



xkocum00 Update README.md

🕒 History

👤 1 contributor

Raw

Blame



208 lines (159 sloc) | 6.95 KB

Lab 3: Introduction to Vivado

1. Preparation tasks (done before the lab at home). Submit:

Figure or table with connection of 16 slide switches and 16 LEDs on Nexys A7 board.


```
-----  
-- Entity declaration for 2-bit binary comparator  
-----
```

```
entity mux_2bit_4to1 is  
    port(  
        a_i          : in  std_logic_vector(2 - 1 downto 0);  
        b_i          : in  std_logic_vector(2 - 1 downto 0);  
        c_i          : in  std_logic_vector(2 - 1 downto 0);  
        d_i          : in  std_logic_vector(2 - 1 downto 0);  
        sel_i        : in  std_logic_vector(2 - 1 downto 0);  
  
        f_o          : out std_logic_vector(2 - 1 downto 0)  
  
    );  
end entity mux_2bit_4to1;
```

```
-----  
-- Architecture body for 2-bit binary comparator  
-----
```

```
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;
```

testbench

```
-----  
--  
-- Testbench for 2-bit binary comparator.  
-- EDA Playground  
--  
-- Copyright (c) 2020-2021 Tomas Fryza  
-- Dept. of Radio Electronics, Brno University of Technology, Czechia  
-- This work is licensed under the terms of the MIT license.  
--  
-----
```

```
library ieee;  
use ieee.std_logic_1164.all;
```

```
-----  
-- Entity declaration for testbench  
-----
```

```
entity tb_mux_2bit_4to1 is  
    -- Entity of testbench is always empty  
end entity tb_mux_2bit_4to1;
```

```
-----
```

```
-- Architecture body for testbench
```

```
-----  
architecture testbench of tb_mux_2bit_4to1 is
```

```
-- Local signals
```

```
signal s_a      : std_logic_vector(2 - 1 downto 0);  
signal s_b      : std_logic_vector(2 - 1 downto 0);  
signal s_c      : std_logic_vector(2 - 1 downto 0);  
signal s_d      : std_logic_vector(2 - 1 downto 0);  
signal s_sel     : std_logic_vector(2 - 1 downto 0);  
  
signal s_f      : std_logic_vector(2 - 1 downto 0);
```

```
begin
```

```
-- Connecting testbench signals with comparator_4bit entity (Unit Under Test)
```

```
uut_mux_2bit_4to1 : entity work.mux_2bit_4to1
```

```
    port map(  
        a_i      => s_a,  
        b_i      => s_b,  
        c_i      => s_c,  
        d_i      => s_d,  
        sel_i     => s_sel,  
        f_o      => s_f  
    );
```

```
-----  
-- Data generation process  
-----
```

```
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 10 ns;
```

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

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

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

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

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

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

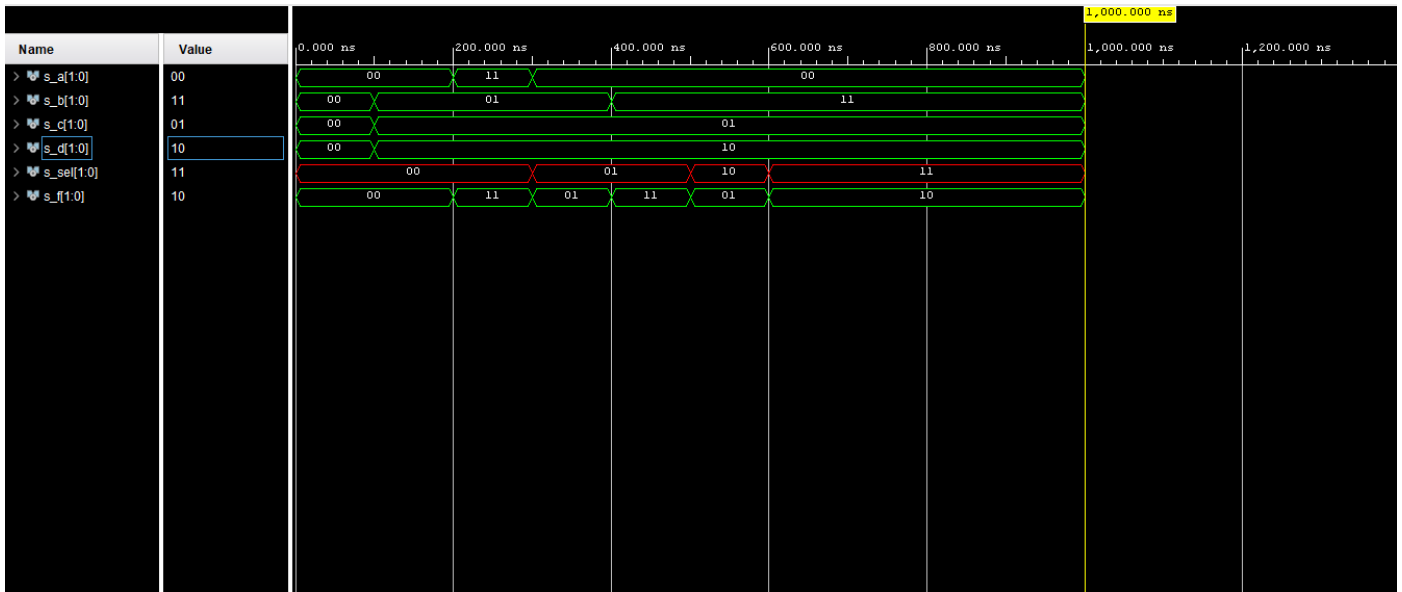
```
-- Report a note at the end of stimulus process
```

```
report "Stimulus process finished" severity note;
```

```
wait;  
end process p_stimulus;
```

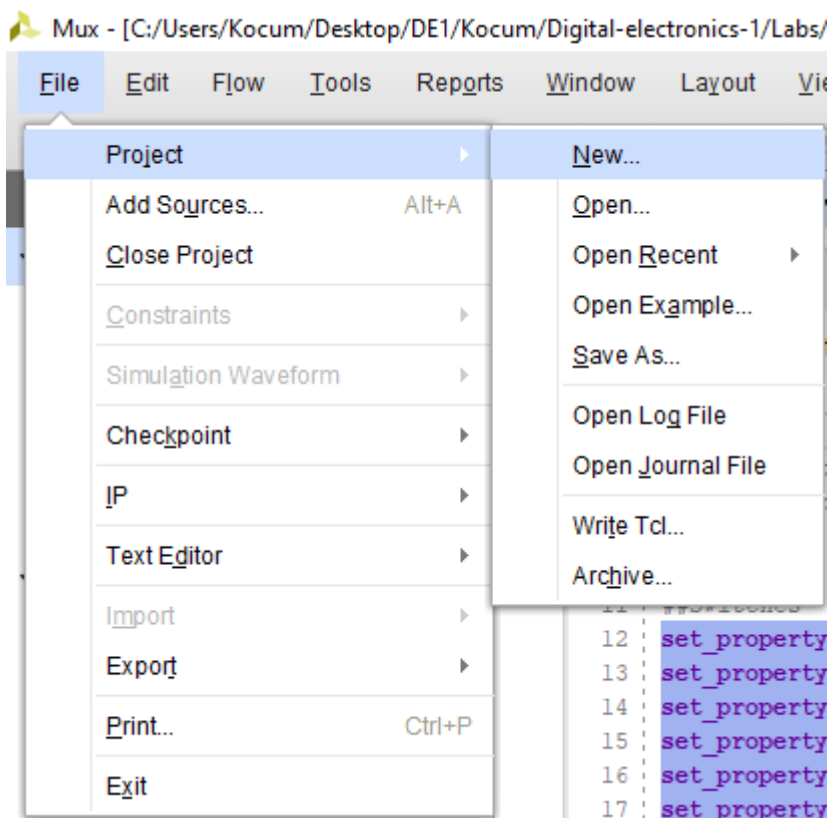
```
end architecture testbench;
```

simulated time waveforms

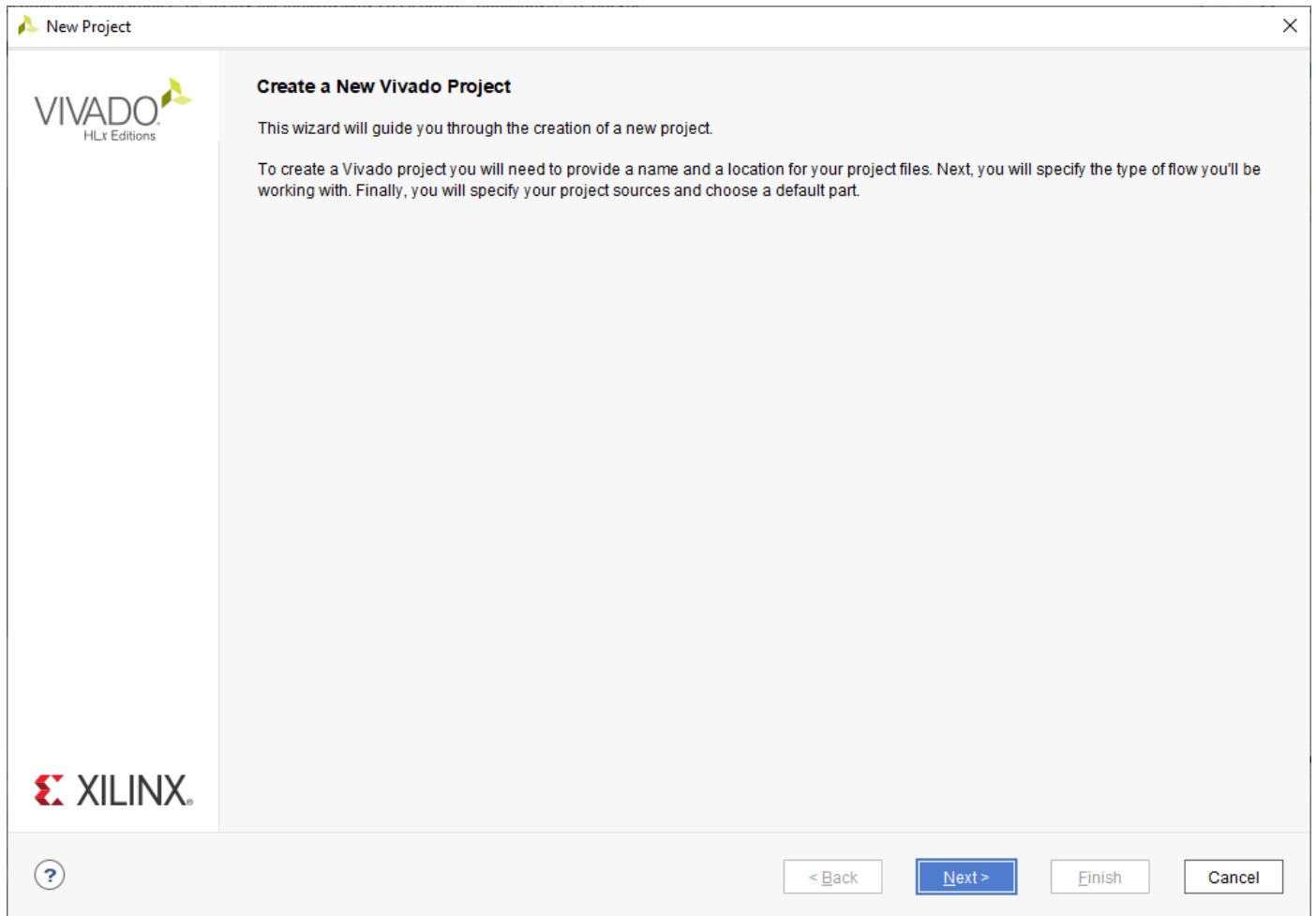


3. A Vivado tutorial. Submit:

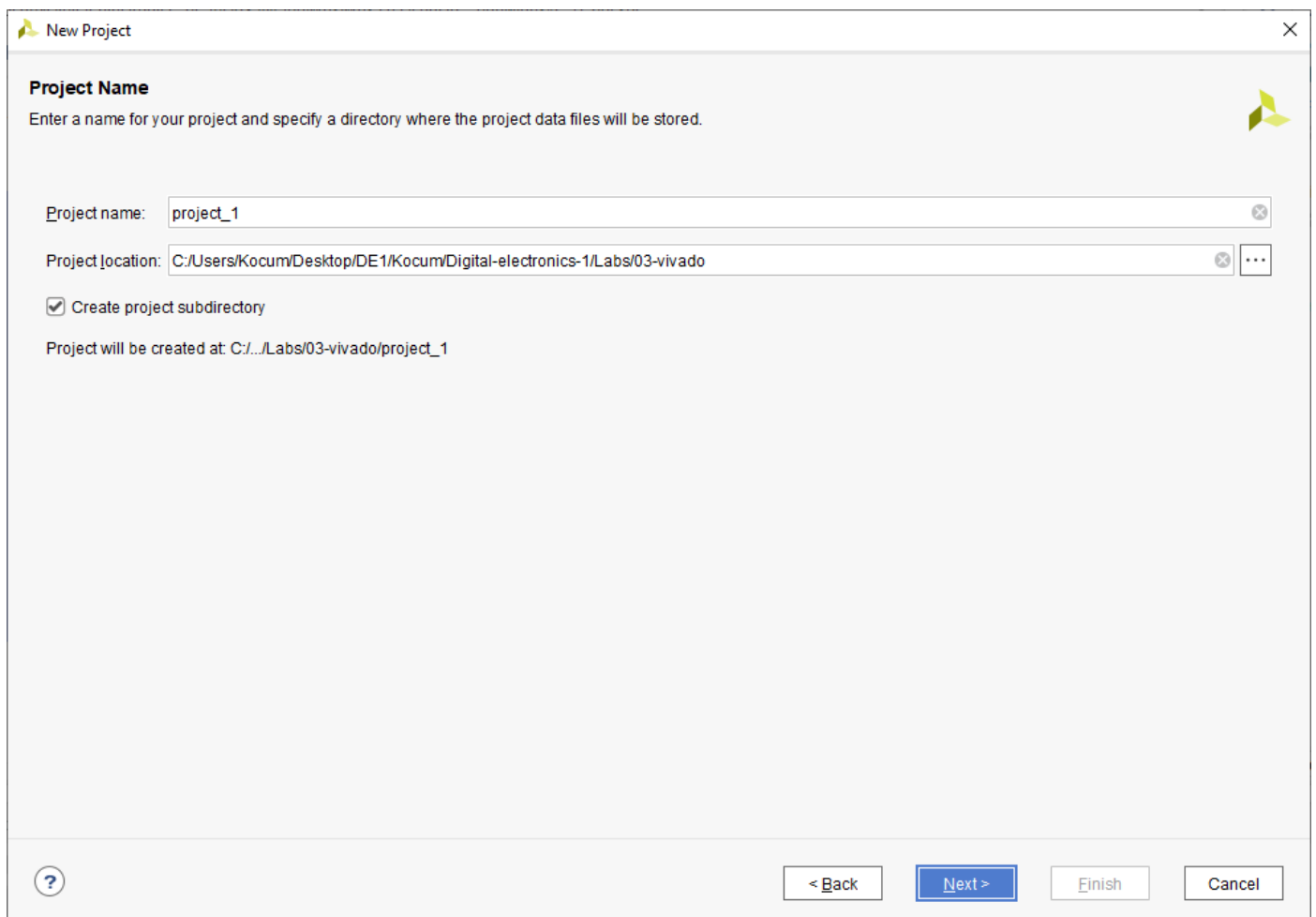
firstly open vivado and click on File -> Project -> New...




click next




give name to your project and select location path



select rtl

 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

create source file

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:

File location:

<Local to Project>

?

OK

Cancel

Add Files

Add Directories

Create File

☐ Scan and add RTL include files into project☐ Copy sources into project☒ Add sources from subdirectories

Target language: VHDL

Simulator language: VHDL



< Back

Next >

Finish

Cancel

next

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

select your board

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: nexys (5 matches)

Display Name	Preview	Vendor	File Version	Part	I/O Pin Count	Board Rev	Available IOBs
Nexys A7-100T		digilentinc.com	1.0	xc7a100tcsg324-1	324	D.0	210
Nexys A7-50T		digilentinc.com	1.0	xc7a50ticsg324-1L	324	D.0	210
Nexys4		digilentinc.com	1.1	xc7a100tcsg324-1	324	B.1	210
Nexys4 DDR		digilentinc.com	1.1	xc7a100tcsg324-1	324	C.1	210
Nexys Video							

?

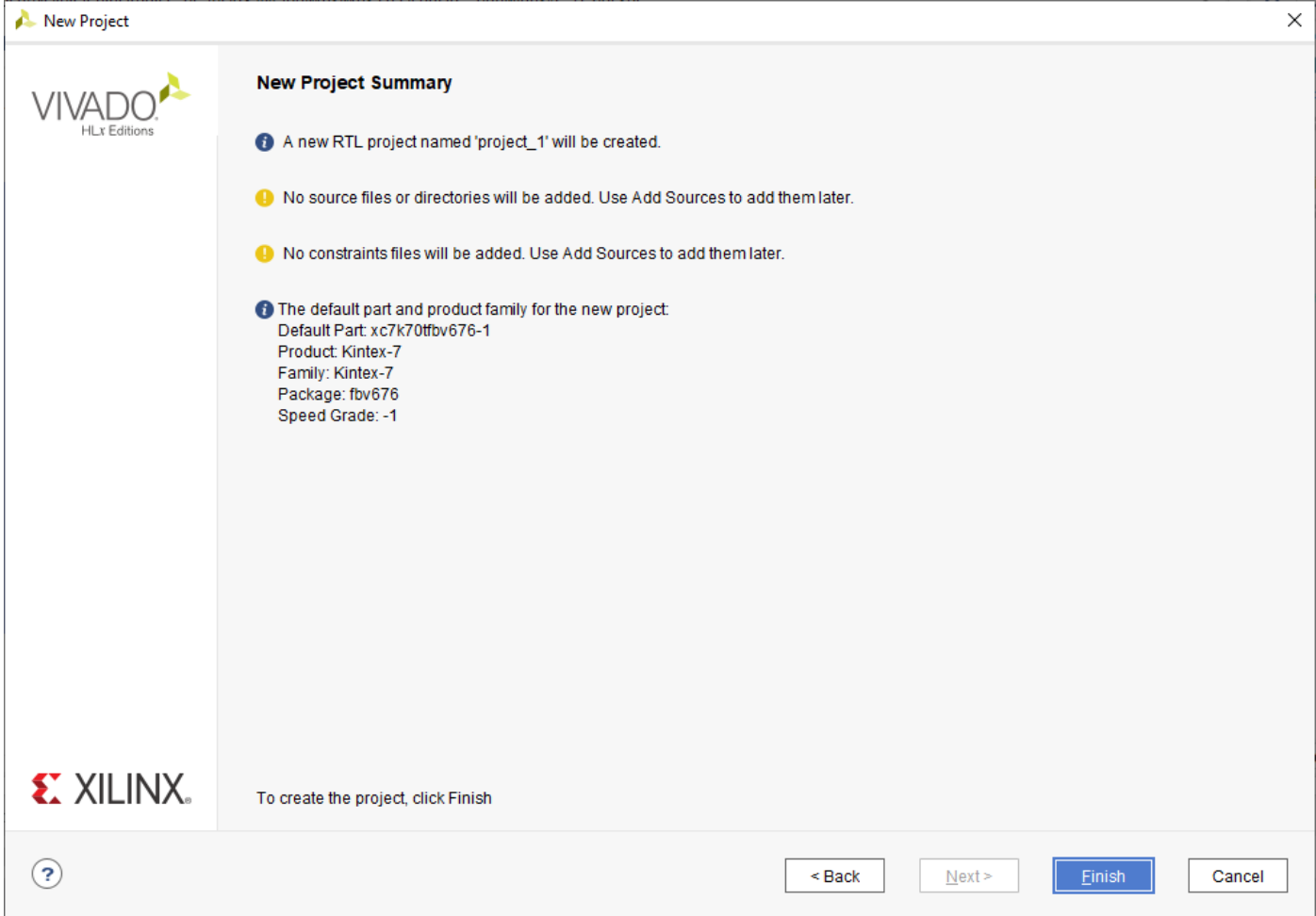
< Back

Next >

Finish

Cancel

finish



xdc file

```

set_property -dict { PACKAGE_PIN J15      IOSTANDARD LVCMOS33 } [get_ports { a_i[0] }]; #IO_L2
set_property -dict { PACKAGE_PIN L16      IOSTANDARD LVCMOS33 } [get_ports { a_i[1] }]; #IO_L3
set_property -dict { PACKAGE_PIN M13      IOSTANDARD LVCMOS33 } [get_ports { b_i[0] }]; #IO_L6
set_property -dict { PACKAGE_PIN R15      IOSTANDARD LVCMOS33 } [get_ports { b_i[1] }]; #IO_L1
set_property -dict { PACKAGE_PIN R17      IOSTANDARD LVCMOS33 } [get_ports { c_i[0] }]; #IO_L1
set_property -dict { PACKAGE_PIN T18      IOSTANDARD LVCMOS33 } [get_ports { c_i[1] }]; #IO_L7
set_property -dict { PACKAGE_PIN U18      IOSTANDARD LVCMOS33 } [get_ports { d_i[0] }]; #IO_L1
set_property -dict { PACKAGE_PIN R13      IOSTANDARD LVCMOS33 } [get_ports { d_i[1] }]; #IO_L5
set_property -dict { PACKAGE_PIN U11      IOSTANDARD LVCMOS33 } [get_ports { sel_i[14] }]; #IO
set_property -dict { PACKAGE_PIN V10      IOSTANDARD LVCMOS33 } [get_ports { sel_i[15] }]; #IO

## LEDs
set_property -dict { PACKAGE_PIN H17      IOSTANDARD LVCMOS33 } [get_ports { f_o[0] }]; #IO_L1
set_property -dict { PACKAGE_PIN K15      IOSTANDARD LVCMOS33 } [get_ports { f_o[1] }]; #IO_L2

```