Quickstart Guide to the Nexys-A7100T FPGA board and Xilinx Vivado

This guide provides a basic introduction to using the Xilinx Vivado software tools to program the Digilent Nexys-A7100T FPGA board. Basic Linux familiarity is assumed on the part of the reader.

1. Prepare the board to be powered from the USB cable and programmed by ensuring that:

Jumper JP3 is set to USB (the rightmost pin labelled WALL is exposed)

2. Set the configuration mode to USB-JTAG (to program the board directly from the USB-JTAB port using a USB cable) by ensuring that:

Jumper JP2 is set to USB (the top most pin exposed)
Jumper JP1 is set to USB/SD (two left most pins exposed)

Refer to Figure 1 for the positions of the three jumper settings.

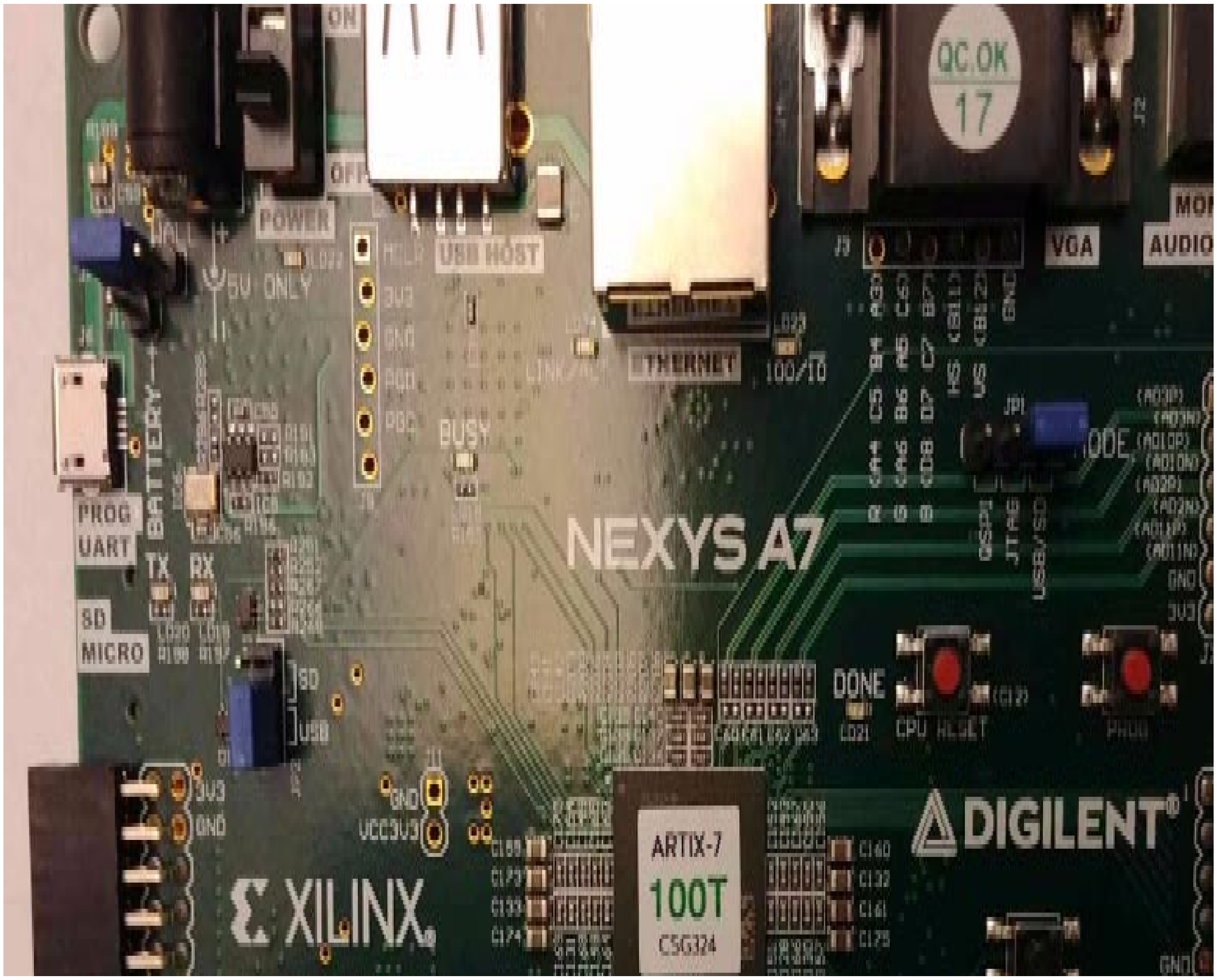


Figure 1: Board jumper settings.

Attach a USB cable to PROG/UART jplug on the baord insert the other end of the USB cable to the host computer. Your lab TA can assist with the procedure. After attaching the USB cable, move the Power slide switch to the ON position. To ensure that the necessary cable drivers are installed, issue the ‘lsusb’ command from the Linux prompt:

```
ted@panther ~/VIVADO 1:20pm >lsusb
Bus 002 Device 009: ID 413c:2107 Dell Computer Corp.
Bus 002 Device 012: ID 0403:6010 Future Technology Devices International, Ltd FT2232C/D/H Dual UART/FIFO IC
Bus 002 Device 005: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 002 Device 004: ID 046d:c077 Logitech, Inc. M105 Optical Mouse
Bus 002 Device 008: ID 03fd:0008 Xilinx, Inc. Platform Cable USB II
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 05e3:0608 Genesys Logic, Inc. Hub
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

The **`Bus 002 Device 012: ID 0403:6010 Future Technology Devices International`**, is the Nexysy FPGA board.

3. To setup the user’s Linux environment to run the Vivado tools, type (all one one line) from the Linux prommpt :

```
source  /CMC/tools/xilinx/Vivado_2018.2/Vivado/2018.2/settings64_CMC_central_license.csh
```

The message:

```
XILINXD_LICENSE_FILE is set to 6062@license-cadconnect:7062@license-cadconnect
```

will appear on the screen.

4. Create a working directory and `cd` into it:

```
mkdir COEN313
cd COEN313
mkdir LAB1
cd LAB1
```

Use a text editor to create the following three files:

half_adder_regular_outputs.vhd:

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

entity half_adder is
    port ( in1, in2 :   in std_logic;
           carry, sum : out std_logic);
end half_adder;

architecture true_outputs of half_adder is
begin
  carry <=  (in1 and in2);
  sum   <=  (in1 xor in2);
end true_outputs;
```

full_adder_nexysboard.vhd:

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

entity full_adder is
  port(carry_in, input1, input2 : in std_logic;
       sum_out, carry_out : out std_logic);
end full_adder;

architecture structural of full_adder is

-- declare a half-adder component

component half_adder
  port ( in1, in2 :   in std_logic;
         carry, sum : out std_logic);
end component;

-- declare internal signals used to “hook up” components
-- and to communicate to the display decoder process

signal carry1, carry2     : std_logic;
signal sum_int            : std_logic;

-- declare configuration specification
-- NOTE: we want to use the half adder with true outputs
-- not the inverted ones we synthesized earlier!!
```

```
for ha1, ha2 : half_adder use entity
WORK.half_adder(true_outputs);

begin

-- component instantiation

ha1: half_adder port map(in1 => input1, in2 => input2,
                         carry => carry1, sum => sum_int);

ha2: half_adder port map(in1 => sum_int, in2 => carry_in,
                         carry => carry2, sum => sum_out);

carry_out <= carry1 or carry2;


-- Nexys boards has active high LED outputs
-- so no need to negate top level output ports

end structural;
```

fulladder.xdc:

```
# Vivado does not support old UCF syntax
# must use XDC syntax


set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMOS33 } [ get_ports { carry_in } ] ;
set_property -dict { PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [ get_ports { input1 }   ] ;
set_property -dict { PACKAGE_PIN M13 IOSTANDARD LVCMOS33 } [ get_ports { input2 }  ] ;
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [ get_ports { carry_out } ] ;
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [ get_ports { sum_out }  ] ;
```

The fulladder.xdc file is a constraints file which is used by the Xilinx implementation tools to bind a specific top-level port in the VHDL entity to specific pins on the FPGA device. In this particular file, the three inputs `carry_in`, `input1`, `input2` are bound to pins connected to three of the switch inputs on the FPGA board, the the outputs `carry_out` and `sum_out` are bound to two of the LED outputs available on the FPGA board.

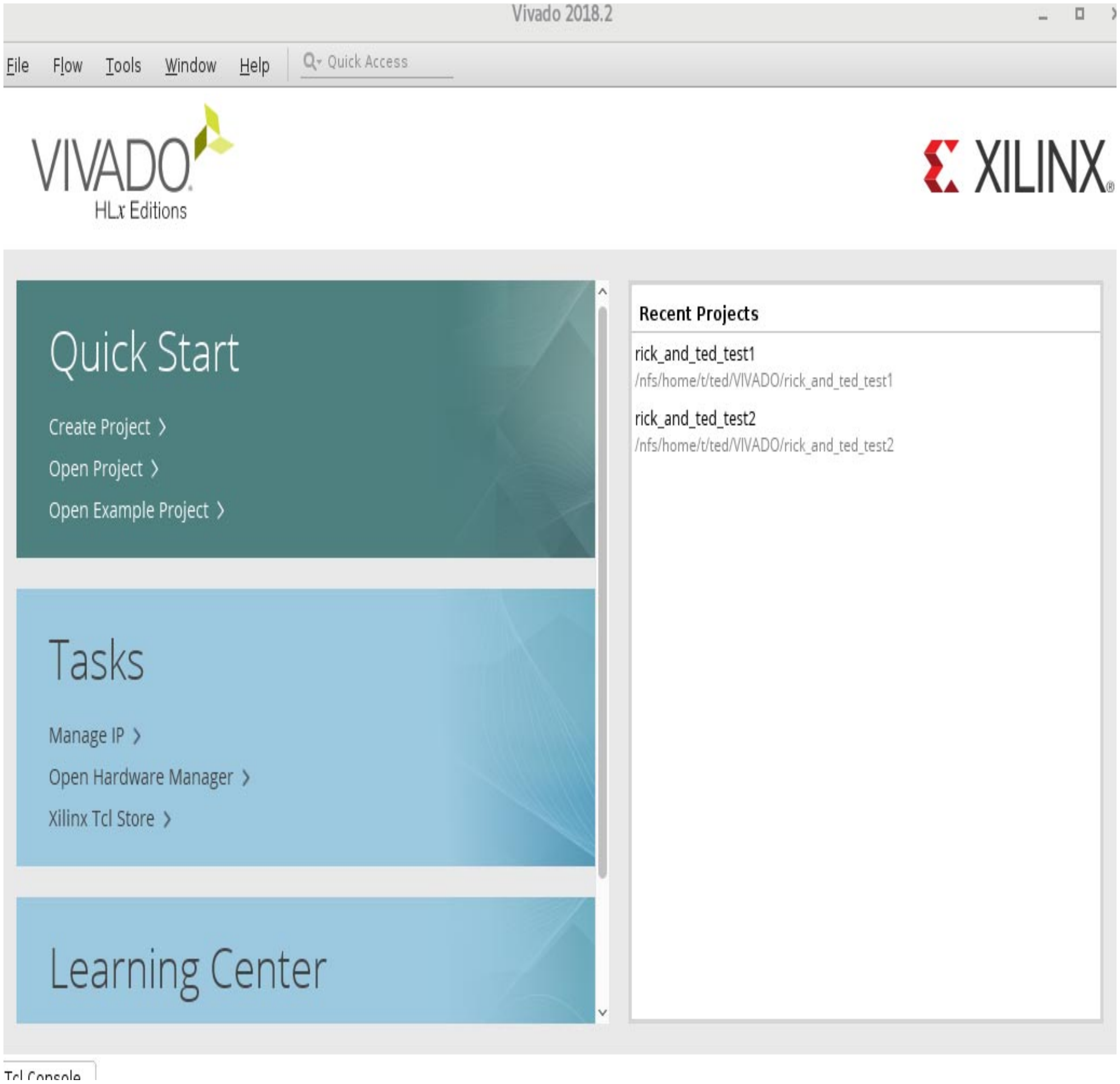
5. From the working directory created in step 4, start the Xilinx Vivado tools by invoking the ‘vivado &” command from the Linux prompt:

```
 ted@panther ~/VIVADO 1:17pm >  vivado &
```

After some time, the following messages will appear on the screen and after that the main Vivado GUI window will appear (be patient it takes a few minutes):

```
****** Vivado v2018.2 (64-bit)
  **** SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
  **** IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
    ** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.

start_gui
```

6. The main Vivado will appear as shown in Figure 2.



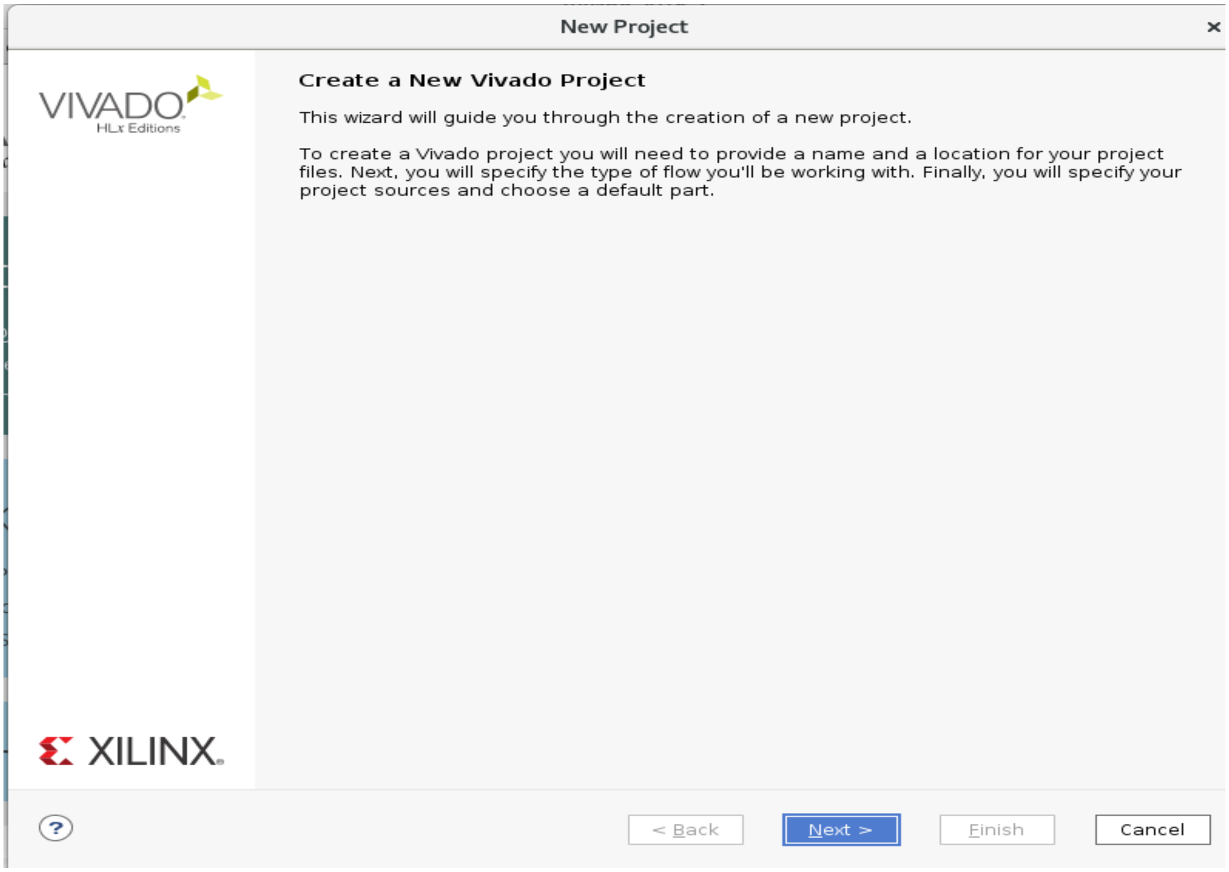
Recent Projects

rick_and_ted_test1
/nfs/home/t/ted/VIVADO/rick_and_ted_test1

rick_and_ted_test2
/nfs/home/t/ted/VIVADO/rick_and_ted_test2

Figure 2: Main Vivado window.

From the Quick Start pane, select Create Project. The Create New Project window appears as shown in Figure 3   Select Next.



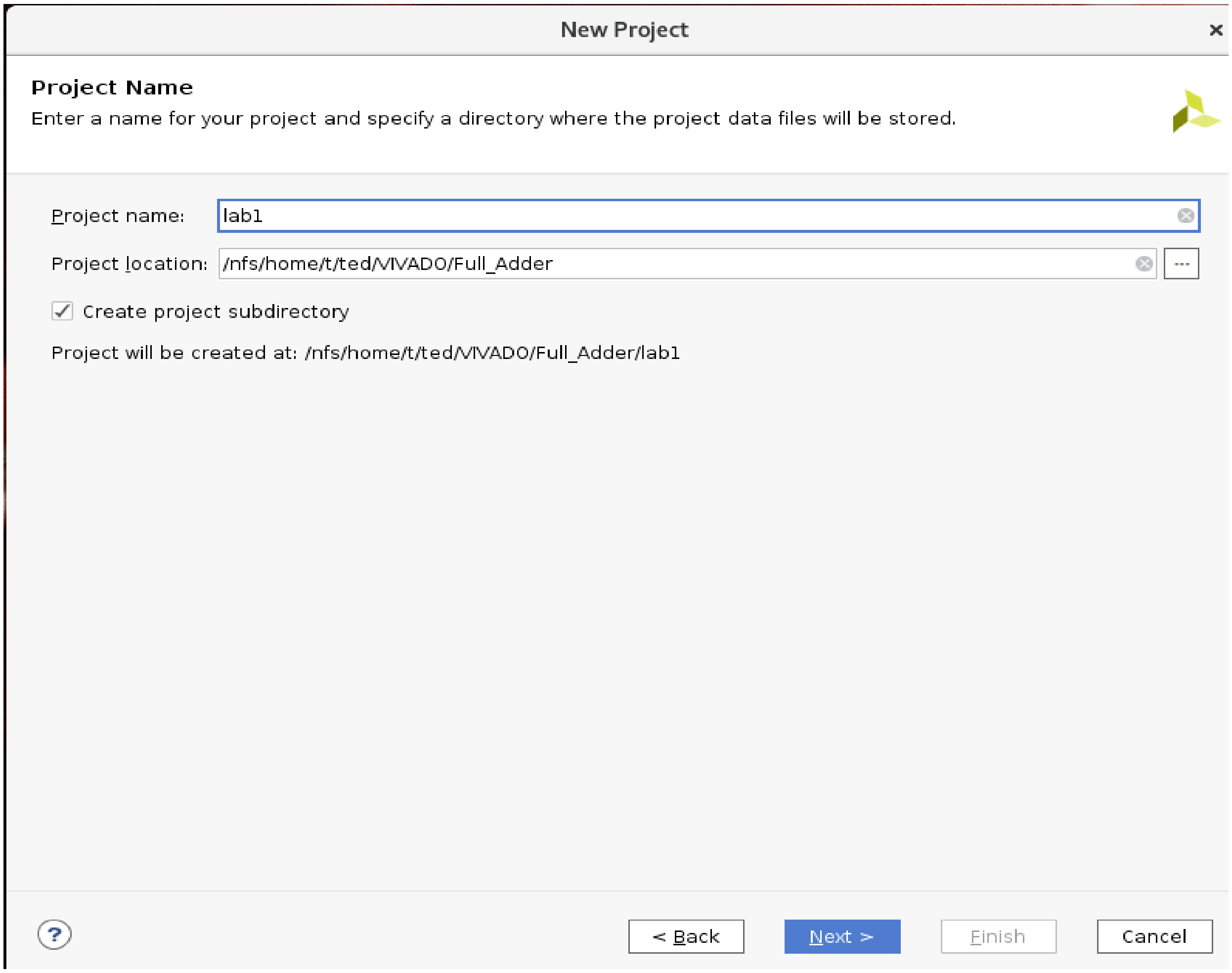
Create a New Vivado Project

This wizard will guide you through the creation of a new project.

To create a Vivado project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and choose a default part.

Fig 3 : New Project window.

Specify a Project name (such as Lab1, do **not** use blanks spaces in Project names) in the Project Name pane (Figure 4) . Select Next.



Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

Project name: lab1

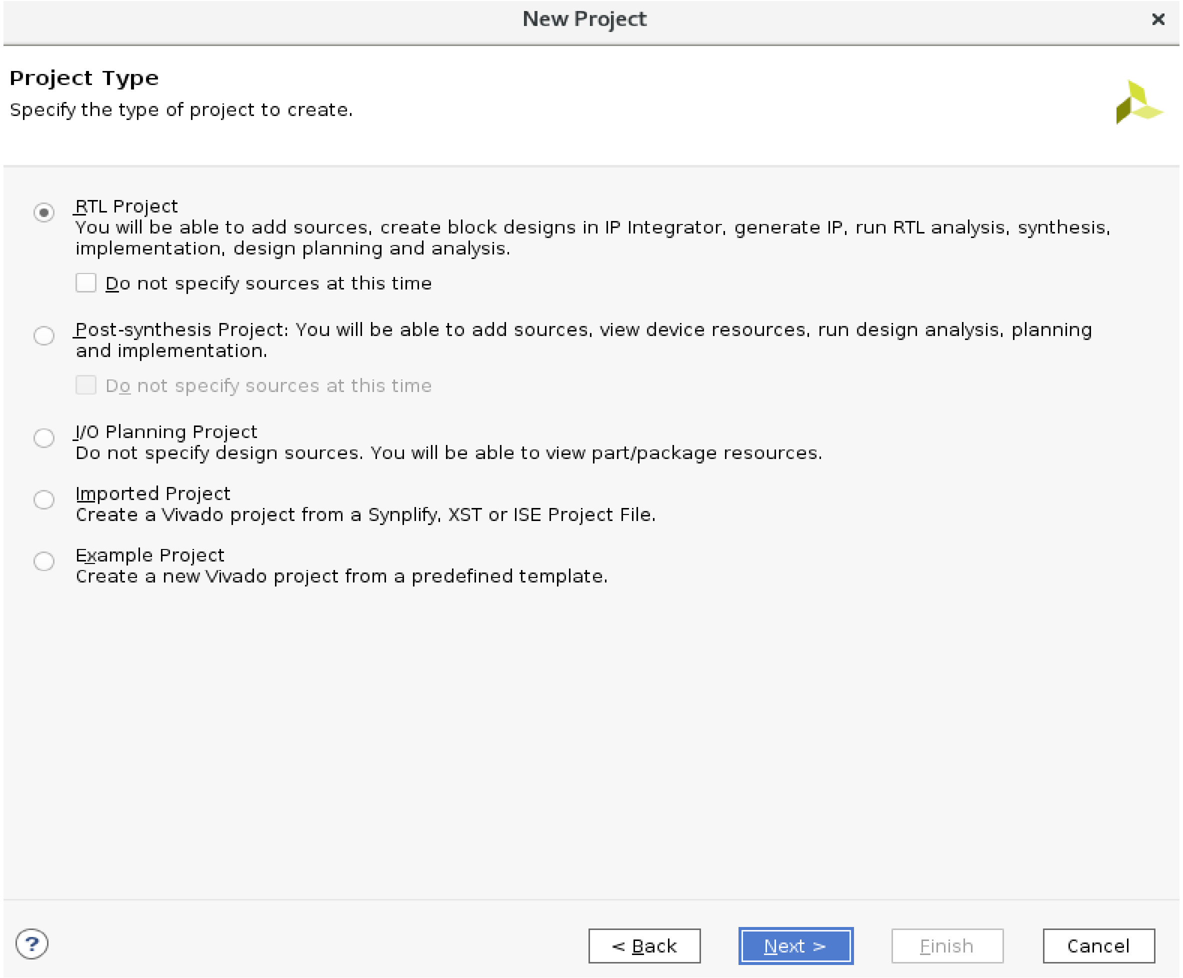
Project location: /nfs/home/t/ted/VIVADO/Full_Adder

☑ Create project subdirectory

Project will be created at: /nfs/home/t/ted/VIVADO/Full_Adder/lab1

Figure 4: Project name window.

Select RTL project in the Project Type form (Figure 5). Select Next.



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

○ 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.

Figure 5: Project type window.

In the Add sources pane, specify VHDL as the Target language and as the Simulator Language. Select Add files to specify the VHDL files to import . Select the two VHDL files:

half_adder_regular_outputs.vhd
full_adder_nexysboard.vhd

Refer to Figures 6 and 7.

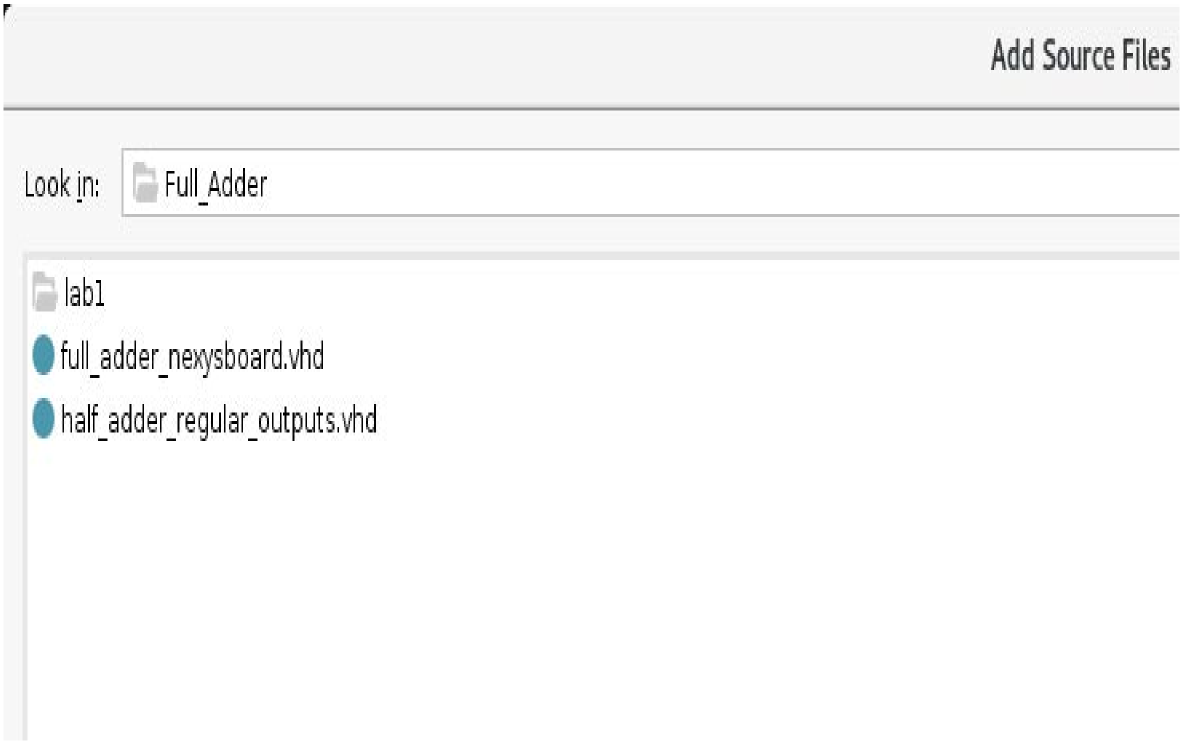
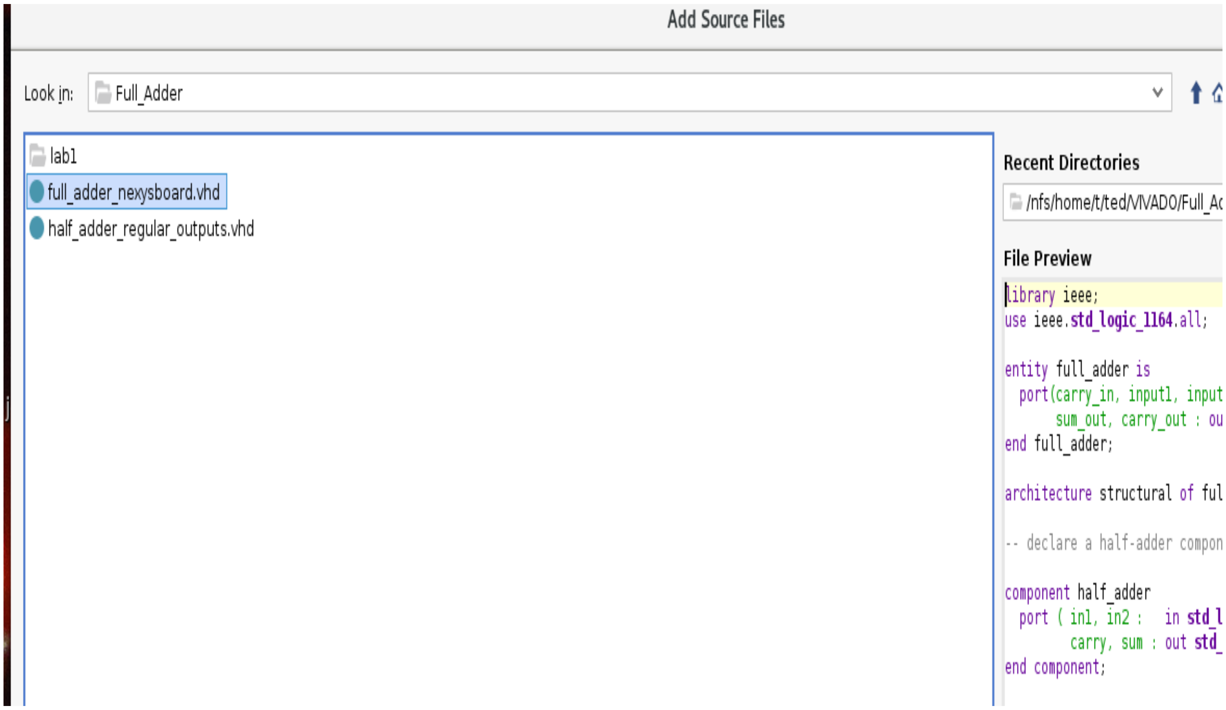


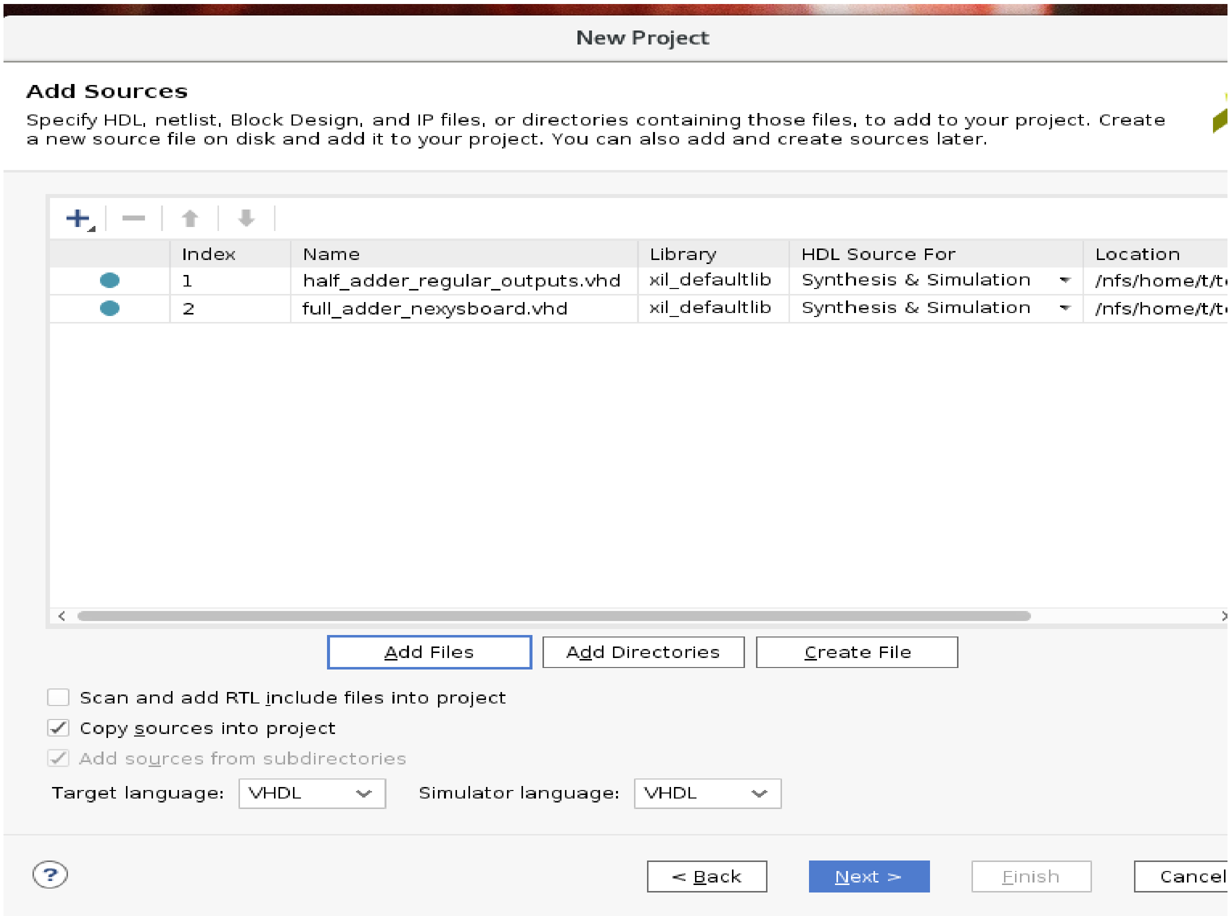
Figure 6: Adding source window.


Add Source Files
Look in: Full_Adder
lab1
full_adder_nexysboard.vhd
half_adder_regular_outputs.vhd
Recent Directories
/nfs/home/t/ted/VIVADO/Full_Ad
File Preview

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

entity full_adder is
  port(carry_in, input1, input
       sum_out, carry_out : ou
end full_adder;

architecture structural of ful

-- declare a half-adder compon

component half_adder
  port ( in1, in2 :   in std_l
         carry, sum : out std_
end component;
```

Figure 7: Adding source window.

After the files have been added, select Next in the Add Source window (Figure 8).



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.

	Index	Name	Library	HDL Source For	Location
●	1	half_adder_regular_outputs.vhd	xil_defaultlib	Synthesis & Simulation	/nfs/home/t/t
●	2	full_adder_nexysboard.vhd	xil_defaultlib	Synthesis & Simulation	/nfs/home/t/t

□ Scan and add RTL include files into project

☑ Copy sources into project

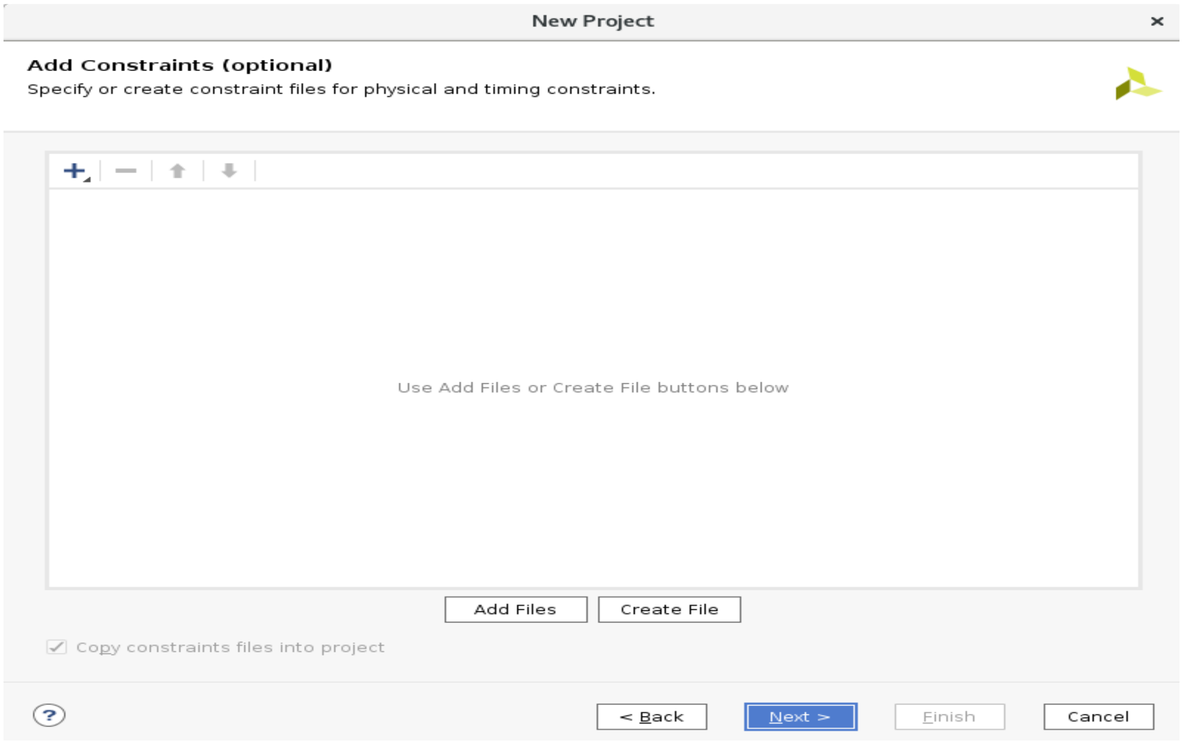
☑ Add sources from subdirectories

Target language: VHDL

Simulator language: VHDL

Figure 8: Adding source window after files have been added.

In the Add Constraints form, add the fulladder.xdc. See Figure 9 and 10.



New Project

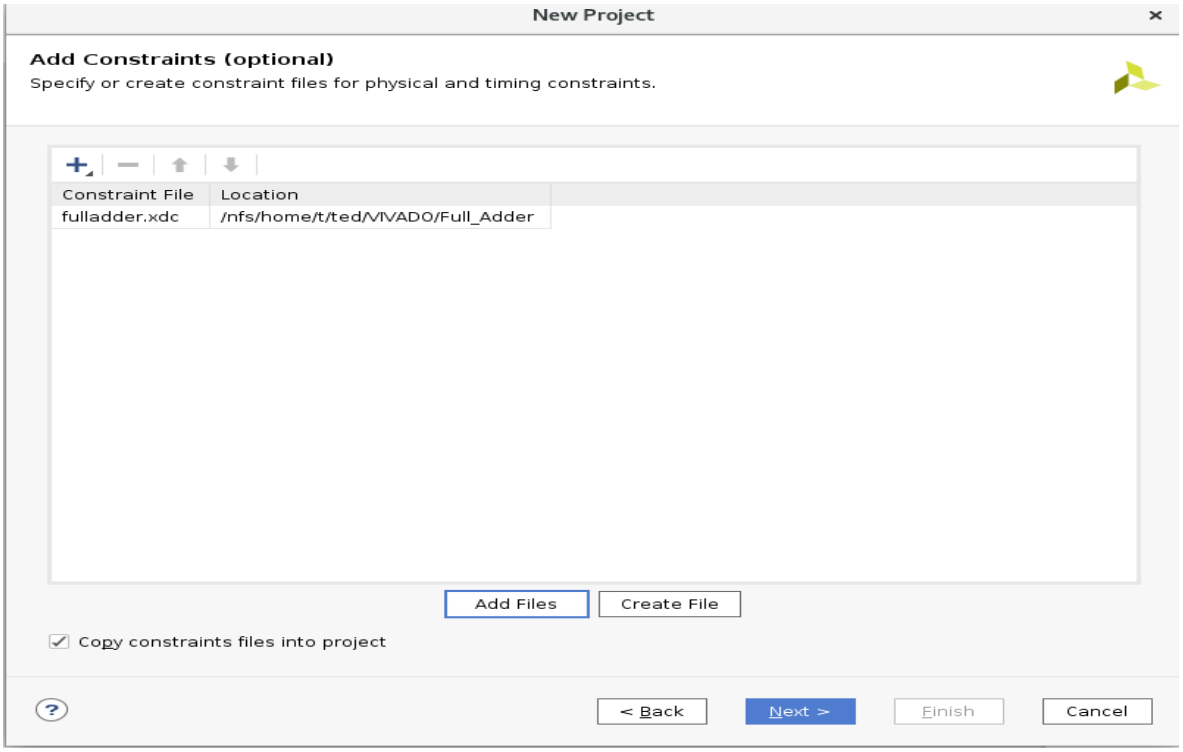
Add Constraints (optional)

Specify or create constraint files for physical and timing constraints.

Use Add Files or Create File buttons below

☑ Copy constraints files into project

Figure 9: Adding Constraint pane.



New Project

Add Constraints (optional)

Specify or create constraint files for physical and timing constraints.

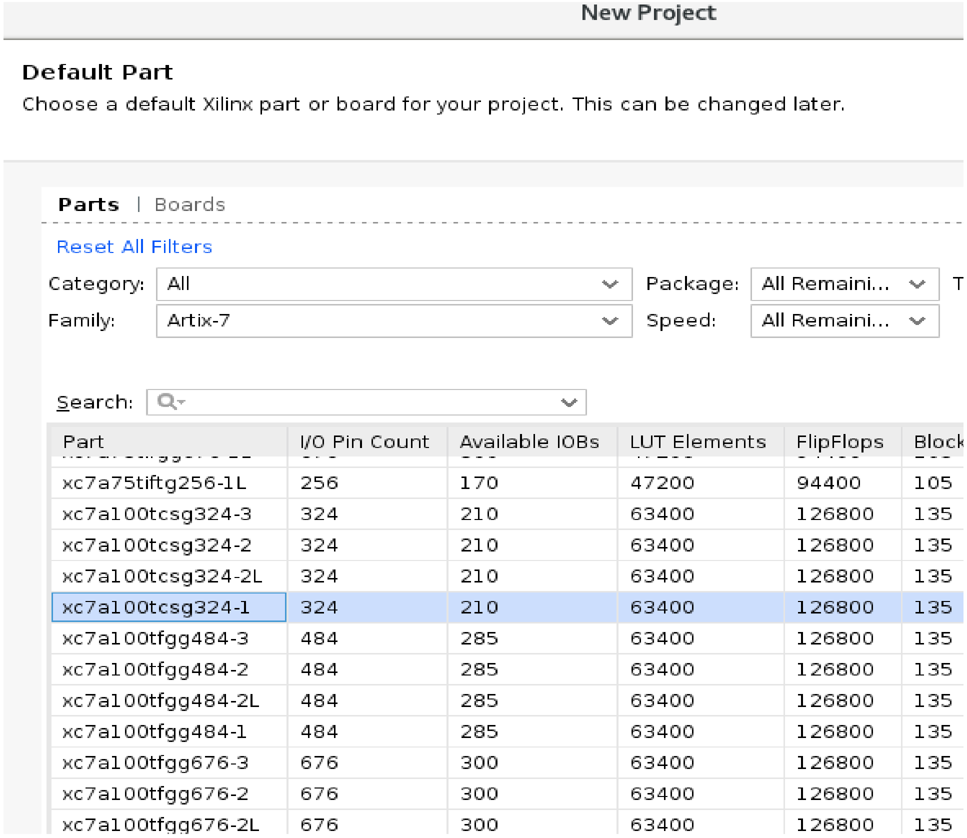
Constraint File	Location
fulladder.xdc	/nfs/home/t/ted/VIVADO/Full_Adder

☑ Copy constraints files into project

Figure 10: Constraints file selected.

In the Default part pane, specify `Family : Artix 7`
`device : xc7a100tcsg324-1`

Refer to Figure 11.



New Project

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

Category: All
Package: All Remaini...
Family: Artix-7
Speed: All Remaini...
Search:

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block
xc7a75tiftg256-1L	256	170	47200	94400	105
xc7a100tcsg324-3	324	210	63400	126800	135
xc7a100tcsg324-2	324	210	63400	126800	135
xc7a100tcsg324-2L	324	210	63400	126800	135
xc7a100tcsg324-1	324	210	63400	126800	135
xc7a100tfgg484-3	484	285	63400	126800	135
xc7a100tfgg484-2	484	285	63400	126800	135
xc7a100tfgg484-2L	484	285	63400	126800	135
xc7a100tfgg484-1	484	285	63400	126800	135
xc7a100tfgg676-3	676	300	63400	126800	135
xc7a100tfgg676-2	676	300	63400	126800	135
xc7a100tfgg676-2L	676	300	63400	126800	135

Figure 11: Device selected.

Select Next. Select Finish in the summary listing.

7. In the Project Manager pane, select the Constaints tab to list the fulladder.xdc file, right click this file and select “Set as Target Constaints file” . Refer to Figure 12.

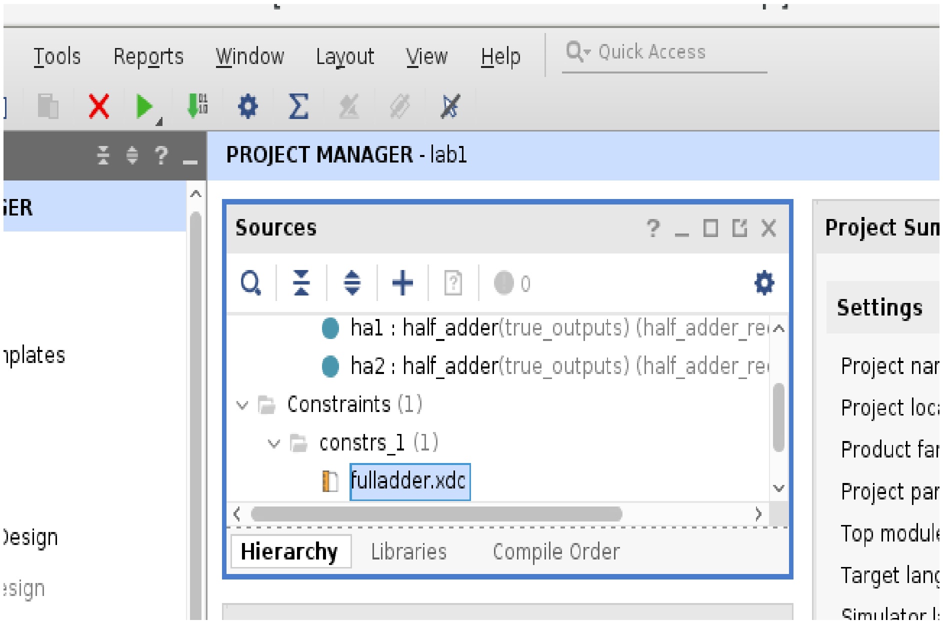
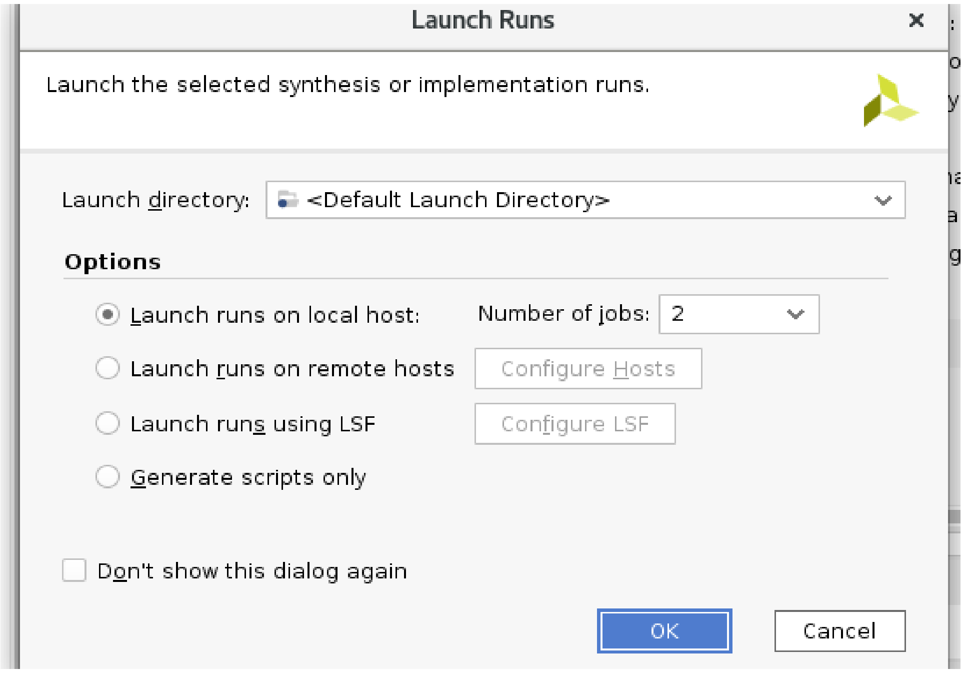


Figure 12: Selecting the constaint file.

Select Run Synthesis, leave Launch Runs on local host as the default setting in the Launch runs window (Figure 13) and select OK.


Launch Runs

Launch the selected synthesis or implementation runs.

Launch directory: <Default Launch Directory>

Options

● Launch runs on local host: Number of jobs: 2

○ Launch runs on remote hosts

○ Launch runs using LSF

○ Generate scripts only

□ Don't show this dialog again

Figure 13: Launch runs window.

Wait until the “synthesis completed message appears and then select “run implementation” Selrct OK in the default Launch Runs on local host in the Launch Runs window.

Wait until the Implementation has completed

Select Generate Bitstream. Select OK in the Launch Runs on local host in the Launch Runs window and wait until the bitstream generation has completed.

8. To program the FPGA board with the .bit file:

Select Open Hardware Manager in the main Vivado window. in the Hardware Manager pane (top part of the Vivado window, refer to Figure 14) the green pane will indicate:

No hardware target is open. Open Target

Select the Open Target and select :Autoconnect.

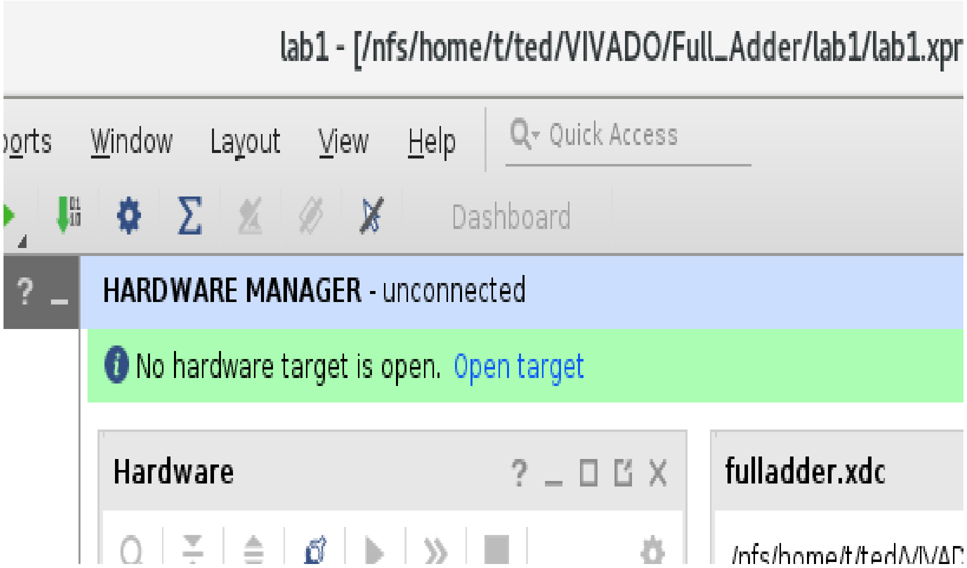
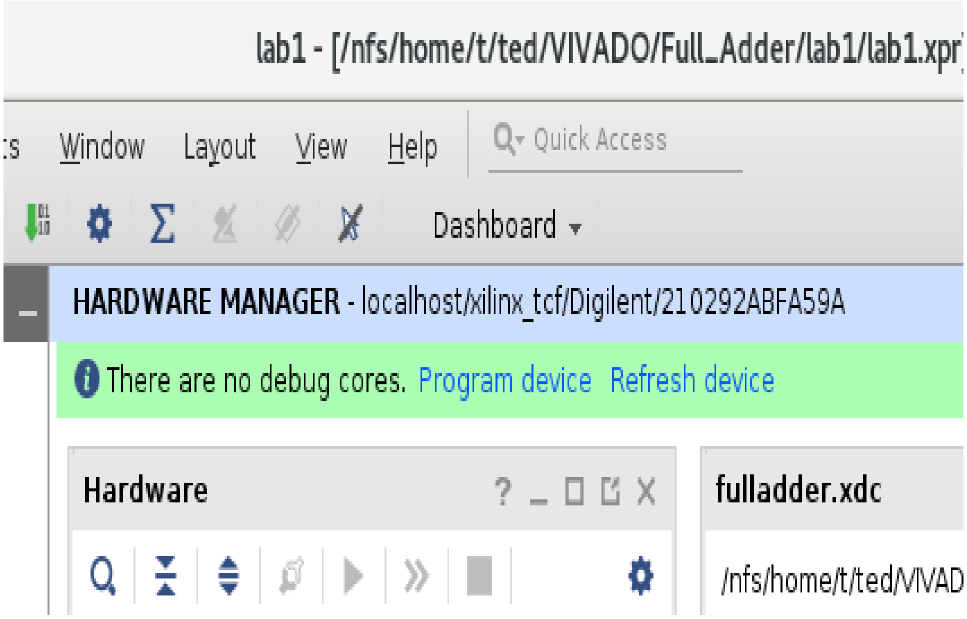

lab1 - [/nfs/home/t/ted/VIVADO/Full_Adder/lab1/lab1.xpr
ports
Window
Layout
View
Help
Quick Access
Dashboard
HARDWARE MANAGER - unconnected
No hardware target is open. Open target
Hardware
fulladder.xdc
/nfs/home/t/ted/VIVAD

Figure 14: Hardware Manager pane with Open Target choice.

The Xilinx device will now be listed in the Hardware Manager pane (-localhost/xilinx_tcf/Digilent/21029ABFA59A - the actual numbers may be different in your listing). In the green pane (Figure 15)will be

“There are no debug cores. Program Device Refresh Device.

Select Program Device.



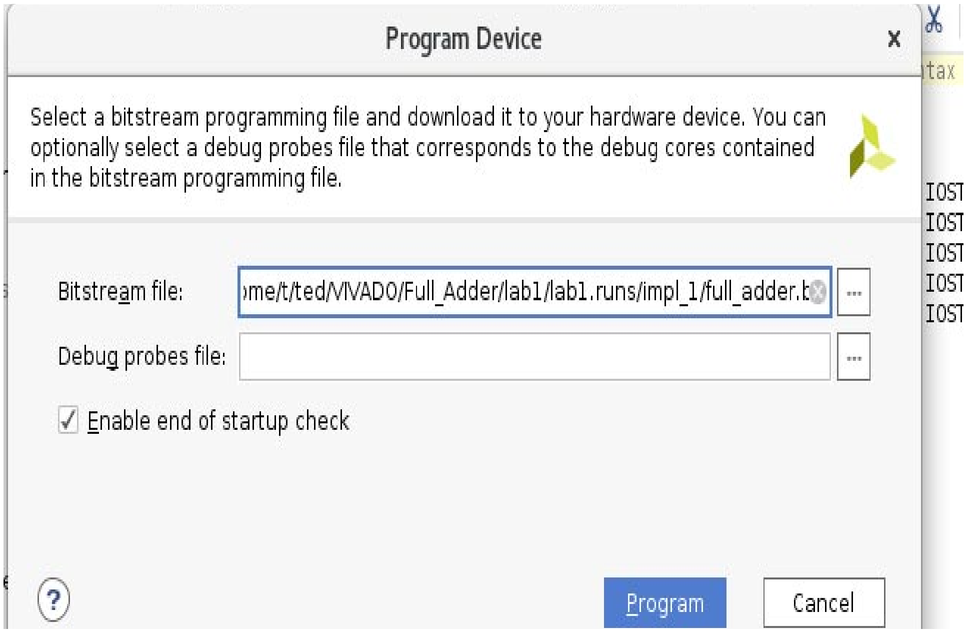
lab1 - [/nfs/home/t/ted/VIVADO/Full_Adder/lab1/lab1.xpr

HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210292ABFA59A

There are no debug cores. Program device Refresh device

Figure 15: Hardware Manager Program Device pane.

A window (Figure 16) appears listing the path to the .bit file, select Program.



Program Device

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: ›me/t/ted/VIVADO/Full_Adder/lab1/lab1.runs/impl_1/full_adder.b

Debug probes file:

☑ Enable end of startup check

Figure 16: Selecting the .bit file.

Test your downloaded design by using the slide switches to set the values of the three inputs and observe the outputs on the two LEDs.

Nexys-A7100T FPGA board

The 16 LEDs are active high. The 16 slide switches produce a logic -1 when in the “UP” position, and a logic - 0 when in the “DOWN” position. The following is a general.xdc file listing all the available input/outputs on the Nexys-A7100T FPGA board and the package pins they are connected to.

```
## This file is a general .xdc for the Nexys A7-100T
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project

## Clock signal
#set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { CLK100MHZ }]; #IO_L12P_T1_MRCC_35
Sch=clk100mhz
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];


##Switches
#set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { SW[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
#set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { SW[1] }]; #IO_L3N_T0_DQS_EMCCLK_14
Sch=sw[1]
#set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { SW[2] }]; #IO_L6N_T0_D08_VREF_14
Sch=sw[2]
#set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { SW[3] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
#set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { SW[4] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { SW[5] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
#set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports { SW[6] }]; #IO_L17N_T2_A13_D29_14
Sch=sw[6]
#set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports { SW[7] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
#set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS18 } [get_ports { SW[8] }]; #IO_L24N_T3_34 Sch=sw[8]
#set_property -dict { PACKAGE_PIN U8    IOSTANDARD LVCMOS18 } [get_ports { SW[9] }]; #IO_25_34 Sch=sw[9]
#set_property -dict { PACKAGE_PIN R16   IOSTANDARD LVCMOS33 } [get_ports { SW[10] }]; #IO_L15P_T2_DQS_RDWR_B_14
Sch=sw[10]
#set_property -dict { PACKAGE_PIN T13   IOSTANDARD LVCMOS33 } [get_ports { SW[11] }]; #IO_L23P_T3_A03_D19_14
Sch=sw[11]
#set_property -dict { PACKAGE_PIN H6    IOSTANDARD LVCMOS33 } [get_ports { SW[12] }]; #IO_L24P_T3_35 Sch=sw[12]
#set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports { SW[13] }]; #IO_L20P_T3_A08_D24_14
Sch=sw[13]
#set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports { SW[14] }]; #IO_L19N_T3_A09_D25_VREF_14
Sch=sw[14]
#set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports { SW[15] }]; #IO_L21P_T3_DQS_14 Sch=sw[15]

## LEDs
#set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { LED[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
#set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { LED[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
#set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { LED[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
#set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports { LED[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
#set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports { LED[4] }]; #IO_L7P_T1_D09_14 Sch=led[4]
#set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { LED[5] }]; #IO_L18N_T2_A11_D27_14
Sch=led[5]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports { LED[6] }]; #IO_L17P_T2_A14_D30_14
Sch=led[6]
#set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { LED[7] }]; #IO_L18P_T2_A12_D28_14
Sch=led[7]
#set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { LED[8] }]; #IO_L16N_T2_A15_D31_14
Sch=led[8]
#set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports { LED[9] }]; #IO_L14N_T2_SRCC_14
Sch=led[9]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports { LED[10] }]; #IO_L22P_T3_A05_D21_14
Sch=led[10]
#set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports { LED[11] }];
#IO_L15N_T2_DQS_DOUT_CSO_B_14 Sch=led[11]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports { LED[12] }]; #IO_L16P_T2_CSI_B_14
Sch=led[12]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports { LED[13] }]; #IO_L22N_T3_A04_D20_14
Sch=led[13]
#set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports { LED[14] }]; #IO_L20N_T3_A07_D23_14
Sch=led[14]
#set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports { LED[15] }]; #IO_L21N_T3_DQS_A06_D22_14
Sch=led[15

## RGB LEDs
```

```
#set_property -dict { PACKAGE_PIN R12   IOSTANDARD LVCMOS33 } [get_ports { LED16_B }]; #IO_L5P_T0_D06_14
Sch=led16_b
#set_property -dict { PACKAGE_PIN M16   IOSTANDARD LVCMOS33 } [get_ports { LED16_G }]; #IO_L10P_T1_D14_14
Sch=led16_g
#set_property -dict { PACKAGE_PIN N15   IOSTANDARD LVCMOS33 } [get_ports { LED16_R }]; #IO_L11P_T1_SRCC_14
Sch=led16_r
#set_property -dict { PACKAGE_PIN G14   IOSTANDARD LVCMOS33 } [get_ports { LED17_B }]; #IO_L15N_T2_DQS_ADV_B_15
Sch=led17_b
#set_property -dict { PACKAGE_PIN R11   IOSTANDARD LVCMOS33 } [get_ports { LED17_G }]; #IO_0_14 Sch=led17_g
#set_property -dict { PACKAGE_PIN N16   IOSTANDARD LVCMOS33 } [get_ports { LED17_R }]; #IO_L11N_T1_SRCC_14
Sch=led17_r

##7 segment display
#set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports { CA }]; #IO_L24N_T3_A00_D16_14 Sch=ca
#set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports { CB }]; #IO_25_14 Sch=cb
#set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports { CC }]; #IO_25_15 Sch=cc
#set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports { CD }]; #IO_L17P_T2_A26_15 Sch=cd
#set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports { CE }]; #IO_L13P_T2_MRCC_14 Sch=ce
#set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports { CF }]; #IO_L19P_T3_A10_D26_14 Sch=cf
#set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports { CG }]; #IO_L4P_T0_D04_14 Sch=cg
#set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports { DP }]; #IO_L19N_T3_A21_VREF_15 Sch=dp
#set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports { AN[0] }]; #IO_L23P_T3_FOE_B_15 Sch=an[0]
#set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports { AN[1] }]; #IO_L23N_T3_FWE_B_15 Sch=an[1]
#set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports { AN[2] }]; #IO_L24P_T3_A01_D17_14
Sch=an[2]
#set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports { AN[3] }]; #IO_L19P_T3_A22_15 Sch=an[3]
#set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 } [get_ports { AN[4] }]; #IO_L8N_T1_D12_14 Sch=an[4]
#set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports { AN[5] }]; #IO_L14P_T2_SRCC_14 Sch=an[5]
#set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVCMOS33 } [get_ports { AN[6] }]; #IO_L23P_T3_35 Sch=an[6]
#set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 } [get_ports { AN[7] }]; #IO_L23N_T3_A02_D18_14
Sch=an[7]

##Buttons
#set_property -dict { PACKAGE_PIN C12   IOSTANDARD LVCMOS33 } [get_ports { CPU_RESETN }]; #IO_L3P_T0_DQS_AD1P_15
Sch=cpu_resetn
#set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports { BTNC }]; #IO_L9P_T1_DQS_14 Sch=btnc
#set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 } [get_ports { BTNU }]; #IO_L4N_T0_D05_14 Sch=btnu
#set_property -dict { PACKAGE_PIN P17   IOSTANDARD LVCMOS33 } [get_ports { BTNL }]; #IO_L12P_T1_MRCC_14 Sch=btnl
#set_property -dict { PACKAGE_PIN M17   IOSTANDARD LVCMOS33 } [get_ports { BTNR }]; #IO_L10N_T1_D15_14 Sch=btnr
#set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVCMOS33 } [get_ports { BTND }]; #IO_L9N_T1_DQS_D13_14 Sch=btnd


##Pmod Headers
##Pmod Header JA
#set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33 } [get_ports { JA[1] }]; #IO_L20N_T3_A19_15 Sch=ja[1]
#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports { JA[2] }]; #IO_L21N_T3_DQS_A18_15
Sch=ja[2]
#set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 } [get_ports { JA[3] }]; #IO_L21P_T3_DQS_15 Sch=ja[3]
#set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports { JA[4] }]; #IO_L18N_T2_A23_15 Sch=ja[4]
#set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports { JA[7] }]; #IO_L16N_T2_A27_15 Sch=ja[7]
#set_property -dict { PACKAGE_PIN E17   IOSTANDARD LVCMOS33 } [get_ports { JA[8] }]; #IO_L16P_T2_A28_15 Sch=ja[8]
#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports { JA[9] }]; #IO_L22N_T3_A16_15 Sch=ja[9]
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports { JA[10] }]; #IO_L22P_T3_A17_15 Sch=ja[10]

##Pmod Header JB
#set_property -dict { PACKAGE_PIN D14   IOSTANDARD LVCMOS33 } [get_ports { JB[1] }]; #IO_L1P_T0_AD0P_15 Sch=jb[1]
#set_property -dict { PACKAGE_PIN F16   IOSTANDARD LVCMOS33 } [get_ports { JB[2] }]; #IO_L14N_T2_SRCC_15 Sch=jb[2]
#set_property -dict { PACKAGE_PIN G16   IOSTANDARD LVCMOS33 } [get_ports { JB[3] }]; #IO_L13N_T2_MRCC_15 Sch=jb[3]
#set_property -dict { PACKAGE_PIN H14   IOSTANDARD LVCMOS33 } [get_ports { JB[4] }]; #IO_L15P_T2_DQS_15 Sch=jb[4]
#set_property -dict { PACKAGE_PIN E16   IOSTANDARD LVCMOS33 } [get_ports { JB[7] }]; #IO_L11N_T1_SRCC_15 Sch=jb[7]
#set_property -dict { PACKAGE_PIN F13   IOSTANDARD LVCMOS33 } [get_ports { JB[8] }]; #IO_L5P_T0_AD9P_15 Sch=jb[8]
#set_property -dict { PACKAGE_PIN G13   IOSTANDARD LVCMOS33 } [get_ports { JB[9] }]; #IO_0_15 Sch=jb[9]
#set_property -dict { PACKAGE_PIN H16   IOSTANDARD LVCMOS33 } [get_ports { JB[10] }]; #IO_L13P_T2_MRCC_15
Sch=jb[10]

##Pmod Header JC
#set_property -dict { PACKAGE_PIN K1    IOSTANDARD LVCMOS33 } [get_ports { JC[1] }]; #IO_L23N_T3_35 Sch=jc[1]
#set_property -dict { PACKAGE_PIN F6    IOSTANDARD LVCMOS33 } [get_ports { JC[2] }]; #IO_L19N_T3_VREF_35 Sch=jc[2]
#set_property -dict { PACKAGE_PIN J2    IOSTANDARD LVCMOS33 } [get_ports { JC[3] }]; #IO_L22N_T3_35 Sch=jc[3]
#set_property -dict { PACKAGE_PIN G6    IOSTANDARD LVCMOS33 } [get_ports { JC[4] }]; #IO_L19P_T3_35 Sch=jc[4]
#set_property -dict { PACKAGE_PIN E7    IOSTANDARD LVCMOS33 } [get_ports { JC[7] }]; #IO_L6P_T0_35 Sch=jc[7]
#set_property -dict { PACKAGE_PIN J3    IOSTANDARD LVCMOS33 } [get_ports { JC[8] }]; #IO_L22P_T3_35 Sch=jc[8]
#set_property -dict { PACKAGE_PIN J4    IOSTANDARD LVCMOS33 } [get_ports { JC[9] }]; #IO_L21P_T3_DQS_35 Sch=jc[9]
#set_property -dict { PACKAGE_PIN E6    IOSTANDARD LVCMOS33 } [get_ports { JC[10] }]; #IO_L5P_T0_AD13P_35
Sch=jc[10]

##Pmod Header JD
#set_property -dict { PACKAGE_PIN H4    IOSTANDARD LVCMOS33 } [get_ports { JD[1] }]; #IO_L21N_T3_DQS_35 Sch=jd[1]
#set_property -dict { PACKAGE_PIN H1    IOSTANDARD LVCMOS33 } [get_ports { JD[2] }]; #IO_L17P_T2_35 Sch=jd[2]
#set_property -dict { PACKAGE_PIN G1    IOSTANDARD LVCMOS33 } [get_ports { JD[3] }]; #IO_L17N_T2_35 Sch=jd[3]
#set_property -dict { PACKAGE_PIN G3    IOSTANDARD LVCMOS33 } [get_ports { JD[4] }]; #IO_L20N_T3_35 Sch=jd[4]
```

```
#set_property -dict { PACKAGE_PIN H2    IOSTANDARD LVCMOS33 } [get_ports { JD[7] }]; #IO_L15P_T2_DQS_35 Sch=jd[7]
#set_property -dict { PACKAGE_PIN G4    IOSTANDARD LVCMOS33 } [get_ports { JD[8] }]; #IO_L20P_T3_35 Sch=jd[8]
#set_property -dict { PACKAGE_PIN G2    IOSTANDARD LVCMOS33 } [get_ports { JD[9] }]; #IO_L15N_T2_DQS_35 Sch=jd[9]
#set_property -dict { PACKAGE_PIN F3    IOSTANDARD LVCMOS33 } [get_ports { JD[10] }]; #IO_L13N_T2_MRCC_35
Sch=jd[10]

##Pmod Header JXADC
#set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33 } [get_ports { XA_N[1] }]; #IO_L9N_T1_DQS_AD3N_15
Sch=xa_n[1]
#set_property -dict { PACKAGE_PIN A13   IOSTANDARD LVCMOS33 } [get_ports { XA_P[1] }]; #IO_L9P_T1_DQS_AD3P_15
Sch=xa_p[1]
#set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMOS33 } [get_ports { XA_N[2] }]; #IO_L8N_T1_AD10N_15
Sch=xa_n[2]
#set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVCMOS33 } [get_ports { XA_P[2] }]; #IO_L8P_T1_AD10P_15
Sch=xa_p[2]
#set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33 } [get_ports { XA_N[3] }]; #IO_L7N_T1_AD2N_15
Sch=xa_n[3]
#set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMOS33 } [get_ports { XA_P[3] }]; #IO_L7P_T1_AD2P_15
Sch=xa_p[3]
#set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports { XA_N[4] }]; #IO_L10N_T1_AD11N_15
Sch=xa_n[4]
#set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports { XA_P[4] }]; #IO_L10P_T1_AD11P_15
Sch=xa_p[4]

##VGA Connector
#set_property -dict { PACKAGE_PIN A3    IOSTANDARD LVCMOS33 } [get_ports { VGA_R[0] }]; #IO_L8N_T1_AD14N_35
Sch=vga_r[0]
#set_property -dict { PACKAGE_PIN B4    IOSTANDARD LVCMOS33 } [get_ports { VGA_R[1] }]; #IO_L7N_T1_AD6N_35
Sch=vga_r[1]
#set_property -dict { PACKAGE_PIN C5    IOSTANDARD LVCMOS33 } [get_ports { VGA_R[2] }]; #IO_L1N_T0_AD4N_35
Sch=vga_r[2]
#set_property -dict { PACKAGE_PIN A4    IOSTANDARD LVCMOS33 } [get_ports { VGA_R[3] }]; #IO_L8P_T1_AD14P_35
Sch=vga_r[3]
#set_property -dict { PACKAGE_PIN C6    IOSTANDARD LVCMOS33 } [get_ports { VGA_G[0] }]; #IO_L1P_T0_AD4P_35
Sch=vga_g[0]
#set_property -dict { PACKAGE_PIN A5    IOSTANDARD LVCMOS33 } [get_ports { VGA_G[1] }]; #IO_L3N_T0_DQS_AD5N_35
Sch=vga_g[1]
#set_property -dict { PACKAGE_PIN B6    IOSTANDARD LVCMOS33 } [get_ports { VGA_G[2] }]; #IO_L2N_T0_AD12N_35
Sch=vga_g[2]
#set_property -dict { PACKAGE_PIN A6    IOSTANDARD LVCMOS33 } [get_ports { VGA_G[3] }]; #IO_L3P_T0_DQS_AD5P_35
Sch=vga_g[3]
#set_property -dict { PACKAGE_PIN B7    IOSTANDARD LVCMOS33 } [get_ports { VGA_B[0] }]; #IO_L2P_T0_AD12P_35
Sch=vga_b[0]
#set_property -dict { PACKAGE_PIN C7    IOSTANDARD LVCMOS33 } [get_ports { VGA_B[1] }]; #IO_L4N_T0_35 Sch=vga_b[1]
#set_property -dict { PACKAGE_PIN D7    IOSTANDARD LVCMOS33 } [get_ports { VGA_B[2] }]; #IO_L6N_T0_VREF_35
Sch=vga_b[2]
#set_property -dict { PACKAGE_PIN D8    IOSTANDARD LVCMOS33 } [get_ports { VGA_B[3] }]; #IO_L4P_T0_35 Sch=vga_b[3]
#set_property -dict { PACKAGE_PIN B11   IOSTANDARD LVCMOS33 } [get_ports { VGA_HS }]; #IO_L4P_T0_15 Sch=vga_hs
#set_property -dict { PACKAGE_PIN B12   IOSTANDARD LVCMOS33 } [get_ports { VGA_VS }]; #IO_L3N_T0_DQS_AD1N_15
Sch=vga_vs

##Micro SD Connector
#set_property -dict { PACKAGE_PIN E2    IOSTANDARD LVCMOS33 } [get_ports { SD_RESET }]; #IO_L14P_T2_SRCC_35
Sch=sd_reset
#set_property -dict { PACKAGE_PIN A1    IOSTANDARD LVCMOS33 } [get_ports { SD_CD }]; #IO_L9N_T1_DQS_AD7N_35
Sch=sd_cd
#set_property -dict { PACKAGE_PIN B1    IOSTANDARD LVCMOS33 } [get_ports { SD_SCK }]; #IO_L9P_T1_DQS_AD7P_35
Sch=sd_sck
#set_property -dict { PACKAGE_PIN C1    IOSTANDARD LVCMOS33 } [get_ports { SD_CMD }]; #IO_L16N_T2_35 Sch=sd_cmd
#set_property -dict { PACKAGE_PIN C2    IOSTANDARD LVCMOS33 } [get_ports { SD_DAT[0] }]; #IO_L16P_T2_35
Sch=sd_dat[0]
#set_property -dict { PACKAGE_PIN E1    IOSTANDARD LVCMOS33 } [get_ports { SD_DAT[1] }]; #IO_L18N_T2_35
Sch=sd_dat[1]
#set_property -dict { PACKAGE_PIN F1    IOSTANDARD LVCMOS33 } [get_ports { SD_DAT[2] }]; #IO_L18P_T2_35
Sch=sd_dat[2]
#set_property -dict { PACKAGE_PIN D2    IOSTANDARD LVCMOS33 } [get_ports { SD_DAT[3] }]; #IO_L14N_T2_SRCC_35
Sch=sd_dat[3]

##Accelerometer
#set_property -dict { PACKAGE_PIN E15   IOSTANDARD LVCMOS33 } [get_ports { ACL_MISO }]; #IO_L11P_T1_SRCC_15
Sch=acl_miso
#set_property -dict { PACKAGE_PIN F14   IOSTANDARD LVCMOS33 } [get_ports { ACL_MOSI }]; #IO_L5N_T0_AD9N_15
Sch=acl_mosi
#set_property -dict { PACKAGE_PIN F15   IOSTANDARD LVCMOS33 } [get_ports { ACL_SCLK }]; #IO_L14P_T2_SRCC_15
Sch=acl_sclk
#set_property -dict { PACKAGE_PIN D15   IOSTANDARD LVCMOS33 } [get_ports { ACL_CSN }]; #IO_L12P_T1_MRCC_15
Sch=acl_csn
#set_property -dict { PACKAGE_PIN B13   IOSTANDARD LVCMOS33 } [get_ports { ACL_INT[1] }]; #IO_L2P_T0_AD8P_15
Sch=acl_int[1]
#set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVCMOS33 } [get_ports { ACL_INT[2] }]; #IO_L20P_T3_A20_15
Sch=acl_int[2]
```

```
##Temperature Sensor
#set_property -dict { PACKAGE_PIN C14   IOSTANDARD LVCMOS33 } [get_ports { TMP_SCL }]; #IO_L1N_T0_AD0N_15
Sch=tmp_scl
#set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVCMOS33 } [get_ports { TMP_SDA }]; #IO_L12N_T1_MRCC_15
Sch=tmp_sda
#set_property -dict { PACKAGE_PIN D13   IOSTANDARD LVCMOS33 } [get_ports { TMP_INT }]; #IO_L6N_T0_VREF_15
Sch=tmp_int
#set_property -dict { PACKAGE_PIN B14   IOSTANDARD LVCMOS33 } [get_ports { TMP_CT }]; #IO_L2N_T0_AD8N_15 Sch=tmp_ct

##Omnidirectional Microphone
#set_property -dict { PACKAGE_PIN J5    IOSTANDARD LVCMOS33 } [get_ports { M_CLK }]; #IO_25_35 Sch=m_clk
#set_property -dict { PACKAGE_PIN H5    IOSTANDARD LVCMOS33 } [get_ports { M_DATA }]; #IO_L24N_T3_35 Sch=m_data
#set_property -dict { PACKAGE_PIN F5    IOSTANDARD LVCMOS33 } [get_ports { M_LRSEL }]; #IO_0_35 Sch=m_lrsel

##PWM Audio Amplifier

#set_property -dict { PACKAGE_PIN A11   IOSTANDARD LVCMOS33 } [get_ports { AUD_PWM }]; #IO_L4N_T0_15 Sch=aud_pwm
#set_property -dict { PACKAGE_PIN D12   IOSTANDARD LVCMOS33 } [get_ports { AUD_SD }]; #IO_L6P_T0_15 Sch=aud_sd

##USB-RS232 Interface
#set_property -dict { PACKAGE_PIN C4    IOSTANDARD LVCMOS33 } [get_ports { UART_TXD_IN }]; #IO_L7P_T1_AD6P_35
Sch=uart_txd_in
#set_property -dict { PACKAGE_PIN D4    IOSTANDARD LVCMOS33 } [get_ports { UART_RXD_OUT }]; #IO_L11N_T1_SRCC_35
Sch=uart_rxd_out
#set_property -dict { PACKAGE_PIN D3    IOSTANDARD LVCMOS33 } [get_ports { UART_CTS }]; #IO_L12N_T1_MRCC_35
Sch=uart_cts
#set_property -dict { PACKAGE_PIN E5    IOSTANDARD LVCMOS33 } [get_ports { UART_RTS }]; #IO_L5N_T0_AD13N_35
Sch=uart_rts

##USB HID (PS/2)
#set_property -dict { PACKAGE_PIN F4    IOSTANDARD LVCMOS33 } [get_ports { PS2_CLK }]; #IO_L13P_T2_MRCC_35
Sch=ps2_clk
#set_property -dict { PACKAGE_PIN B2    IOSTANDARD LVCMOS33 } [get_ports { PS2_DATA }]; #IO_L10N_T1_AD15N_35
Sch=ps2_data

##SMSC Ethernet PHY
#set_property -dict { PACKAGE_PIN C9    IOSTANDARD LVCMOS33 } [get_ports { ETH_MDC }]; #IO_L11P_T1_SRCC_16
Sch=eth_mdc
#set_property -dict { PACKAGE_PIN A9    IOSTANDARD LVCMOS33 } [get_ports { ETH_MDIO }]; #IO_L14N_T2_SRCC_16
Sch=eth_mdio
#set_property -dict { PACKAGE_PIN B3    IOSTANDARD LVCMOS33 } [get_ports { ETH_RSTN }]; #IO_L10P_T1_AD15P_35
Sch=eth_rstn
#set_property -dict { PACKAGE_PIN D9    IOSTANDARD LVCMOS33 } [get_ports { ETH_CRSDV }]; #IO_L6N_T0_VREF_16
Sch=eth_crsdv
#set_property -dict { PACKAGE_PIN C10   IOSTANDARD LVCMOS33 } [get_ports { ETH_RXERR }]; #IO_L13N_T2_MRCC_16
Sch=eth_rxerr
#set_property -dict { PACKAGE_PIN C11   IOSTANDARD LVCMOS33 } [get_ports { ETH_RXD[0] }]; #IO_L13P_T2_MRCC_16
Sch=eth_rxd[0]
#set_property -dict { PACKAGE_PIN D10   IOSTANDARD LVCMOS33 } [get_ports { ETH_RXD[1] }]; #IO_L19N_T3_VREF_16
Sch=eth_rxd[1]
#set_property -dict { PACKAGE_PIN B9    IOSTANDARD LVCMOS33 } [get_ports { ETH_TXEN }]; #IO_L11N_T1_SRCC_16
Sch=eth_txen
#set_property -dict { PACKAGE_PIN A10   IOSTANDARD LVCMOS33 } [get_ports { ETH_TXD[0] }]; #IO_L14P_T2_SRCC_16
Sch=eth_txd[0]
#set_property -dict { PACKAGE_PIN A8    IOSTANDARD LVCMOS33 } [get_ports { ETH_TXD[1] }]; #IO_L12N_T1_MRCC_16
Sch=eth_txd[1]
#set_property -dict { PACKAGE_PIN D5    IOSTANDARD LVCMOS33 } [get_ports { ETH_REFCLK }]; #IO_L11P_T1_SRCC_35
Sch=eth_refclk
#set_property -dict { PACKAGE_PIN B8    IOSTANDARD LVCMOS33 } [get_ports { ETH_INTN }]; #IO_L12P_T1_MRCC_16
Sch=eth_intn

##Quad SPI Flash
#set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports { QSPI_DQ[0] }]; #IO_L1P_T0_D00_MOSI_14
Sch=qspi_dq[0]
#set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 } [get_ports { QSPI_DQ[1] }]; #IO_L1N_T0_D01_DIN_14
Sch=qspi_dq[1]
#set_property -dict { PACKAGE_PIN L14   IOSTANDARD LVCMOS33 } [get_ports { QSPI_DQ[2] }]; #IO_L2P_T0_D02_14
Sch=qspi_dq[2]
#set_property -dict { PACKAGE_PIN M14   IOSTANDARD LVCMOS33 } [get_ports { QSPI_DQ[3] }]; #IO_L2N_T0_D03_14
Sch=qspi_dq[3]
#set_property -dict { PACKAGE_PIN L13   IOSTANDARD LVCMOS33 } [get_ports { QSPI_CSN }]; #IO_L6P_T0_FCS_B_14
Sch=qspi_csn
```

January 2020

Revision History:

- September 15, 2021. Corrected minor typographic errors. Emphasized that Project names do not have blank spaces.