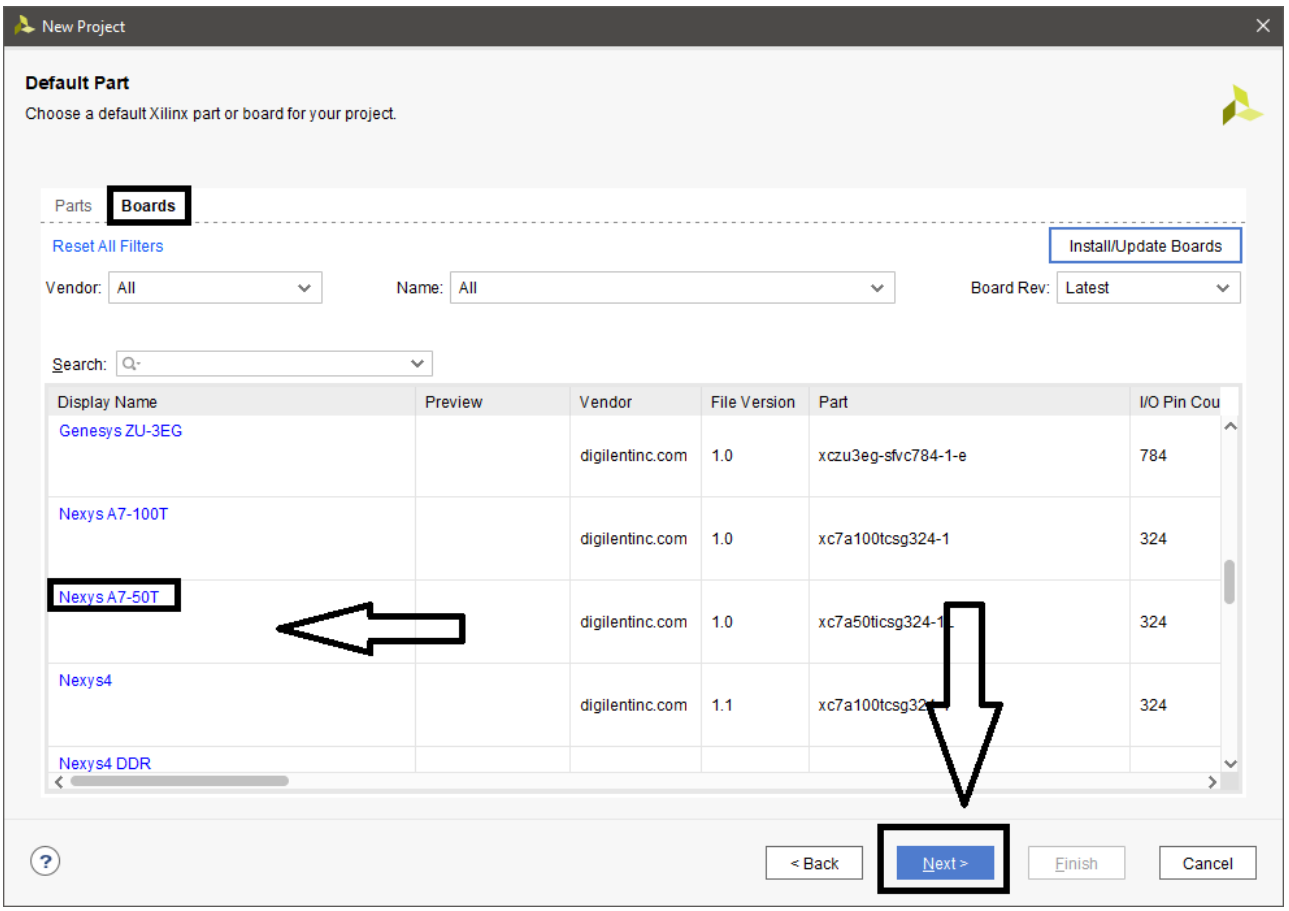
5. krok


New Project
Add Sources
Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project.
You can also add and create sources later.
Create Source File
Create a new source file and add it to your project.
File type: VHDL
File name: test
File location: <Local to Project>
OK
Cancel
2
3
Add Files
Add Directories
Create File
1
4
Scan and add RTL include files into project
Copy sources into project
Add sources from subdirectories
Target language: Verilog
Simulator language: Mixed
< Back
Next >
Finish
Cancel

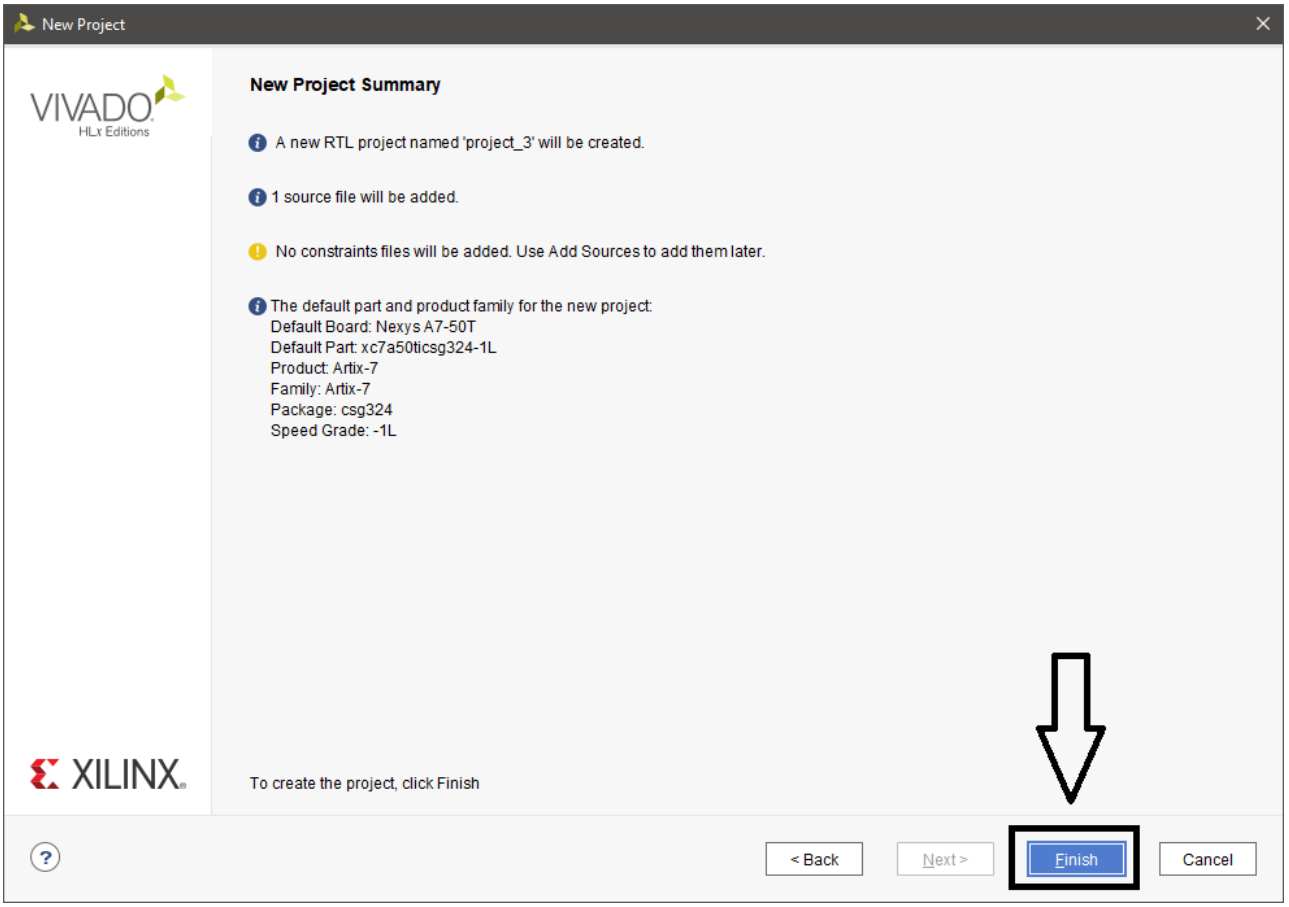
6. krok


New Project
Add Constraints (optional)
Specify or create constraint files for physical and timing constraints.
Use Add Files or Create File buttons below
Add Files
Create File
Copy constraints files into project
< Back
Next >
Finish
Cancel

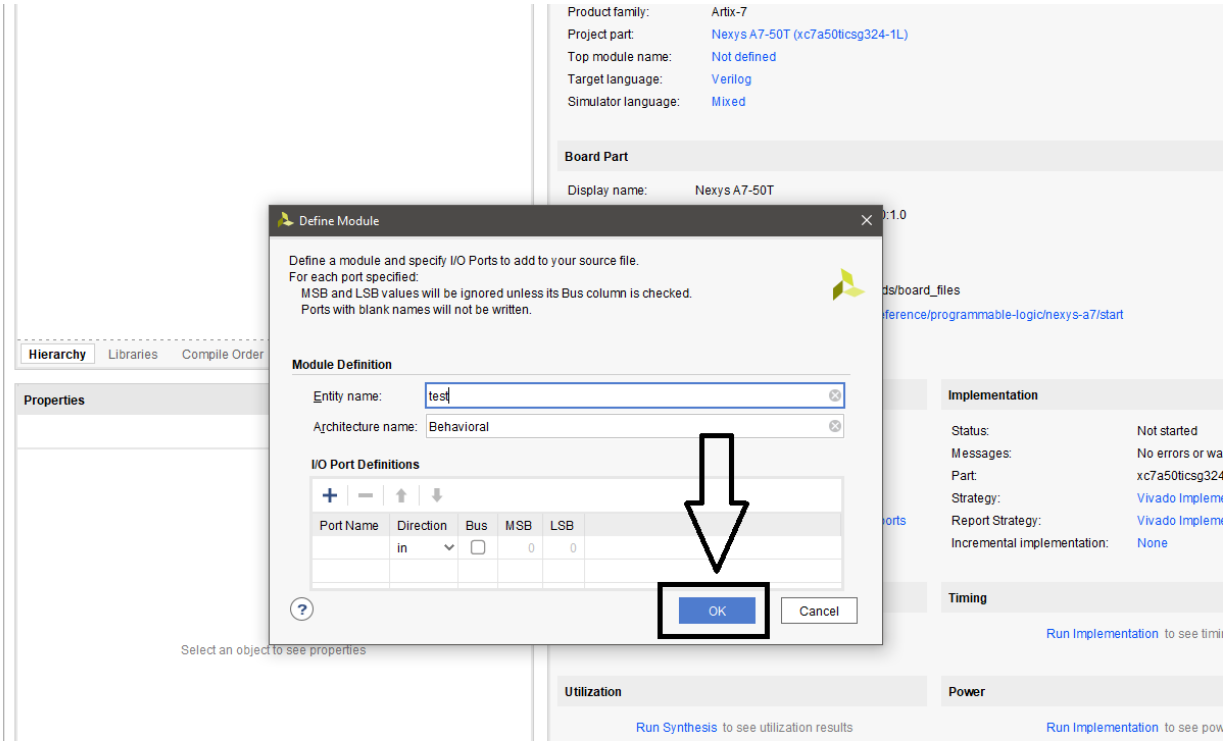
7. krok


New Project
Default Part
Choose a default Xilinx part or board for your project.
Parts
Boards
Reset All Filters
Install/Update Boards
Vendor: All
Name: All
Board Rev: Latest
Search:
Display Name
Preview
Vendor
File Version
Part
I/O Pin Cou
Genesys ZU-3EG
digilentinc.com
1.0
xczu3eg-sfvc784-1-e
784
Nexys A7-100T
digilentinc.com
1.0
xc7a100tcsg324-1
324
Nexys A7-50T
digilentinc.com
1.0
xc7a50ticsg324-1
324
Nexys4
digilentinc.com
1.1
xc7a100tcsg32
324
Nexys4 DDR
< Back
Next >
Finish
Cancel

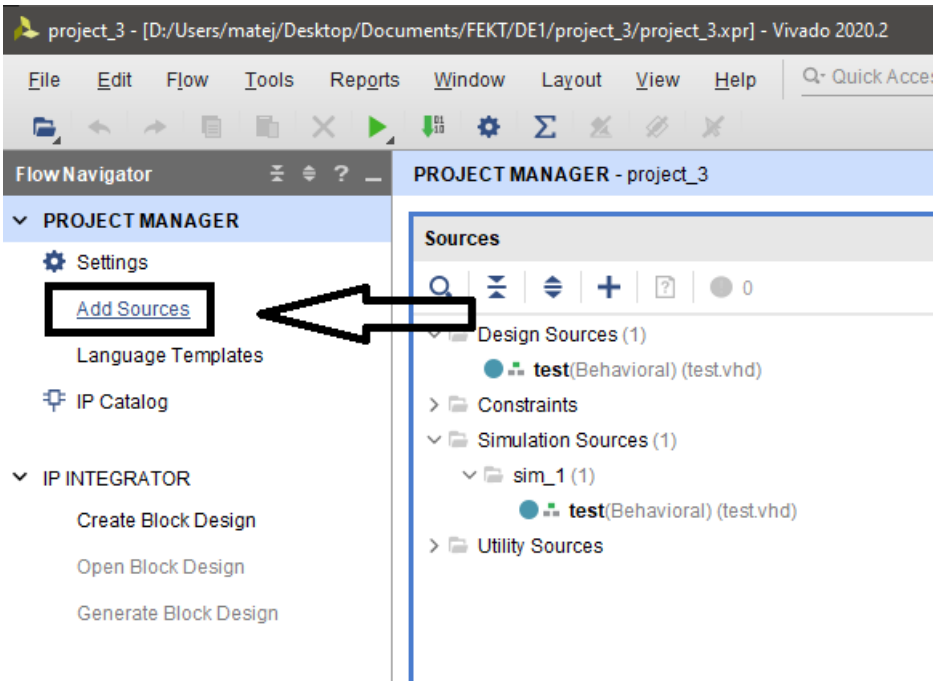
8. krok


New Project
VIVADO
HLx Editions
New Project Summary
A new RTL project named 'project_3' will be created.
1 source file will be added.
No constraints files will be added. Use Add Sources to add them later.
The default part and product family for the new project:
Default Board: Nexys A7-50T
Default Part: xc7a50ticsg324-1L
Product: Artix-7
Family: Artix-7
Package: csg324
Speed Grade: -1L
XILINX
To create the project, click Finish
< Back
Next >
Finish
Cancel

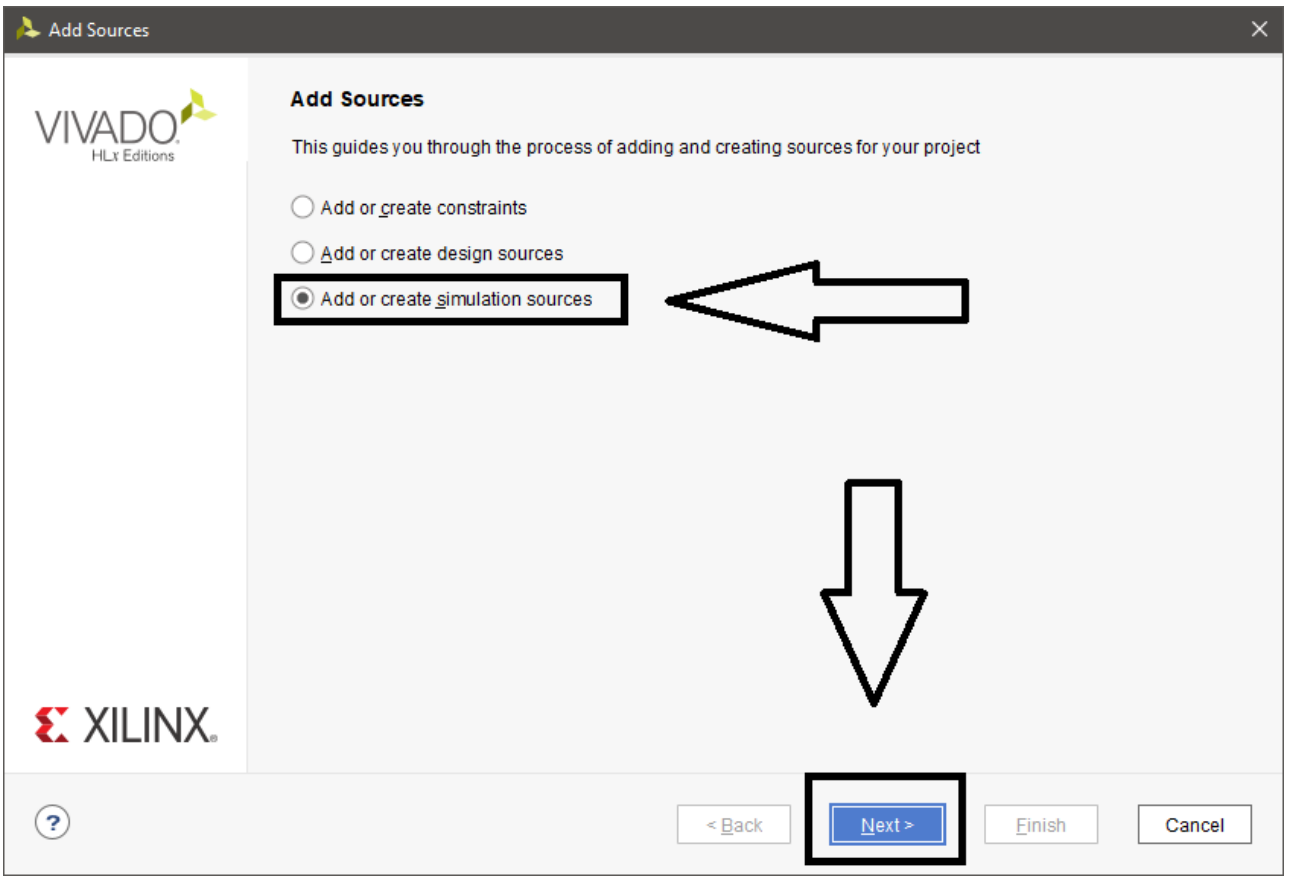
9. krok



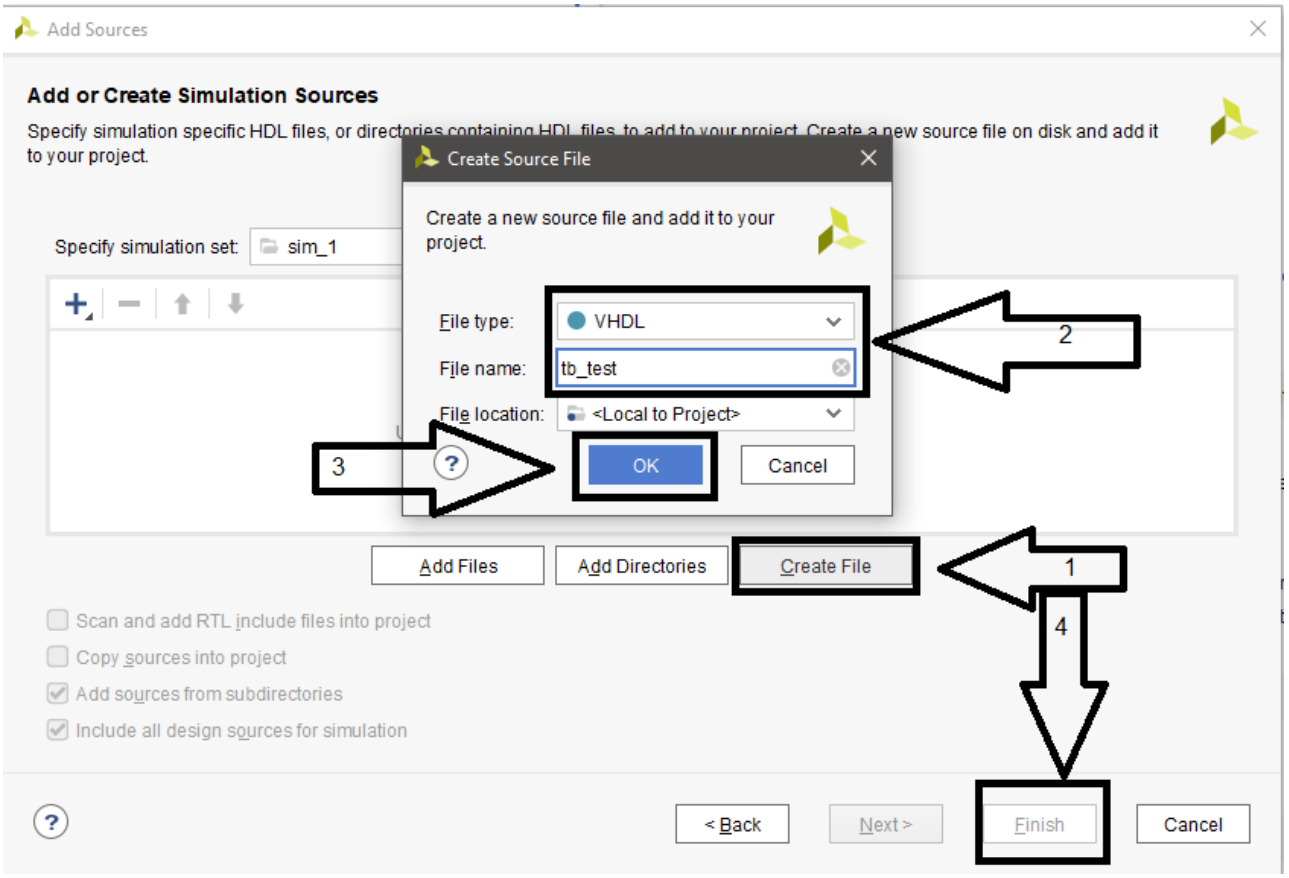

10. krok



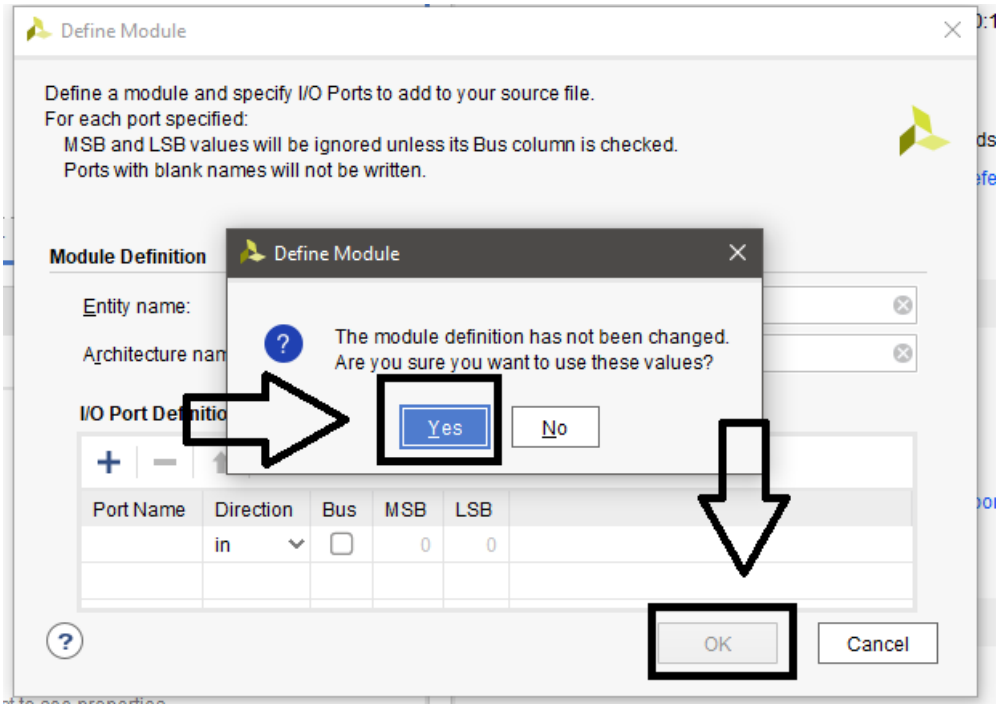

11. krok



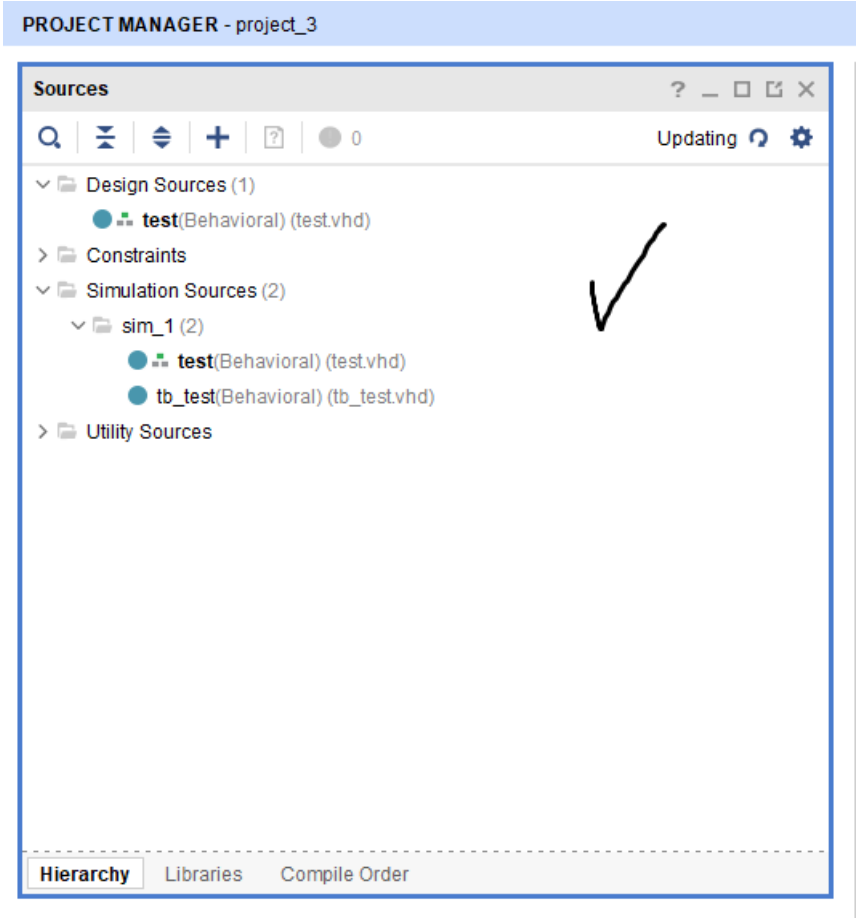

12. krok




13. krok


Define Module
Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.
Module Definition
Entity name:
Architecture nam
I/O Port Definitio
Port Name
Direction
Bus
MSB
LSB
in
0
0
OK
Cancel
Define Module
The module definition has not been changed.
Are you sure you want to use these values?
Yes
No

14. krok


PROJECT MANAGER - project_3
Sources
Updating
Design Sources (1)
test(Behavioral) (test.vhd)
Constraints
Simulation Sources (2)
sim_1 (2)
test(Behavioral) (test.vhd)
tb_test(Behavioral) (tb_test.vhd)
Utility Sources
Hierarchy
Libraries
Compile Order