A. Appendix 1 – ISE Quick Start Tutorial

ISE Quick Start Tutorial, adapted to ISE 14.7

Starting the ISE Software

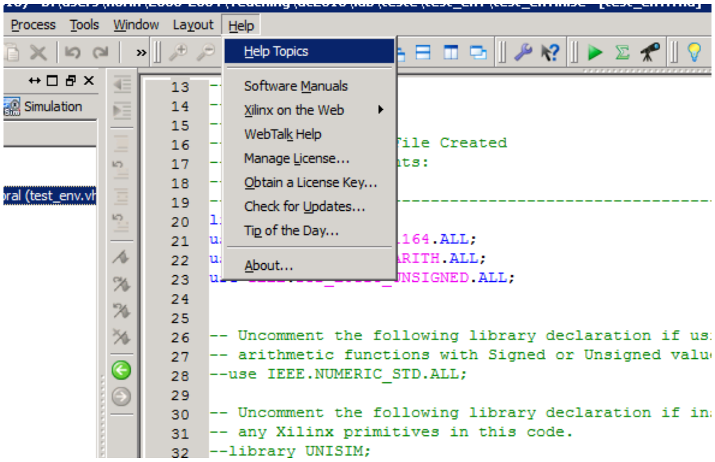
Double click the desktop icon, or go to **Start→Programs→ Xilinx Design Tools → ISE Design Suite 14.7 → ISE Design Tools → Project Navigator**

Attention (!) Be careful to use the latest version of ISE not the 9.2i version that may be installed on the computers from the lab.

Accessing Help

At any time during the tutorial, you can access online help for additional information about the ISE software and related tools.
To open Help, do either of the following:

- Press **F1** to view Help for the specific tool or function that you have selected or highlighted.
- Launch the **ISE Help Contents** from the Help menu. It contains information about creating and maintaining your complete design flow in ISE.



```
-- 
-- 
-- 
--                                    File Created
--                                    nts:
-- 
--                                    ---------------------------------
l
u                                     164.ALL;
u                                     ARITH.ALL;
u                                     UNSIGNED.ALL;


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

-- Uncomment the following library declaration if in
-- any Xilinx primitives in this code.
--library UNISIM;
```

Figure A-1: ISE Help Topics

Create a New Project

Create a new ISE project, which will target the FPGA device on the Basys 1 or Basys 2 development board, Spartan 3E.
To create a new project:

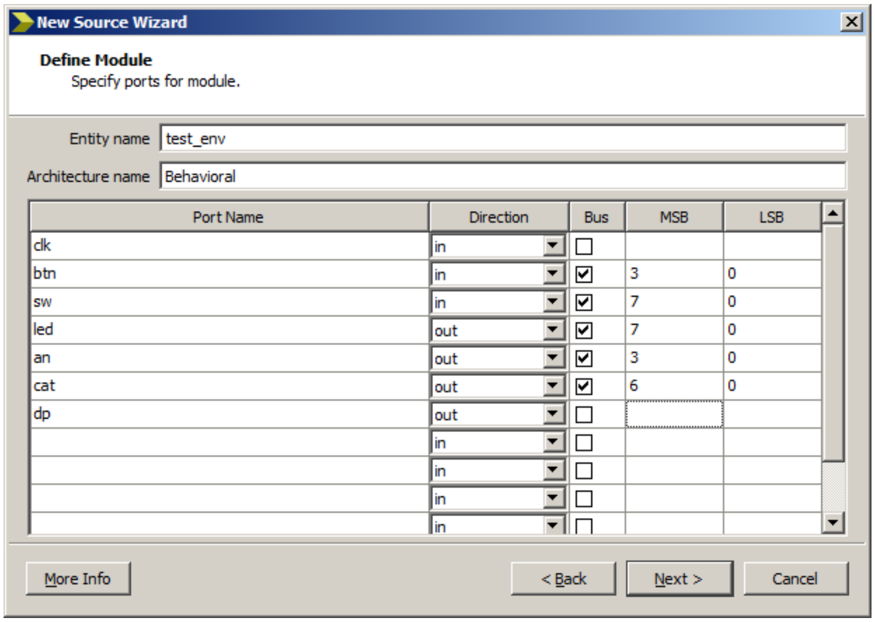
1. Select **File → New Project...** The New Project Wizard appears.
2. Type **test_env** in the Entity Name field.

3. Enter or browse to a location (directory path) for the new project (remember the laboratory rules). A test_env subdirectory is created automatically.
4. Verify that **HDL** is selected from the Top-Level Source Type list.
5. Click **Next** to move to the device properties page.
6. Fill in the properties in the table as shown below:
   - Product Category: **All**
   - Family: **Spartan 3E**
   - Device: **XC3S100E**
   - Package: **TQ144 (for Basys 1) / CP132 (for Basys 2)**
   - Speed Grade: **-4**
   - Top-Level Module Type: **HDL**
   - Synthesis Tool: **XST (VHDL/Verilog)**
   - Simulator: **ISim (VHDL/Verilog)**
   - Preferred language: **VHDL**
   - Leave the default values in the remaining fields.

Creating a VHDL Source

Create a VHDL source file for the project as follows:

1. Click the menu **Project/New Source**.
2. Select VHDL Module as the source type.
3. Type in the file name that you want to create. For example, “test_env”.
4. Verify that the Add to project checkbox is selected.
5. Click Next.
6. Declare the ports for your design by filling in the port information as in the following figure. **These ports are particularly defined for the Basys 1 or Basys 2 board, being enough for the majority of the laboratory designs for this semester.**



New Source Wizard

Define Module

Specify ports for module.

Entity name: test_env

Architecture name: Behavioral

Port Name	Direction	Bus	MSB	LSB
clk	in	□		
btn	in	☑	3	0
sw	in	☑	7	0
led	out	☑	7	0
an	out	☑	3	0
cat	out	☑	6	0
dp	out	□		
	in	□		
	in	□		
	in	□		
	in	□		

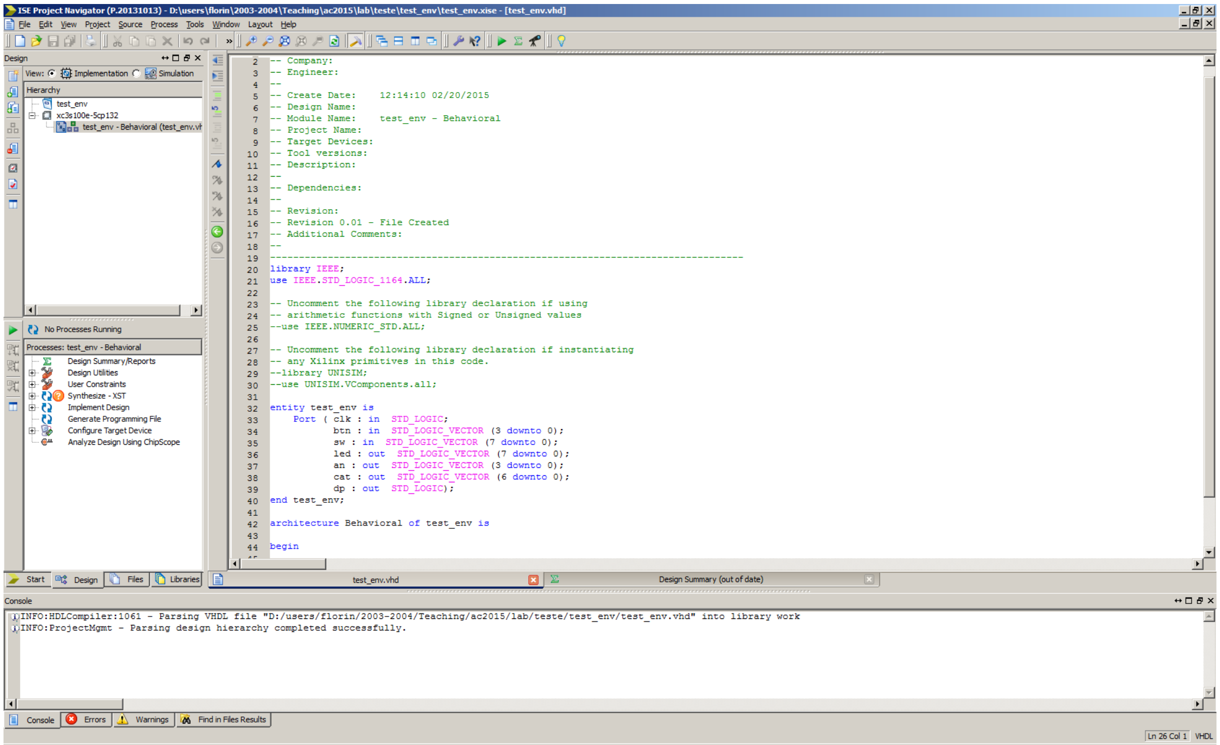
Figure A-2 Port definition through the Xilinx Interface

7. Click **Next** (re-verify the summary of the port declarations), and then **Finish** to complete the new source file template.

The source file containing the entity *test_env* and its architecture is displayed in the ISE environment, and in the **Hierarchy** tab appears as **Top Module** for the current design.

Remember, in projects containing multiple source files, if one accidentally changes the top module entity, you can reset it as a top module by right click on a source in **Hierarchy**, and select **Set as Top Module**.

Attention: the parent of the test_env entity in the hierarchy is formed by the properties of the FPGA target device. For Basys, one must see **xc3s100e-4tq144.** If it does not coincide, this means that you have probably skipped step 6 from **Creating a VHDL Source**. Double click on the parent and enter the target device properties.


ISE Project Navigator (P.20131013) - D:\users\florin\2003-2004\Teaching\ac2015\lab\teste\test_env\test_env.xise - [test_env.vhd]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
test_env
xc3s100e-5cp132
test_env - Behavioral (test_env.vhd)
No Processes Running
Processes: test_env - Behavioral
Design Summary/Reports
Design Utilities
User Constraints
Synthesize - XST
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using ChipScope
Start
Design
Files
Libraries
test_env.vhd
Design Summary (out of date)
Console
Errors
Warnings
Find in Files Results
Ln 26 Col 1
VHDL

```
-- Company:
-- Engineer:
--
-- Create Date:    12:14:10 02/20/2015
-- Design Name:
-- Module Name:    test_env - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
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 test_env is
    Port ( clk : in  STD_LOGIC;
           btn : in  STD_LOGIC_VECTOR (3 downto 0);
          sw : in  STD_LOGIC_VECTOR (7 downto 0);
           led : out  STD_LOGIC_VECTOR (7 downto 0);
          an : out  STD_LOGIC_VECTOR (3 downto 0);
           cat : out  STD_LOGIC_VECTOR (6 downto 0);
           dp : out  STD_LOGIC);
end test_env;

architecture Behavioral of test_env is

begin
```

```
INFO:HDLCompiler:1061 - Parsing VHDL file "D:/users/florin/2003-2004/Teaching/ac2015/lab/teste/test_env/test_env.vhd" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
```

Figure A-3: The new ISE Project

Make shore that the following libraries are included in the source file header:

use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

If they are missing from any VHDL file, include them!

Using Language Templates (VHDL) – optional (*You will probably have to use this in the future…*)

Language Templates includes VHDL synthesizable examples that you can use in your designs. The “Light Bulb” takes you directly to Language Templates tab” or you can do the following”

1. Place the cursor under the begin statement of your architecture.
2. Open Language Templates by selecting the menu **Edit → Language Templates…**
3. Navigate in the hierarchy “+”, to the coding examples: **VHDL → Synthesis Constructs → Coding Examples → …**
4. Select the desired component in the hierarchy, then right click → **Use in File**. This step will copy the model code to your source file at the place of your cursor.
5. Close the **Language Templates**.
6. Change the signal names so that they will match the signals in your entity.

Editing the VHDL Source Code

1. Add component and/or signal declarations between the architecture and the begin statements.
2. Add the rest of the code (component instantiation, behavioral description, etc.) after the begin statement and before the end statement.
3. For the first example add the following statements after begin.

```
led <= sw;
an  <= btn;
cat <= (others=>'0');
dp  <= '0';
```

4. Save the file by selecting **File → Save or Ctrl + S**.
5. Select the top-level entity in the Hierarchy tab: *test_env*.
6. Verify that your VHDL syntax is correct: in the **Processes** zone: **Synthesize – XST → Check Syntax → Run**
7. Correct the errors if they appear in the bottom part of the ISE environment. Start from the top with the first error.
8. Synthesize your design: double click **Synthesize – XST**
9. View the resulting circuit: double click **Synthesize – XST → View RTL Schematic.** In the next dialog be sure to select the second variant (*Start with a schematic of the top-level block*), press **OK**. The top-level entity will appear. Double click to view its internal organization. **You should recognize at least a part of the declared entity.** This is a first method to verify that your code is correct and implements the desired circuits.

You have now created the VHDL source for the “test_env” project with no errors.
Note: You can also create a UCF file for your project by selecting **Project → Create New Source**.

Assigning Pin Location Constraints

Specify the pin locations for the ports of the design so that they are connected correctly on the DDB. You can Edit the User Constraints File (*.ucf) manually (**Users Constraints → Edit Constraints (Text)**). You can find the user constraints file for the Basys 1 Board here and for the Basys 2 Board here. Download the file and add it to your design. Open the constraints file and see the syntax for every port (net).

For the future, you can add new ports to the constraints file.

Implement Design and Verify Constraints

Implement the design and verify that it meets all constraints.

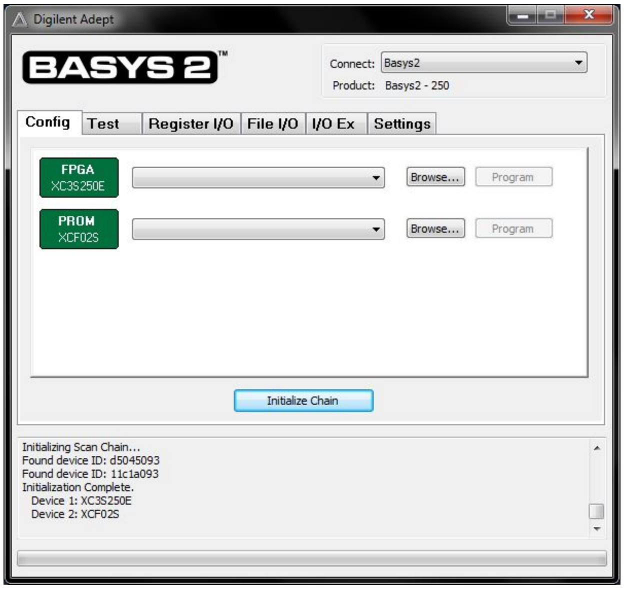
1. Double-click the **Implement Design** process in the Processes tab.
2. Notice that after Implementation is complete, the Implementation processes have a green check mark next to them indicating that they completed successfully without Errors or Warnings. If there are errors or warnings, you can correct them.
3. Open **Design Summary/Reports**. Analyze the reports of your design (Summary, Timing Constraints, etc.). In the next designs, these reports will be relevant.

Generate Programming File

1. Before generating the programming file, you must set the start-up clock option to JTAG clock: **Generate Programming File → Properties → Startup Options → FPGA Start-Up Clock → JTAG Clock.**
2. Generate the programming file: double click **Generate Programming File.** The bit file for the DDB configuration is created.

If you notice that one or more processes have an orange question mark next to them, it indicates that they are out-of-date with one or more of the design files. You will have to re-run these processes.

If there are no errors at this time the file “test_env.bit” should be in the project folder.



Connect: Basys2
Product: Basys2 - 250

```
Initializing Scan Chain...
Found device ID: d5045093
Found device ID: 11c1a093
Initialization Complete.
  Device 1: XC3S250E
  Device 2: XCF02S
```

Figure A-4: Digilent Adept tool

Download Design to the Spartan™-3E Demo Board → Basys (1 or 2)

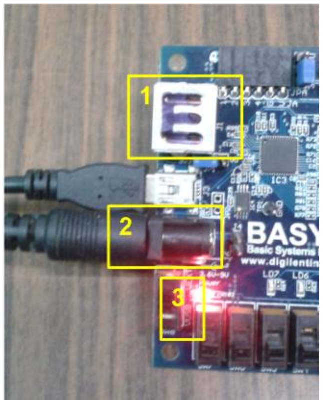
If you encounter problems during the programming of the board please go to the end of this tutorial (after the following figure).

1. Connect the Basys (1 or 2) board to the USB port.
2. Start the Adept Tool from Adept programming software: Start → Programs → Digilent → Adept (Figure A-4).
3. Press the Initialize Chain button
4. Browse for the project's bit file.
5. Program the FPGA device.

Possible problems when connecting the board and solutions:

Problem: The Basys (1 or 2) board is not recognized.
Solutions (Start in order and restart Adept after every fail, ask the TA to assist you):

a) Try a different USB port (front or rear of the computer). If a driver install process initiates, call your TA. You will need administrative privileges.
b) Verify that the board does not require external power.
   - If it does not require external power (no "E" sign – see the figure below position 1), make shore that the switch (Position 3 in the image) is in the VUSB position
   - If it requires external power, do the following
     - ➢ Use a 3.3 V power supply in the external power socket (position 2 in the image)
     - ➢ Move the switch (position 3 in the image) to the VEXT position



c) Try a new programming cable
d) Try a new board (report this to your TA)
e) Change the workstation