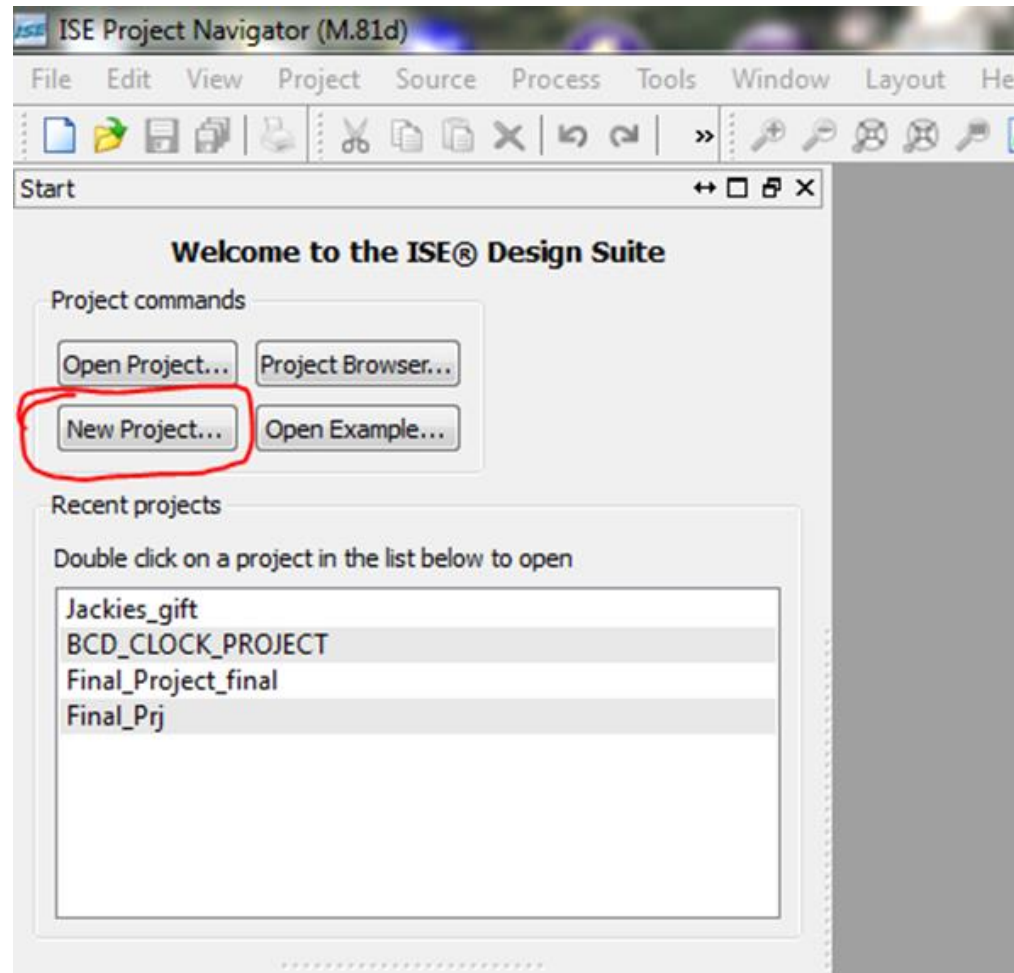


Module 8.2

Xilinx ISE Project

Creating a Project



New Project Wizard

New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name: Full_Adder_2

Location: C:\Users\Jordan\Documents\Udemy\Lab_1\Full_Adder_2

Working Directory: C:\Users\Jordan\Documents\Udemy\Lab_1\Full_Adder_2

Description: Udemy Lab 1
Full_Adder_2
Implementation of a full adder

Select the type of top-level source for the project

Top-level source type: HDL

More Info Next Cancel

Project Settings

New Project Wizard

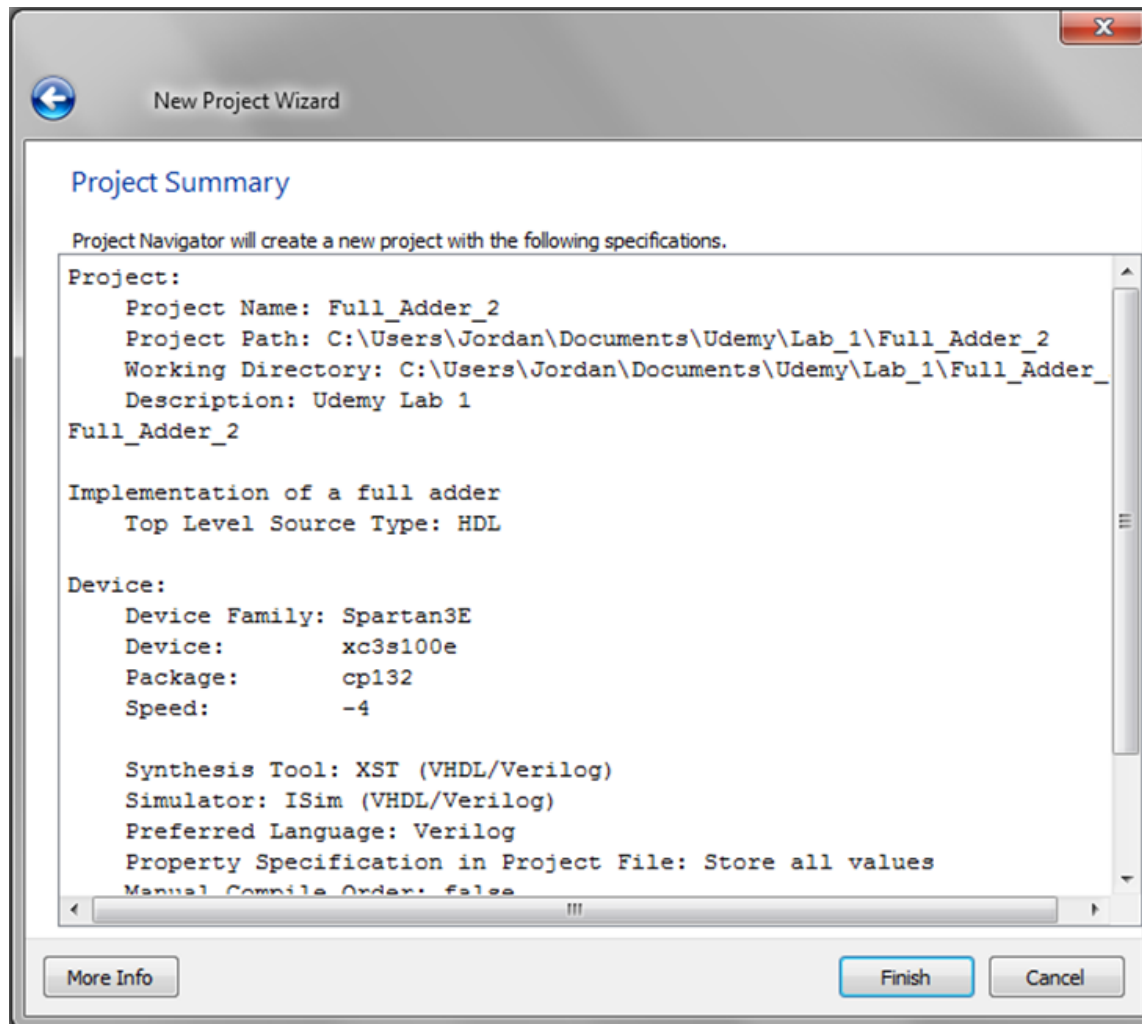
Project Settings

Specify device and project properties.
Select the device and design flow for the project

| Property Name | Value |
|--|--------------------------|
| Product Category | All |
| Family | Spartan3E |
| Device | XC3S100E |
| Package | CP132 |
| Speed | -4 |
| | |
| Top-Level Source Type | HDL |
| Synthesis Tool | XST (VHDL/Verilog) |
| Simulator | ISim (VHDL/Verilog) |
| Preferred Language | Verilog |
| Property Specification in Project File | Store all values |
| Manual Compile Order | <input type="checkbox"/> |
| VHDL Source Analysis Standard | VHDL-93 |
| | |
| Enable Message Filtering | <input type="checkbox"/> |

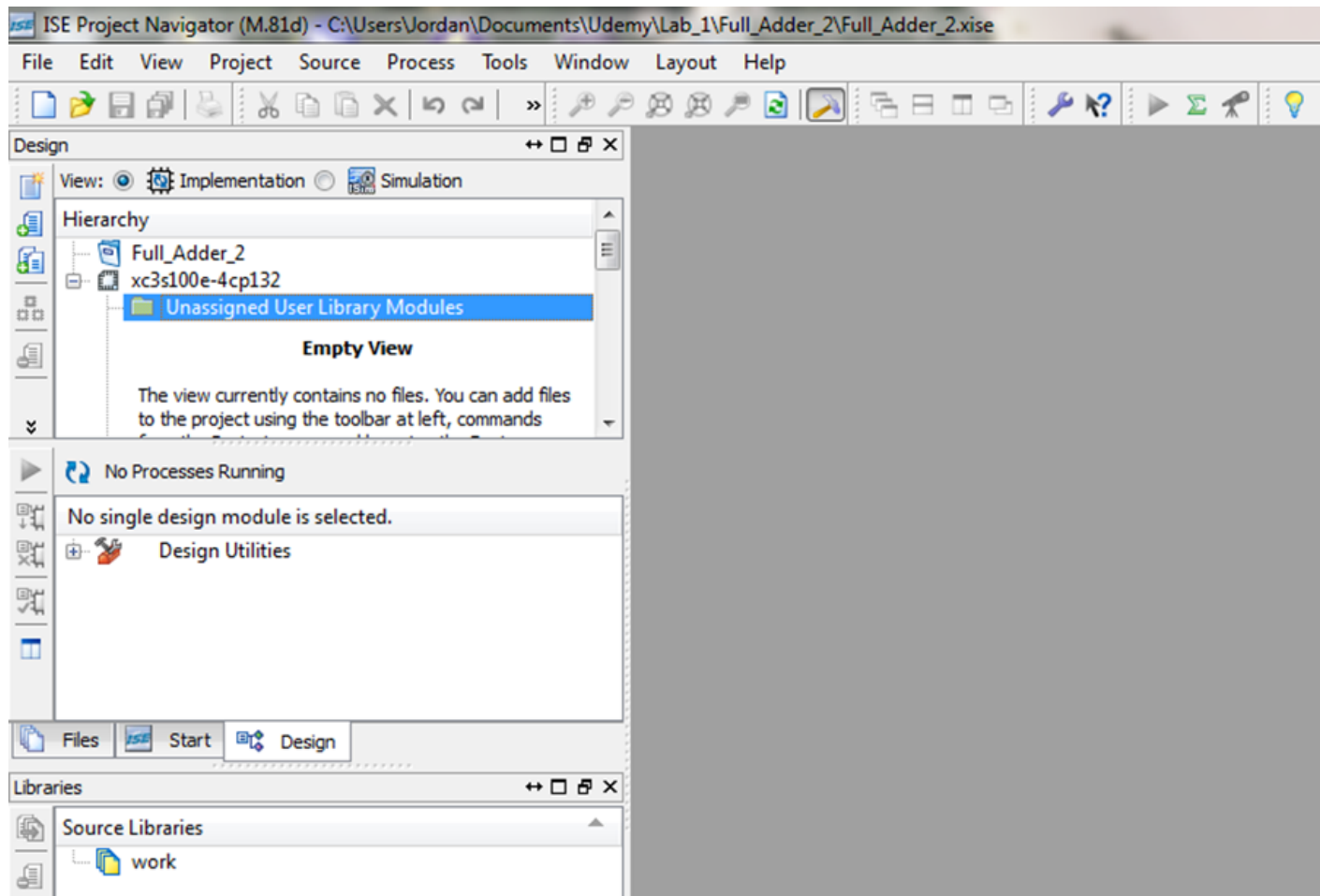
More Info Next Cancel

Project Summary

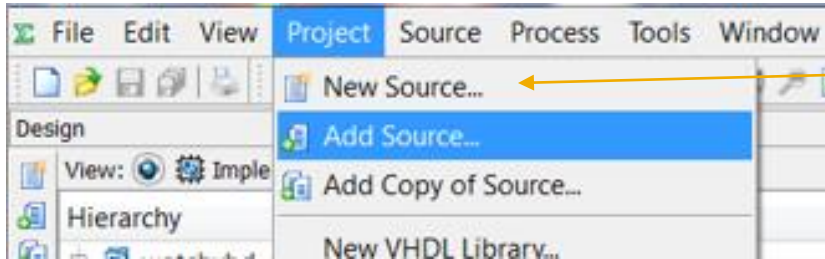


Add Design file

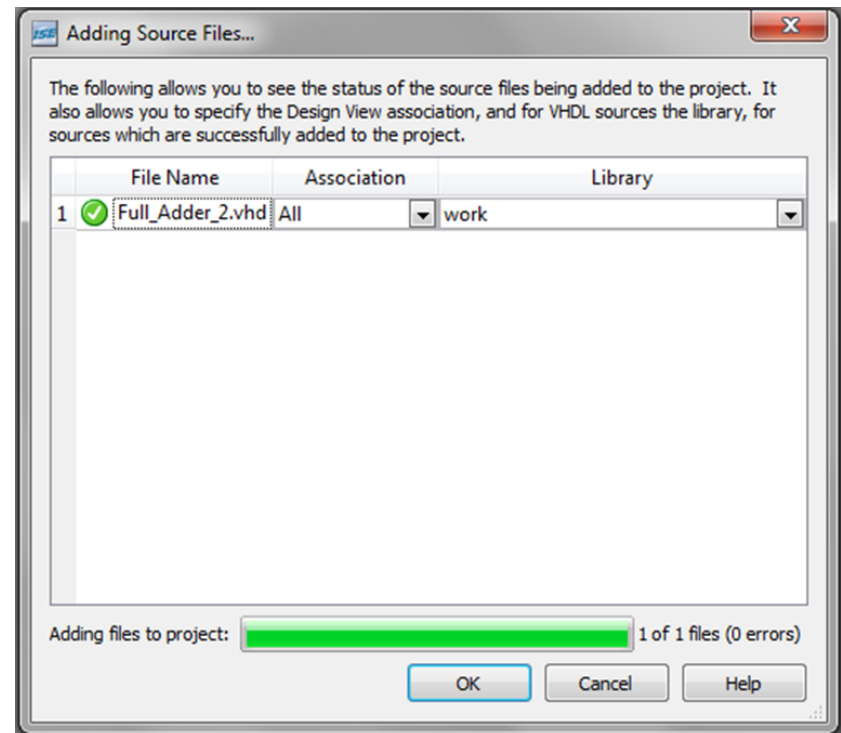
Note: Implementation radio button is selected



Add Design file



Select "New Source" if you
Want to code your VHDL file
Inside of Xilinx ISE



Full_Adder_2 example

The image shows the ISE Project Navigator interface with the following components and annotations:

- Name of Project:** An arrow points to the project name "Full_Adder_2" in the Hierarchy window.
- Entity of VHDL file:** An arrow points to the entity "Full_Adder_2 - behavior (Full_Adder_2.vhd)" in the Hierarchy window.
- VHDL editing area:** An arrow points to the VHDL code in the main editor window.
- Indicates top level:** An arrow points to the "Full_Adder_2" entity in the Hierarchy window.

The VHDL code in the editing area is as follows:

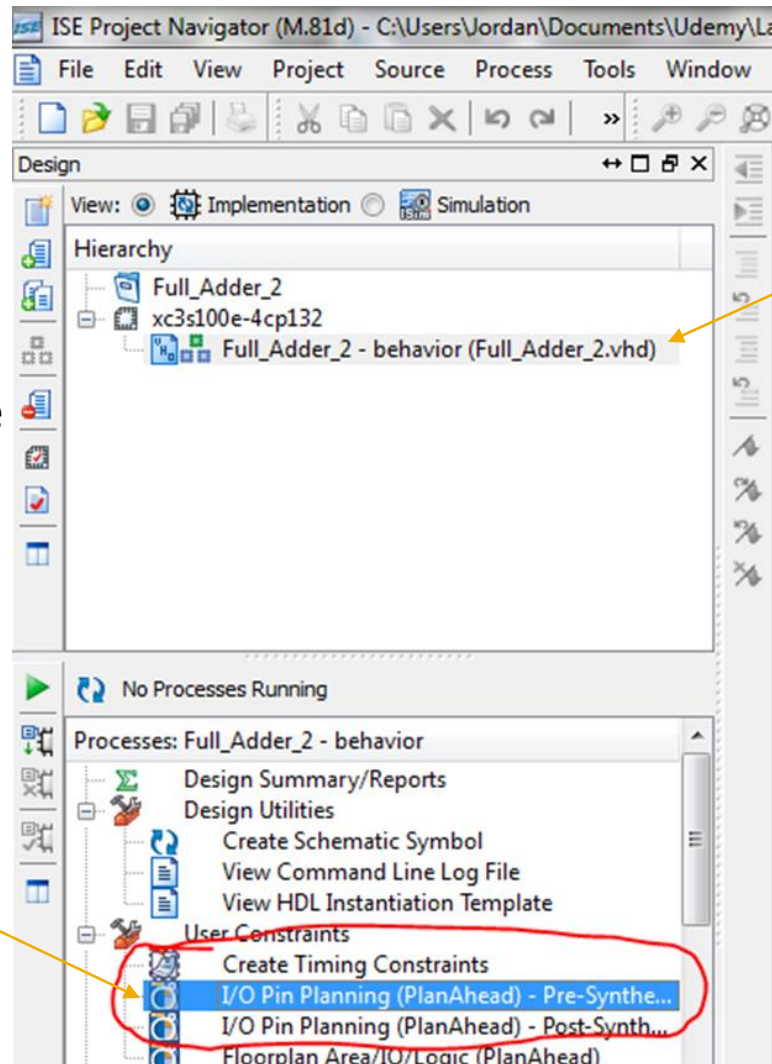
```
1  -- Behavioral Full Adder
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity Full_Adder_2 is
6  port (
7      S          : out std_logic;
8      C_out      : out std_logic;
9      x          : in  std_logic;
10     y          : in  std_logic;
11     C_in       : in  std_logic);
12 end Full_Adder_2;
13
14 architecture behavior of Full_Adder_2 is
15
16     signal inputs  : std_logic_vector(2 downto 0);
17     signal outputs : std_logic_vector(1 downto 0);
18
19     begin
20
21     -- Combine inputs & outputs into std_logic_vector
22     inputs <= C_in & x & y;
23     C out <= outputs(1);
```

Map the I/O

NOTE:

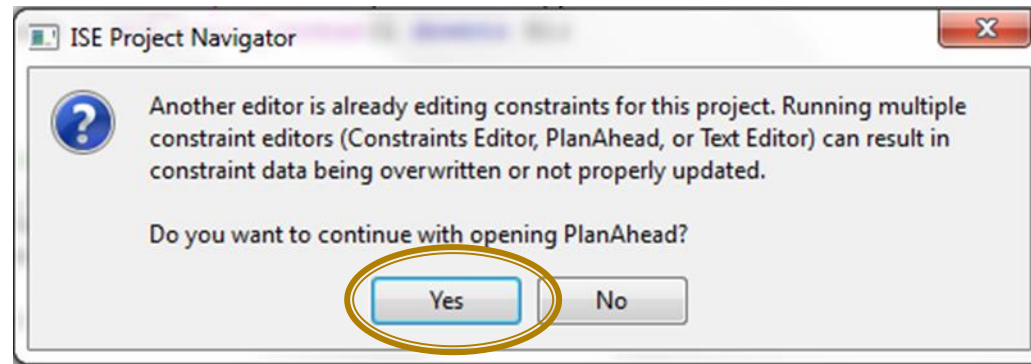
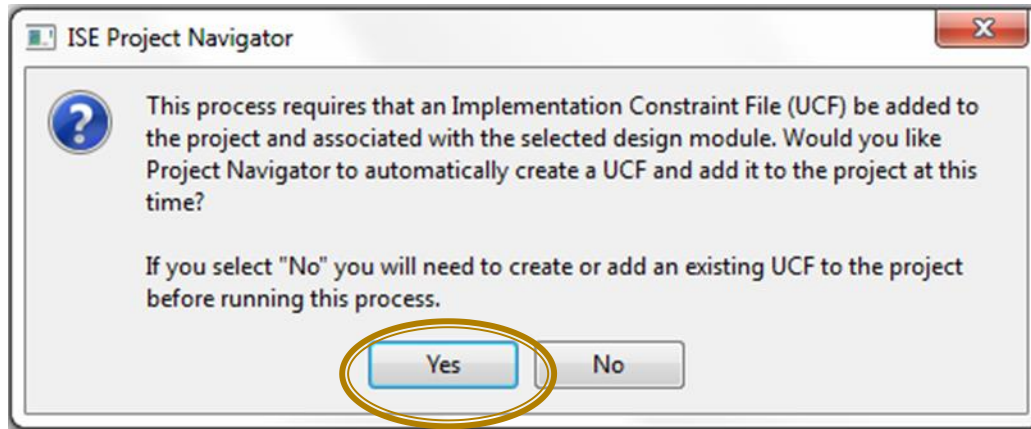
For the labs in this course I will be providing you with the UCF (user constraints file). You will just need to go to Project → Add source and then select the *.ucf file given to you

2) Right click. Then select run



1) Be sure to click on the top level VHDL file to activate it

Map the I/O



Plan Ahead

The screenshot displays the Xilinx ISE software interface for a project named "Full_Adder_2". The main window is titled "RTL Design - Full_Adder_2.ucf (target) | xc3s100ecp132-4". The "I/O Ports" window is highlighted with a yellow circle and contains the following table:

| Name | Dir | Neg Diff Pair | Site | B |
|------------------|--------|---------------|------|---|
| All ports (5) | | | | |
| Scalar ports (5) | | | | |
| C_in | Input | | | |
| C_out | Output | | | |

The "I/O Port Properties" window shows the "General" tab with the "Name" field set to "C_in".

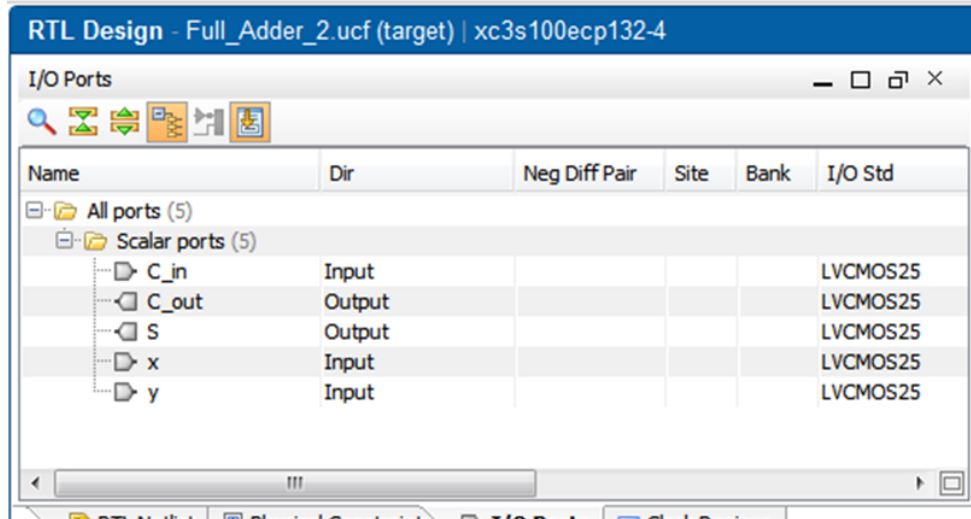
The "Package" view shows a grid of pins (A-P, 1-14) with various colored indicators. The "Device" view shows a schematic of the device with two "CPU" blocks.

The "Tcd Console" window displays the following log messages:

```
INFO: [HD-UCFReader 0] Parsing UCF File : .\Full_Adder_2.ucf
INFO: [HD-UCFReader 1] Finished Parsing UCF File : .\Full_Adder_2.ucf
INFO: [HD-LIB 0] Reading timing library C:/Xilinx/12.4/ISE_DS/PlanAhead/parts/xilinx/spartan3e/spartan3e/spartan3e-4.lib .
INFO: [HD-LIB 1] Done reading timing library C:/Xilinx/12.4/ISE_DS/PlanAhead/parts/xilinx/spartan3e/spartan3e/spartan3e-4.1
```

The bottom of the interface shows the "Tcd Console" tab selected, with a text input field "Type a Tcl command here" and buttons for "Elaboration Messages" and "Reports". The status bar at the bottom right indicates "ISE Integration Flow".

Plan Ahead



Note:

The port members of the Entity are what show up as inputs and outputs in the Plan ahead tool

```
1  -- Behavioral Full Adder
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity Full_Adder_2 is
6  port (
7      S          : out std_logic;
8      C_out      : out std_logic;
9      x          : in  std_logic;
10     y          : in  std_logic;
11     C_in       : in  std_logic);
12 end Full_Adder_2;
13
14 architecture behavior of Full_Adder_2 is
15
16 signal inputs  : std_logic_vector(2 downto 0);
17 signal outputs : std_logic_vector(1 downto 0);
18
```

Plan Ahead

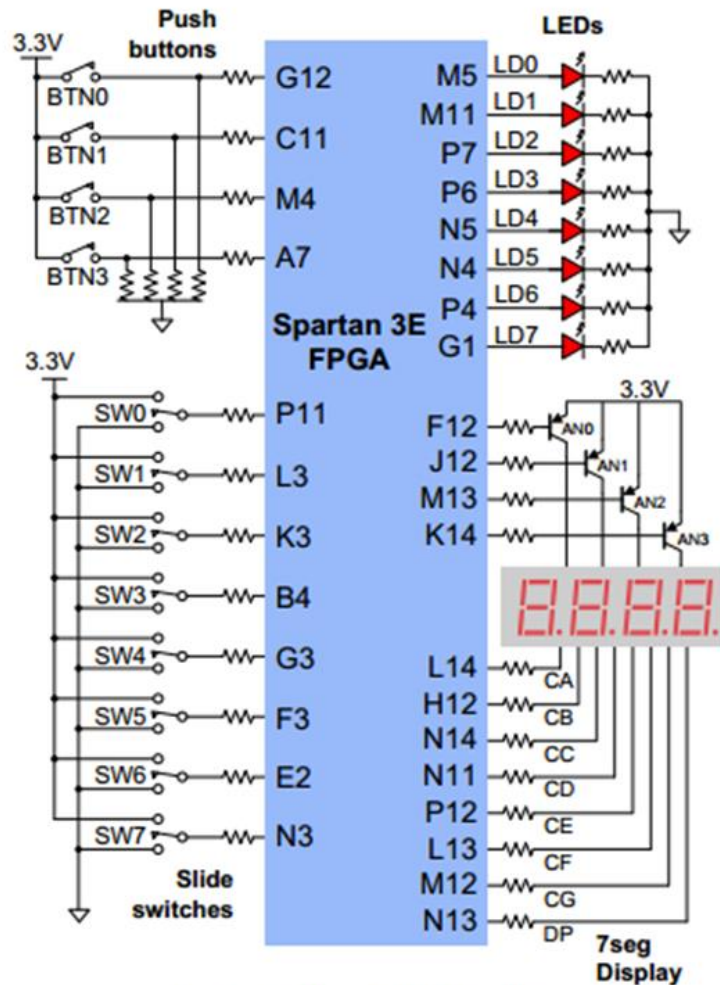


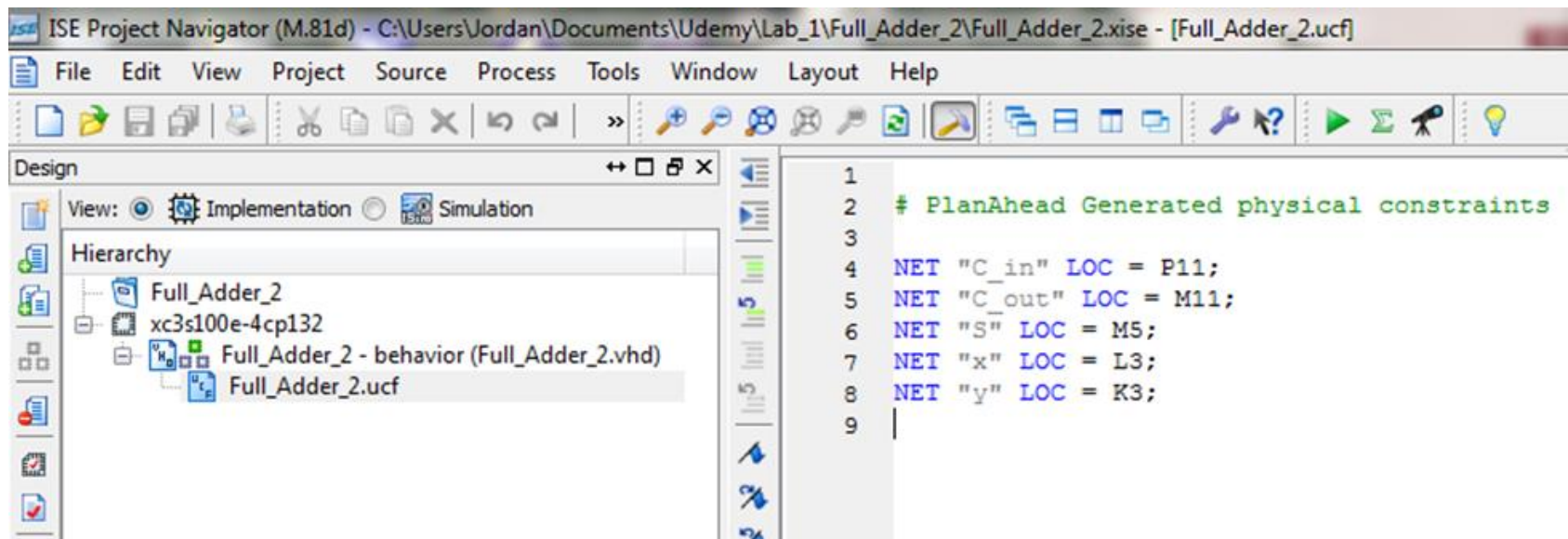
Figure 6. Basys2 I/O circuits

The screenshot shows the I/O Ports window in the RTL Design tool. The window title is "RTL Design - Full_Adder_2.ucf (target) | xc3s100ecp132-4". The I/O Ports table is as follows:

| Name | Dir | Neg Diff Pair | Site | Bank | I/O Std |
|------------------|--------|---------------|------|------|----------|
| All ports (5) | | | | | |
| Scalar ports (5) | | | | | |
| C_in | Input | | P11 | 2 | LVCMOS25 |
| C_out | Output | | M11 | 2 | LVCMOS25 |
| S | Output | | M5 | 2 | LVCMOS25 |
| x | Input | | L3 | 3 | LVCMOS25 |
| y | Input | | K3 | 3 | LVCMOS25 |

The I/O Port Properties window is also visible at the bottom, showing the selected port 'x'.

UCF – User Constraints File

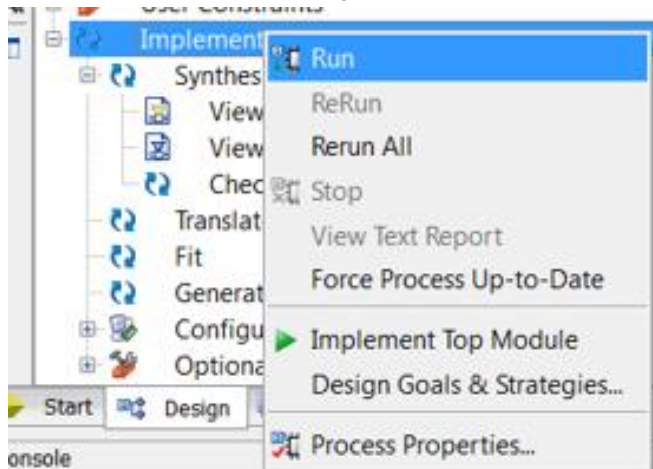


The screenshot shows the ISE Project Navigator interface. The title bar indicates the project is 'Full_Adder_2.ucf'. The menu bar includes File, Edit, View, Project, Source, Process, Tools, Window, Layout, and Help. The Design window is active, showing a Hierarchy view of the project. The hierarchy includes 'Full_Adder_2' (xc3s100e-4cp132) containing 'Full_Adder_2 - behavior (Full_Adder_2.vhd)' and 'Full_Adder_2.ucf'. The main editor displays the following UCF code:

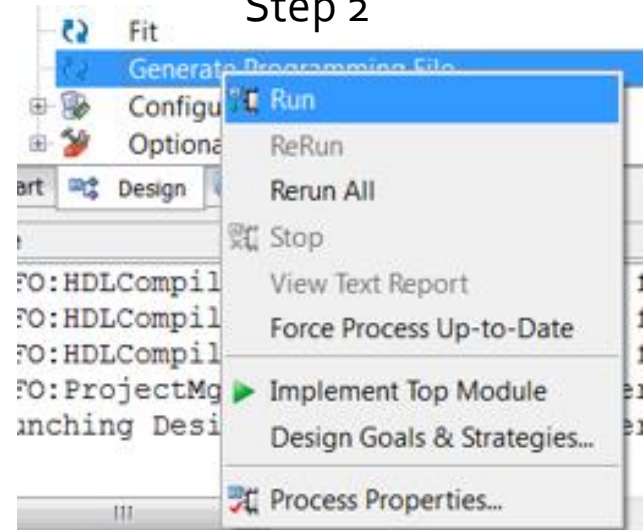
```
1  
2 # PlanAhead Generated physical constraints  
3  
4 NET "C_in" LOC = P11;  
5 NET "C_out" LOC = M11;  
6 NET "S" LOC = M5;  
7 NET "x" LOC = L3;  
8 NET "y" LOC = K3;  
9
```

Implement Design & Generate Programming File

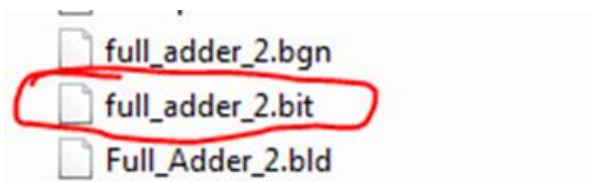
Step 1



Step 2



Final file we are wanting



Final Report

Full_Adder_2 Project Status (01/31/2015 - 20:29:14)

| | | | |
|-------------------------|---|------------------------------|---|
| Project File: | Full_Adder_2.xise | Parser Errors: | No Errors |
| Module Name: | Full_Adder_2 | Implementation State: | Programming File Generated |
| Target Device: | xc3s100e-4cp132 | • Errors: | No Errors |
| Product Version: | ISE 12.4 | • Warnings: | No Warnings |
| Design Goal: | Balanced | • Routing Results: | All Signals Completely Routed |
| Design Strategy: | Xilinx Default (unlocked) | • Timing Constraints: | |
| Environment: | System Settings | • Final Timing Score: | 0 (Timing Report) |

Device Utilization Summary [-]

| Logic Utilization | Used | Available | Utilization | Note(s) |
|--|------|-----------|-------------|---------|
| Number of 4 input LUTs | 2 | 1,920 | 1% | |
| Number of occupied Slices | 1 | 960 | 1% | |
| Number of Slices containing only related logic | 1 | 1 | 100% | |
| Number of Slices containing unrelated logic | 0 | 1 | 0% | |
| Total Number of 4 input LUTs | 2 | 1,920 | 1% | |
| Number of bonded IOBs | 5 | 83 | 6% | |
| Average Fanout of Non-Clock Nets | 1.60 | | | |

Performance Summary [-]

| | | | |
|----------------------------|---|---------------------|-------------------------------|
| Final Timing Score: | 0 (Setup: 0, Hold: 0) | Pinout Data: | Pinout Report |
| Routing Results: | All Signals Completely Routed | Clock Data: | Clock Report |
| Timing Constraints: | | | |

Summary

- We can use Plan Ahead inside of Xilinx ISE to create our UCF (User Constraints File)
- You import VHDL files into Xilinx ISE or you can create them using the integrated VHDL editor
- Xilinx ISE is the tool we will be using to generate a programming file to load onto our FPGA (*.bit file)