TUTORIAL ON USING XILINX ISE DESIGN SUITE 14.6:
Design Entry using Schematic Capture
"Design of a Full Adder Circuit"
Using Spartan-6 (NEXYS 3 Board)

Shawki Areibi

September 8, 2024

1 Introduction

The objective of this tutorial is to familiarize the student with the Xilinx ISE Design Suite 14.6. This software package provides the digital designer with a wide variety of software tools. These allow digital circuits to be designed and simulated before they are implemented in hardware. CAD tools such as these are essential for designing complex digital circuits. They remove much of the drudgery from the design process and allow the designer to concentrate on the creative part of the design. In this tutorial you will learn the following topics:

1. How to use the Xilinx Foundation Design Suite.
2. How to enter your design using Schematic Capture, and creating a symbol.
3. How to design a Full Adder by reusing the design of a Half Adder.
4. How to synthesize, implement your design.
5. How to reconfigure the FPGA with your design.

2 Xilinx ISE 14.6 Tool

The ISE Software controls all aspects of the design flow. Through the Project Navigator interface, you can access all the design entry (Schematic, VHDL) and design implementation tools. You can also access the files and documents associated with your project. By default, the Project Navigator interface is divided into three panel sub-windows:

1. Design Panel, which provides access to the View, Hierarchy, and Process panes. The View pane radio buttons enable you to view the source modules associated with the **Implementation** or **Simulation** Design View in the Hierarchy pane. If you select Simulation, you must select a simulation phase from the drop-down list.
2. Console Panel, which provides all standard output from processes run from the Project Navigator. It displays errors, warnings, and information messages.
3. Workspace panel, is where design editors, viewers, and analysis tools open. These include ISE Text Editor, Schematic Editor, Constraint Editor, Design Summary/Report Viewer.

In this tutorial you will explore the functionality of only a sub-set of the Project Navigator capabilities.

3 Design of Full Adder Circuit

Before designing a schematic, you must derive Boolean equations for your circuit. Once you have derived these equation, you may verify and implement them using the Xilinx software.
Binary adder circuits are present in every microprocessor as well as in other circuits. They are used to take the arithmetic sum of two binary numbers. They are also the building blocks for other arithmetic circuits such as subtracters and multipliers. The simplest adder circuit is Half Adder which has only two inputs. A full adder performs the same operation as a half adder except that it also considers the carry bit generated by the preceding adder circuit. This enables full adders to be cascaded in order to create larger adders.

Specifications

The full adder will take two binary digits and a carry-in bit and produce a sum bit and a carry-out bit.

Define inputs and outputs

Inputs: two binary digits (A and B) and a carry-in bit (Ci).
Outputs: a sum bit (S) and a carry-out bit (Co).

Create a truth table

Inputs			Outputs	
A	B	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Derive Boolean equations

As can be seen in the truth table, the carry bit is only set when both A and B are set or when one of these and the carry-in bit is set. The sum bit is set when there is an odd number of 1s across the inputs. This function can be implemented by cascading two 2 input Exclusive-OR function gates.

$Co = AB + AC_i + BC_i = AB + C_i(A \oplus B)$
$S = \bar{A}\bar{B}C_i + \bar{A}B\bar{C}_i + A\bar{B}\bar{C}_i + ABC_i = (A \oplus B) \oplus C_i$

This circuit is constructed from two half adders and an OR gate. We will first design a half adder create a symbol out of it and then reuse the half adder to create our final full adder circuit. If you have already designed a half adder previously you can skip the design and schematic capture of Half Adder.

3.1 Design of Half Adder Circuit

In this section we will implement the half-adder circuit.

Specifications

The half adder will take in two binary digits in order to produce a sum bit and carry bit.

Define inputs and outputs

Inputs: two bits (A and B).
Outputs: a sum bit (S) and a carry bit (C).

Create a truth table

Inputs		Outputs	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Derive Boolean equations

As we can see from the truth table, the carry bit is only set when both A and B are set. This is an AND operation. The sum is set when only one of the two inputs are set. This is an Exclusive-OR operation.

Therefore,
$C = AB$
$S = \overline{A}B + A\overline{B} = A \oplus B$

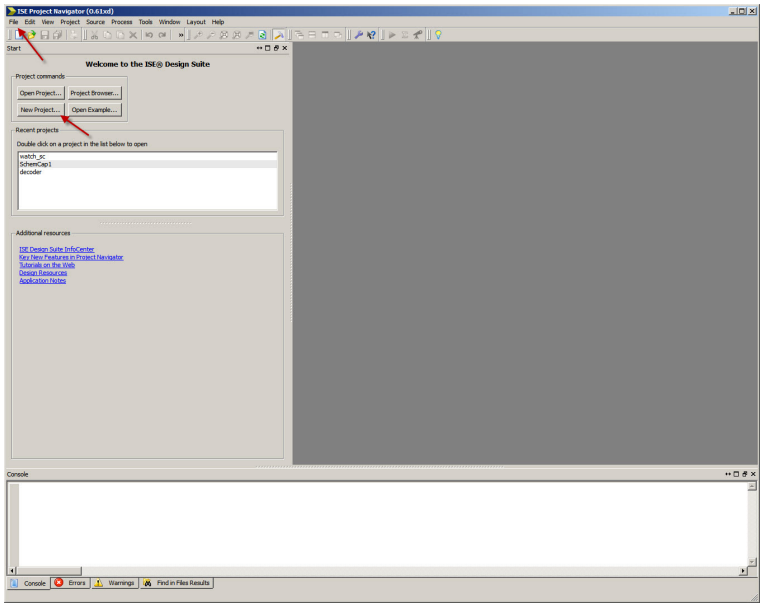
We are now ready to enter our design using a CAD tool.

3.2 Starting a new project

To enter a new design we must first start a new project in the Project Navigator; the main interface for the Xilinx ISE Suite.

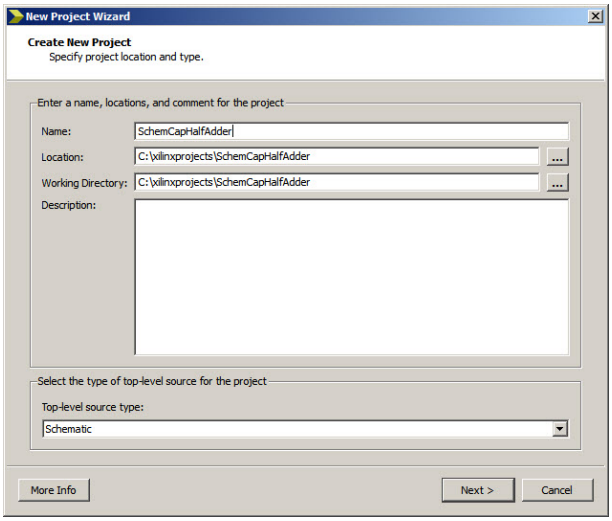
1. Load the Project Navigator from the Start → **All Programs** → **Xilinx ISE Design Suite 14.6** → **ISE Design Tools** → **Project Navigator**.

2. The Project Navigator window will appear.

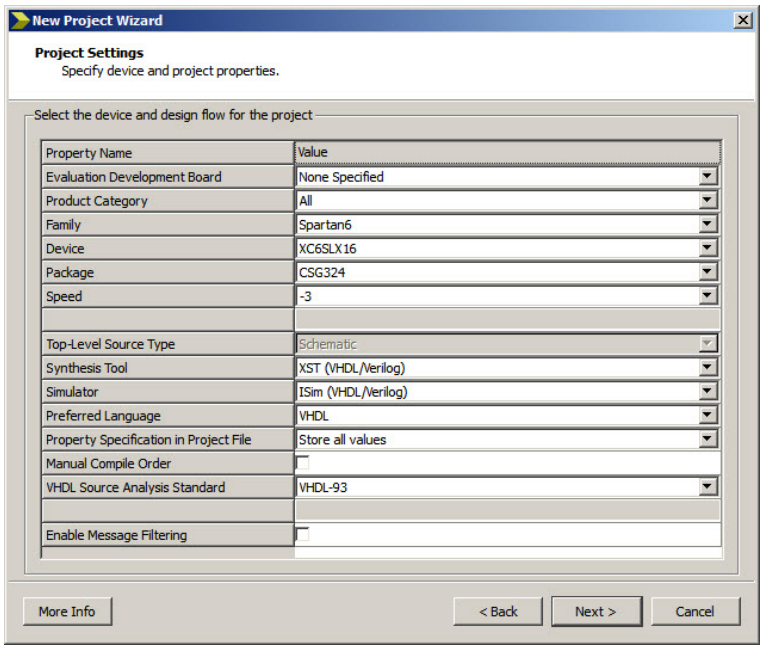


3. Select either **File** → **New Project** or click on the **New Project** tab.

4. The New Project Wizard dialog box will appear. Specify the directory in which you want to store the project in and name the project "SchemCapHalfAdder". In the **Top-Level Source Type** section select **Schematic** and click **Next**.



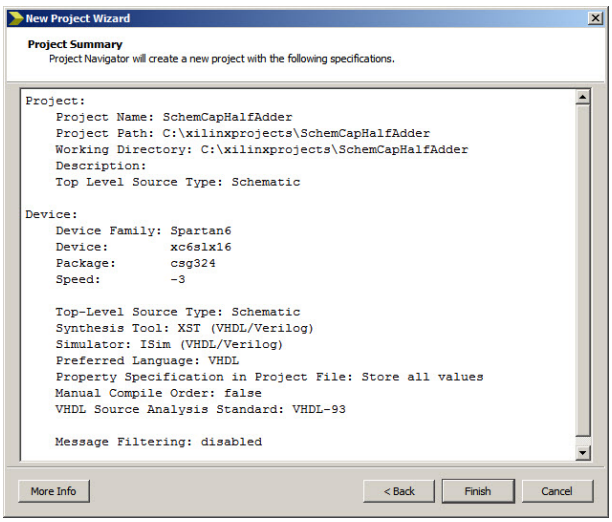
5. Another New Project Wizard dialog box will appear prompting you for device, synthesis and simulation settings for the project.



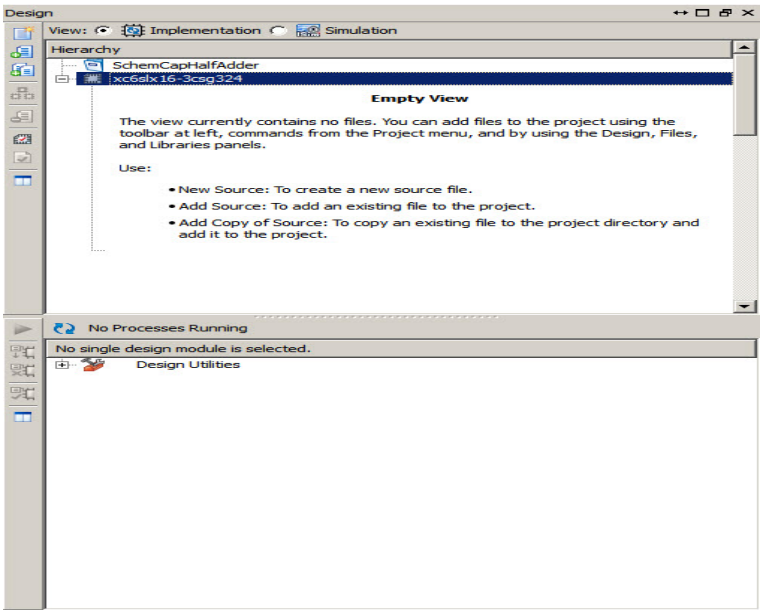
In this dialog box verify the following settings:

- **Family** → **Spartan6**.
- **Device** → **XC6SLX16**.
- **Package** → **CSG324**.
- **Speed Grade** → **-3**.
- **Synthesis Tool** → **XST (VHDL/Verilog)**.
- **Simulator** → **ISim (VHDL/Verilog)**.
- **Generated Simulation Language** → **VHDL**.

6. If the information is correct click **Next**. The following window will appear and click **Finish**.

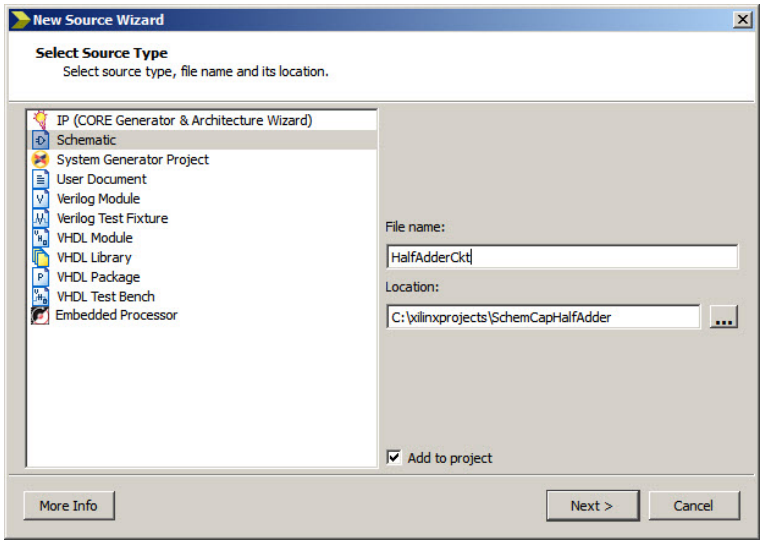


7. The next dialog box will allow you to create a new source file and add it to the project.



Click on the  **New Source** button.

8. In the new dialog box that appears, select **Schematic** from the list of file types and enter "HalfAdderCkt" as the file name.



The default location is the current project directory and can be left as is. Ensure the **Add to Project** box is selected and click the **Next** button.

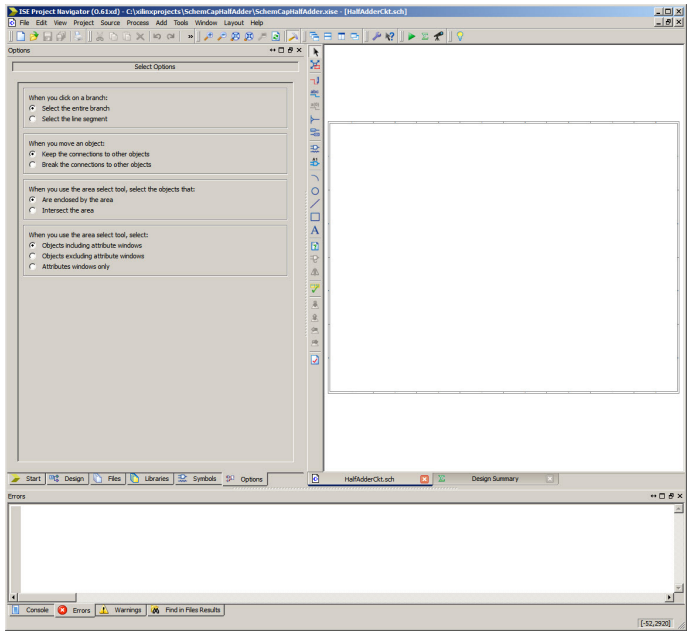
Verify the information in the next dialog box and click **Finish**.

9. If you wish to add an existing source file to the project the Add Existing Sources dialog allows you to do this. Since we are only using new source files in this project we will not worry about this now.

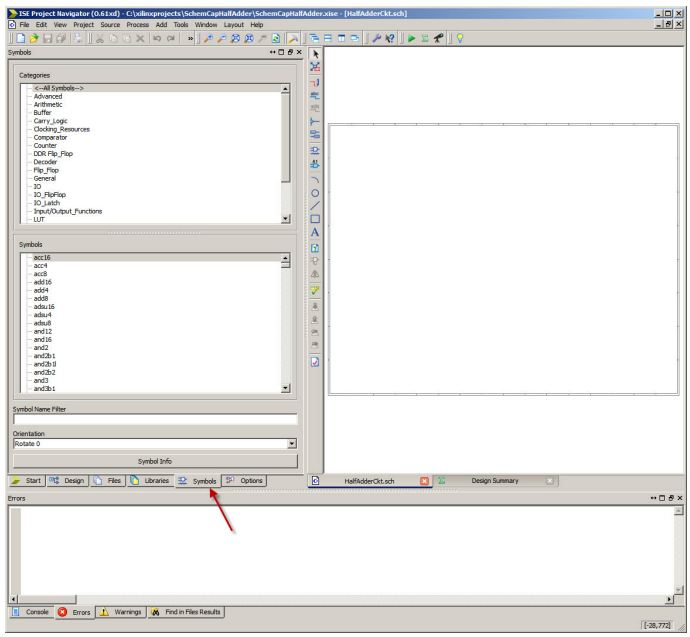
3.3 Using the Schematic Editor in the Project Navigator

We are now ready to start working with the schematic editor.

1. The Schematic Editor window will have appeared within the project navigator interface at the completion of the last step in the previous section.



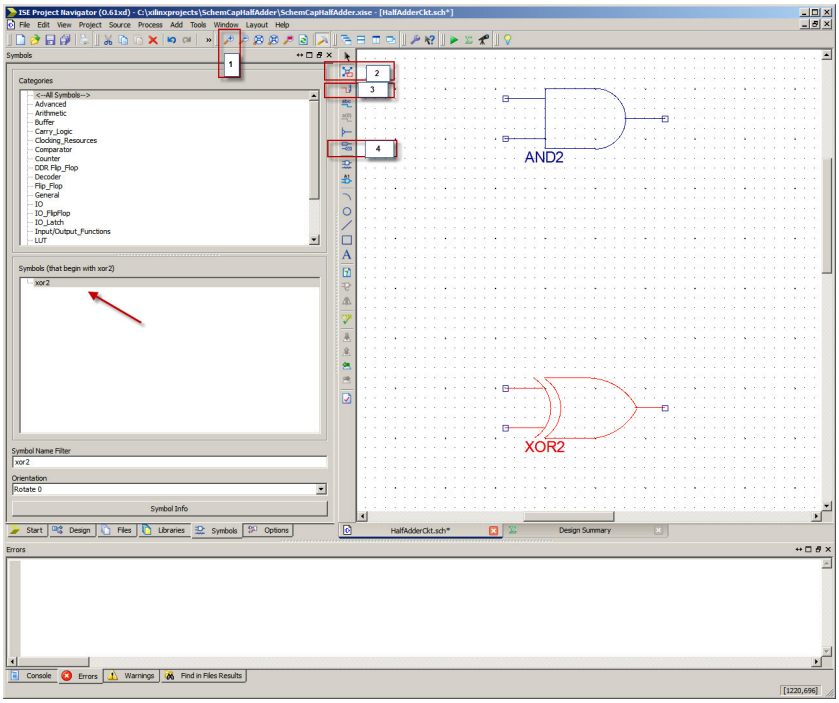
2. We will now insert logic gates into our schematic. Select the Symbols **Symbols** tab in the **Options** toolbar and a list of symbol categories will appear on the left hand side of the **Project Navigator** window.



Select **Logic** from this list and a list of simple logic gates will be shown in the lower **Symbols** scroll box. For the half adder, we will require a 2 input XOR gate and a 2 input AND gate. To add a simple logic gate to the circuit, left-click to select a symbol from the Symbols list, drag the cursor to the pallet where

you want the symbol to be placed, and left-click again to drop the gate in the schematic. You can also find a symbol by typing a name into the symbol name filter.

3. So, now, left click on **AND2** then move your mouse onto the drawing area. You will see the symbol for the AND gate appear. Put it in the desired location and left click the mouse. This will drop it. Do the same for the XOR gate (XOR2).



The figure above highlights the tools necessary for creating a basic digital circuit using schematic capture:

(a) The Magnifying Glass icon indiscriminately zooms to the center of the schematic.

(b) The Zoom box icon is used to draw a box using the mouse to magnify a specific area of the schematic.

(c) the Wire-Add tool icon places cursor in wire-add mode.

(d) The Add I/O marker tool icon places cursor into add I/O marker mode.

4. Next, we need to switch to **Wire** mode to add wires to our circuit. This can be accomplished by:

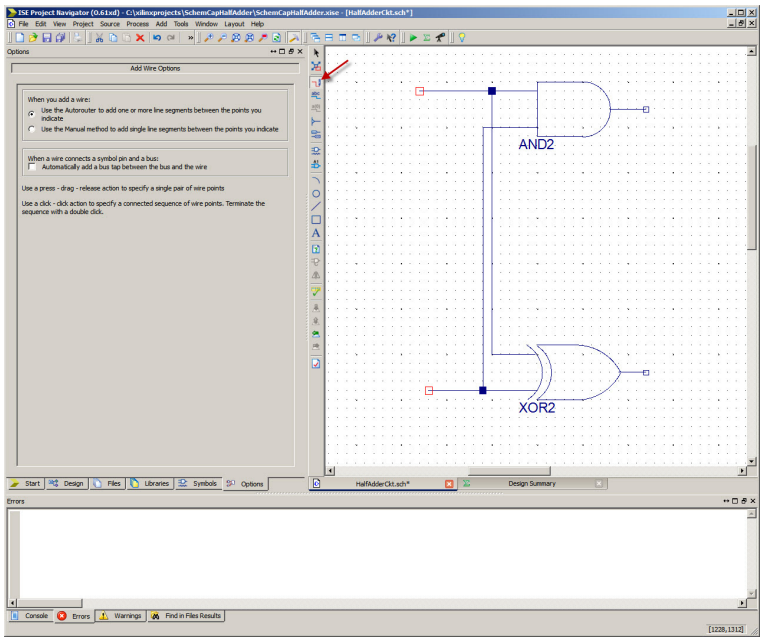
(a) Clicking on the button in the Tools toolbar, **or**,

(b) Selecting **Add** → **Wire** in the menu bar, **or**,

(c) Control-W key pair, **or**,

(d) Pressing the right mouse button while in the drawing area and select **Add**→**Wire** from the floating menus.

5. Once in **Wire** mode connect the gates by doing the following. Place the mouse on a pin of a gate. The shape of the cursor will change to a different shape. Left click the mouse button and move to new location that you wish to be connected. Left click the mouse again. A wire will be created between the two desired pins.

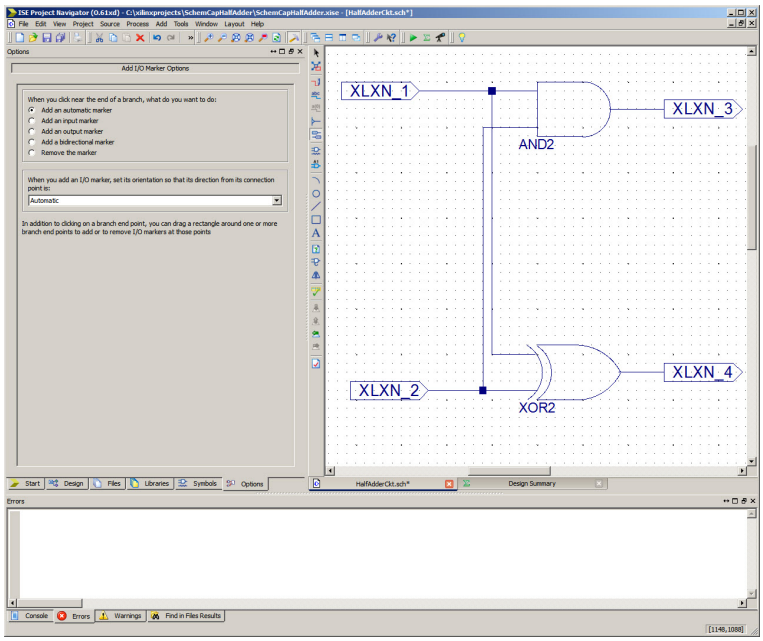


6. Useful Hints:

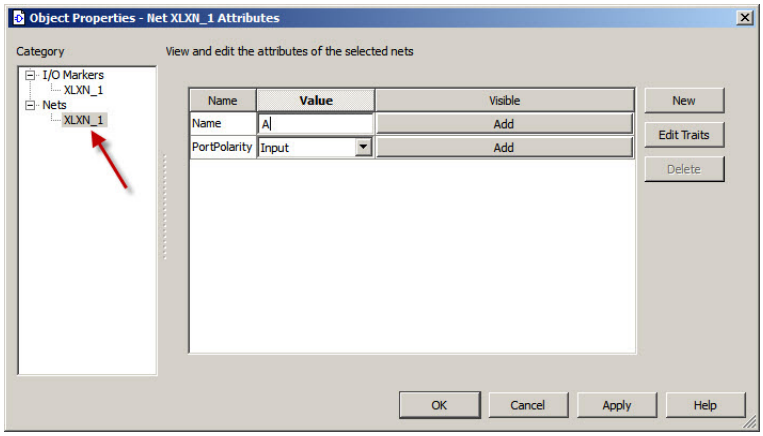
   - To delete a wire or a gate, click on it while in the **Select** mode  and press [DELETE].
   - The **Edit** menu contains standard editor functions (undo, select, cut, paste, copy, etc . . .)
   - You will find that you cannot connect two or more wires to an input or output of a gate. Connect the first wire to the gate's connection, then connect additional wires to the first wire.
   - You may also find that the wires will refuse to connect if there are too many wires or gates cluttered together. To solve this, make room for the new wire. Use the **Select** mode to move the wires and the gates around. Click and hold on them then drag them to a new location.
   - To move a gate, you must go back to select mode by either selecting the cursor in the tool bar or pressing the Esc key on the keyboard.

7. Adding top-level I/O markers to your circuit tells the synthesizer and simulator tools which ports to regard as overall inputs and outputs.

8. To add these to your circuit, left-click on the **Add I/O** marker tool icon  to place the cursor into add I/O Marker mode. Left-click on the end of a wire to add an I/O marker and then repeat until a marker is placed on all inputs and outputs.

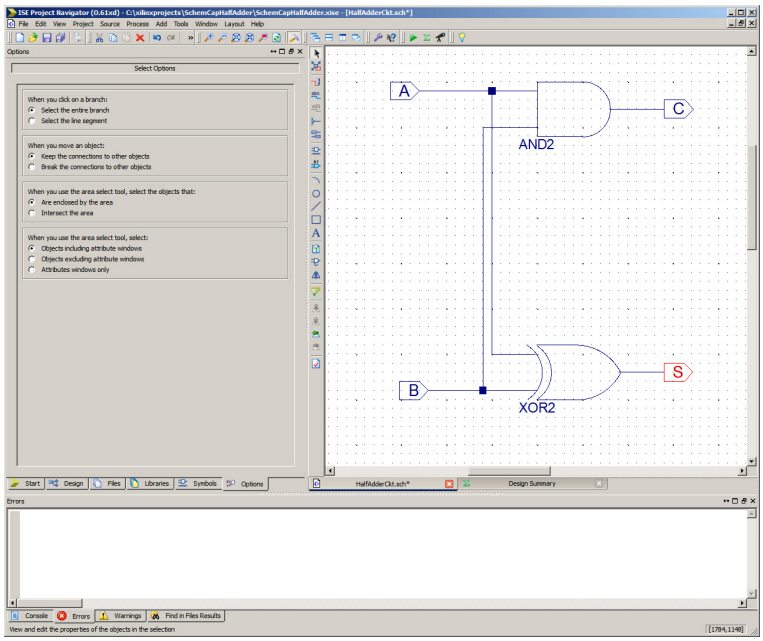


9. Go back to regular cursor mode (Select Mode Icon)  and double left-click on an I/O marker. When the I/O marker's object properties dialog box appears, select the Nets category and enter a meaningful value for the Name field of the I/O marker. The finished dialog box should look similar to the following:

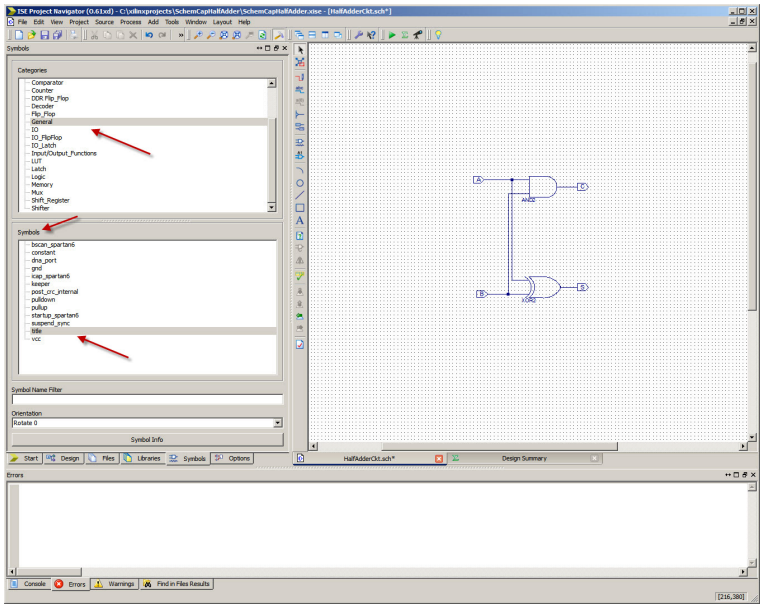


Press **Apply** and then **OK**.

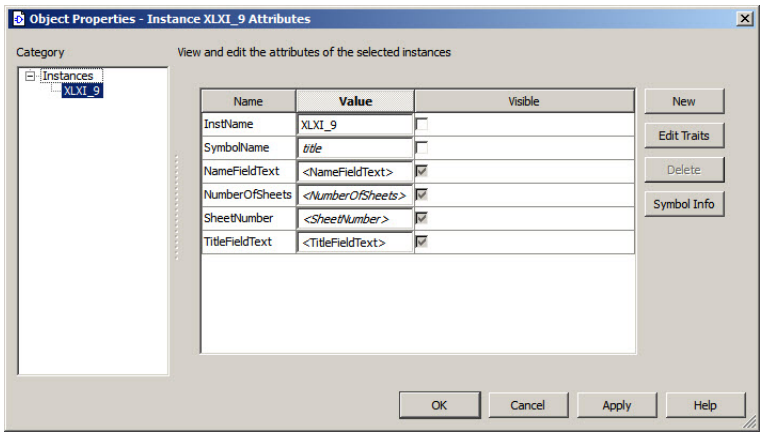
Repeat this process for all IO (i.e., Enter "A,B" as inputs and "S,C" as outputs). The finished circuit should look similar to the following:



10. In the **Symbols** tab  select **General** in the **Categories** pane. In the **Symbols** pane below select **Title**.

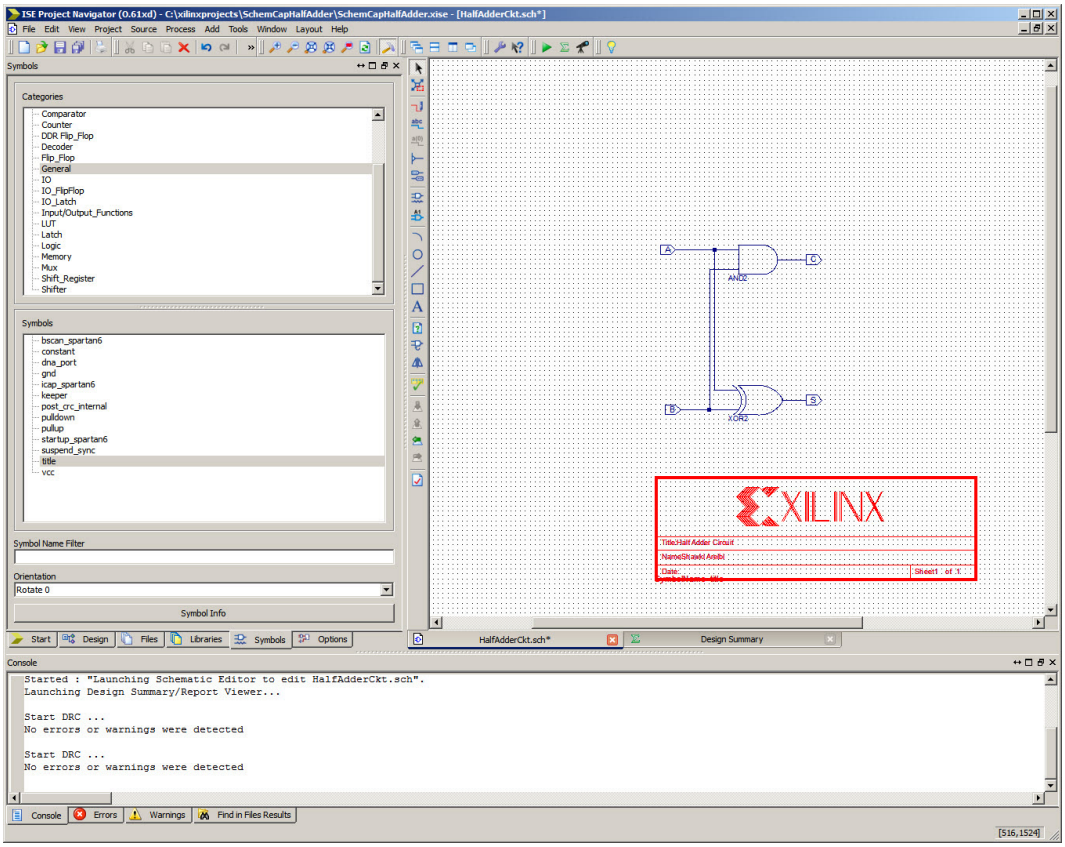


Place this symbols in the lower right hand corner of the schematic. To add a title, your name and the date switch to **Select** mode by clicking the  icon in the **Tools** toolbar and double left click on the title block symbol in the drawing area. In the Object Properties dialog box that appears change the **Value** field of **TitleFieldText** to "Half Adder" and change **NameFieldText** to an appropriate values. Click **OK**.



11. We will now check the design for connection errors. Select **Tools** → **Check Schematic** in the menu bar. A dialog box will appear showing error numbers and messages or if all is well a "No errors or warning were detected" message. If the test indicates that there are no errors, save your schematic.

We now should have a view as shown below:

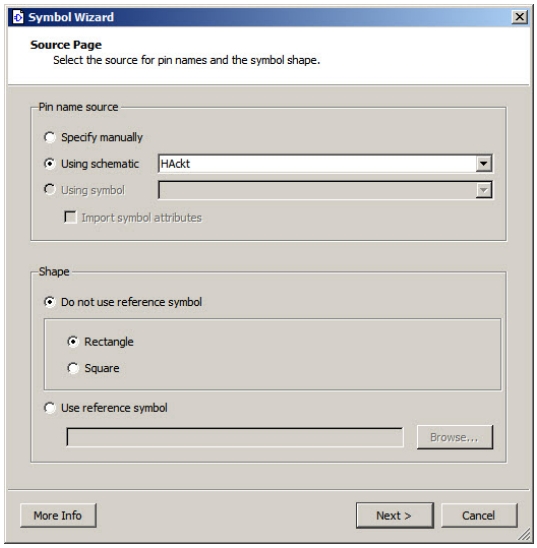


We will next create a macro symbol of the half adder to be used in a full adder design.

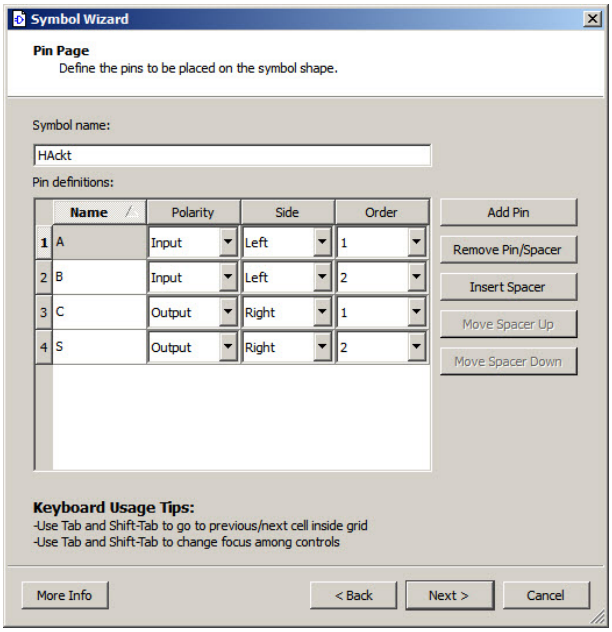
3.4 Creating a Macro Symbol for Half Adder

After designing the Half Adder and checking it for connection errors we will create a macro symbol so that we can use it in the design of the Full Adder. To create a macro circuit, follow these steps:

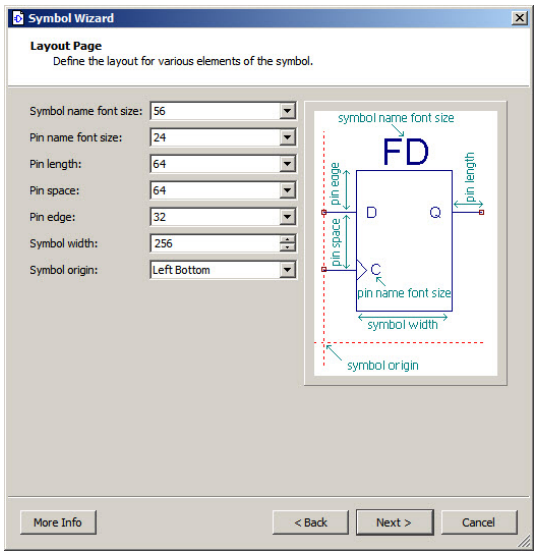
1. Select **Tools** → **Symbol Wizard** from the main menu toolbar.
2. The Symbol Wizard dialog box appears.



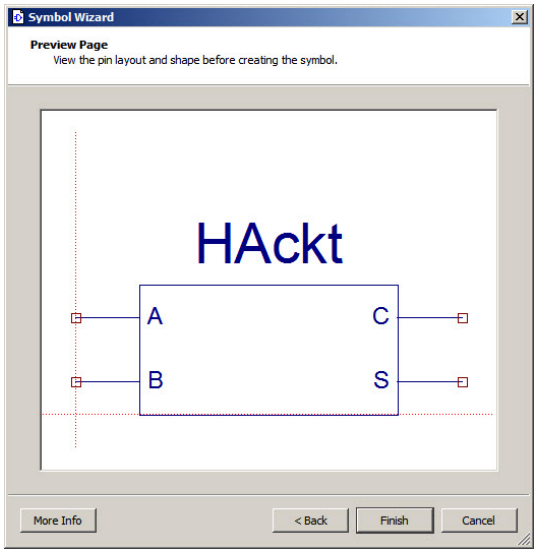
3. Under "Pin name source" select "Using Schematic" which defaults to the circuit you just created, HAckt or HalfAdderCkt depending on how you named your original schematic. Press **Next**.
4. The next dialog box presents general pin placement options that you can modify depending on how you want to name your pins. Press **Next**.



5. The next dialog box presents symbol size options that you can modify depending on how you want the schematic symbol to appear. Press **Next**.



6. The final dialog box gives you a preview of the symbol, given the settings you've selected. Click **Finish** to finalize the symbol or **Back** to modify the symbol settings.



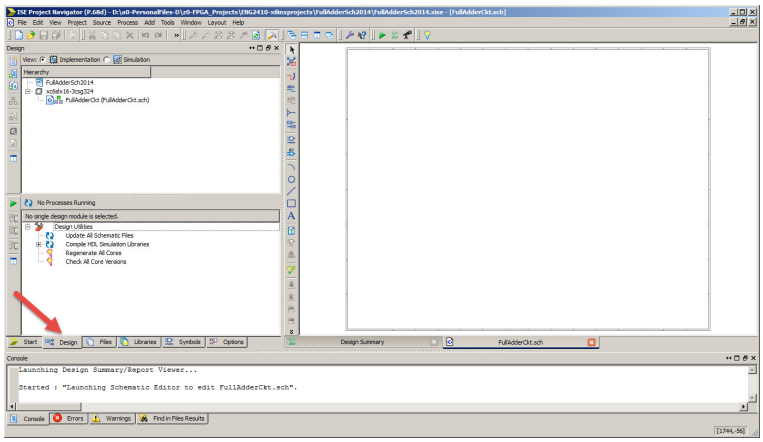
7. Save all your work and close the project.

8. We will next use the Half Adder circuit symbol in design of the Full Adder circuit.

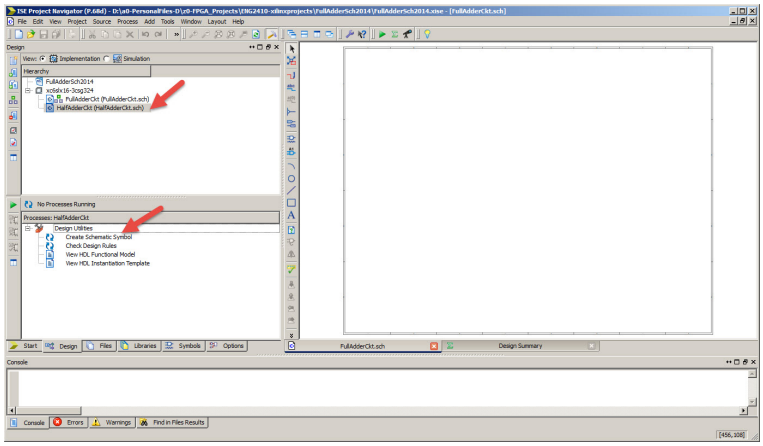
4 Completing the Design of the Full Adder

A full adder can now be constructed by using two half adders and an OR gate. Here is brief description of the steps that have to be taken to design the Full Adder circuit.

1. Load the Project Navigator.
2. Start a new project by clicking the **New Project** tab.
3. Call the project FullAdderSch.
4. Ensure that the setting for the target FPGA, device, synthesis tool, e.t.c is correct.
5. Create a new schematic file for your Full Adder and call it "FullAdderCkt".
6. Make sure you are now in the Design Mode as seen in the figure



7. While the **FullAdderCkt.sch** is highlighted in the Hierarchy **Right Click** on Add Source. The System will give you the option to add an existing schematic. You will choose the directory where the HAckt.sch exists and add it by double clicking on it. This will make the Half Adder circuit schematic available for use in the symbol pane.
8. Make sure you highlight **HalfAdderCkt.sch** or **HAckt.sch** (depending on how you created it earlier) in the Hierarchy pane (upper left view).
9. Now in the Processes pane, double click on **Create Schematic Symbol** as seen in the figure below



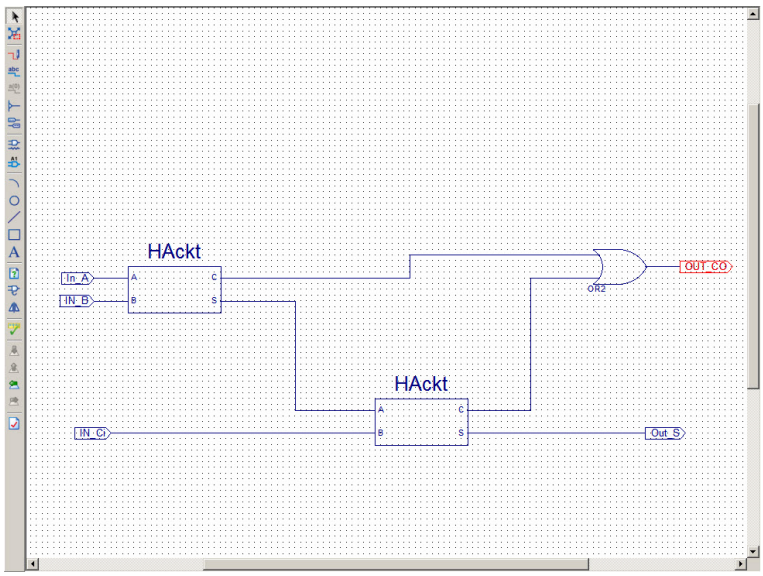
10. While the **FullAdderCkt.sch** is highlighted in the Hierarchy, move the mouse to the Symbols tab. Highlight the new entry created in the Categories (points to the directory you are working on). You will notice that the Half Adder Ckt has been created automatically for you. Double click on the HalfAdderCkt and drop it in the Schematic pane.

11. Use the Schematic Editor to connect two half adders and an OR gate as seen below.

12. Name your inputs as follows: IN_A, IN_B, IN_Ci, OUT_S, OUT_CO.

13. Make sure you check the design for connection errors.

14. Create a Macro Symbol for the Full Adder so that you can use it in future designs.



In the next section we will repeat the same procedure of creating a UCF file for the Full Adder, Generate a Bit Stream and downloading the bit-stream onto the FPGA to verify the correctness of our design.

5 Creating a UCF File

A **U**ser **C**onstraint **F**ile (UCF) is used to assign I/O pins in a design to the actual pins on the FPGA.
Please refer to **Appendix B** for more information.

1. Click on the Design Tab  so that you can view the Hierarchy and Processes panes.
2. Highlight and right-click on the schematic source **FullAdderCkt.sch** in the **Hierarchy** pane of the Project Navigator and select **New Source**  from the floating menu.
3. From the list of file types select **Implementation Constraints File**. Name the file "FullAdder_UCF". Ensure the **Add to Project** box is selected. Click **Next**.
4. The final dialog box is for confirming the information input in the previous dialog boxes. Click **Finish** if the information is correct.
5. In the **Hierarchy** pane ensure "FullAdder_UCF.ucf" is highlighted. In the **Processes** pane click on the "+" in  to expand the section and double click **Edit Constraints (Text)** in the list.
   - The UCF file has the following format:

     NET <*pin name in Schematic design*> LOC=P<*pin number on FPGA*>

     NOTE: Make sure to use upper case letters for the pin names.
   - Let us assign the inputs to the dip switches on the FPGA and the outputs to the LEDs. Please refer to the NEXYS 3 board schematic for more information about pin connections. We will connect 'IN_A' to Switch 2 (SW0), 'IN_B' to Switch 1 (SW1), 'IN_C_i' to Switch 0 (SW0), 'OUT_S' to LED 0 (LD0) and 'OUT_C_o' LED 1 (LD1). For further information check the link "NEXYS 3 Board Pins (FOR UCF FILE ASSIGNMENT) on the web.

—Design pin	—FPGA pin	—Description
IN_A	V9	SW2
IN_BB	T9	SW1
IN_Ci	T10	SW0
OUT_S	U16	LD0
OUT_Co	V16	LD1

6. Enter the following statements in the **Workspace** panel:
   NET IN_A LOC=V9;
   NET IN_B LOC=T9;
   NET IN_Ci LOC=T10;
   NET OUT_S LOC=U16;
   NET OUT_Co LOC=V16;
7. You can now save and close the file by pressing **File** → **Save**

You can get a better idea of our setup by looking at the Digilent NEXYS 3 Programmer's model in **Appendix B**. We will look at this setup in more detail in the following sections.

6 Generating a Bit Stream "Compiling the Design"

Hitherto, we have examined how to design a digital circuit using the Xilinx ISE Design Suite 14.6 Software. We will now look at how to compile (generate the bit-stream) and download the design to the Digilent NEXYS 3 board.

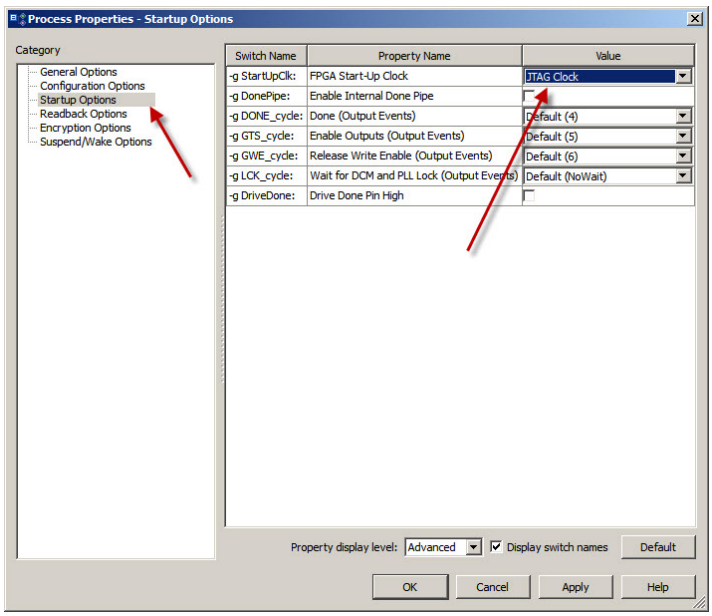
1. Now go back to the Project Navigator window. Highlight **FullAdderCkt.sch** in the **Hierarchy** pane. Left double click Synthesize - XST in the **Processes** pane to synthesize the design.

2. When the synthesis stage has completed, you will see the following message on the console panel (process "synthesis – xst" completed successfully).

3. Left double click Implement Design to implement the design.

4. When the implement stage has completed successfully you will see the following message on the console panel (process "Generate Post-Place & Route static Timing" completed successfully).

5. Right click on Generate Programming File and choose properties. A window will pop as seen below:



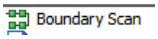
   Highlight startup options and change the switch name "-g StartUpClk" value to JTAG clock. Press **Apply** then **OK**.

6. Finally, left double click Generate Programming File to generate the programming file that will be downloaded to the FPGA. It is possible to go straight to generating the programming file, and the Project Navigator will determine which of the previous step need to be run to produce an up to date programming file.

7. When the generation of the bit-stream stage has completed successfully you will see the following message on the console panel (process "Generate Programming File" completed successfully).

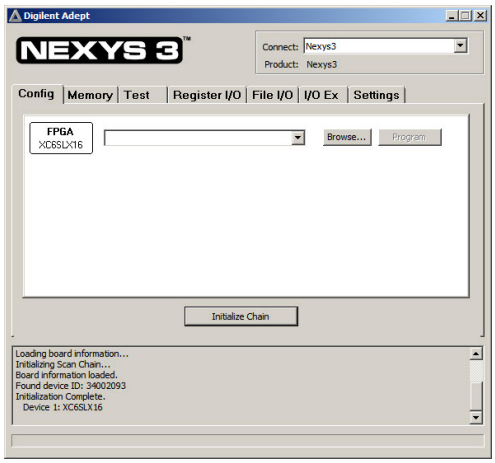
7 Downloading the design to the FPGA

Downloading a bit-stream to the FPGA is accomplished via either the **iMPACT** tool within the ISE Project Navigator or the Digilent **Adept** tool for NEXYS 3 board.

1. Steps to download the design using the **iMPACT** tool within the ISE Project Navigator:
   (a) Select "Configure Target Device". **Double click** and a new window will appear stating that "No iMPACT project file exists". You can ignore the message and **press** OK.
   (b) A new window will then appear (ISE iMPACT).
   (c) **Double click** on the Boundary Scan icon 

to start the ISE iMPACT tool.
   (d) Click on the boundary scan box in the top menu "Initialize Chain". Make sure that your FPGA is powered on and connected to the host. A small window will appear with the message "Do you want to continue and assign the configuration files(s)?". **Press** yes.
   (e) A new window will appear (Assign New Configuration File). Here you will zoom to the directory where your *.bit file exists.
   (f) **Double click** on the file name.
   (g) A Message window titled Attach SPI or BPI PROM will appear. **Press** Yes.
   (h) A new window titled "Add Prom File" will appear. Just simply **close** the window.
   (i) A message box will appear (Device Programming Properties). **Click** OK.
   (j) Move the cursor over the device that appears in the **Boundary-Scan** tab and right click the mouse button. A menu will appear. **Press** the left mouse button and select **Program**.
   (k) If the programming succeeds you will see the following message "Program Succeeded".
2. Steps to download the design using the Digilent **Adept** tool:
   (a) Load the Digilent Adept from the 

→ **All Programs** → **Digilent** → **Adept** → 
   (b) The Digilent Adept window will appear as seen in the Figure below.
   (c) Click the **Browse** icon. A new window will appear to choose your bit file.
   (d) Zoom onto the directory where your bit file resides and double click it.
   (e) Click the **Program** button.
   (f) The system will start to program the device and at the bottom of the Adept Tool you will see some messages indicating that it has successfully programmed the device.



8 Testing the Design

Depending on the state of the inputs, you may or may not see some of the LEDs on the bar-graph display glowing. We have assigned

1. The Sum (S) bit to **LEDs** LD_0.
2. The carry-out bit (C_O) is displayed on **LED** LD_1.

We are using the slide switches for our A, B and C_i inputs.

1. The A inputs is assigned to **Switch** SW_2.
2. The B input is assigned to **Switch** SW_1.
3. The C_i is assigned to **Switch** SW_0.

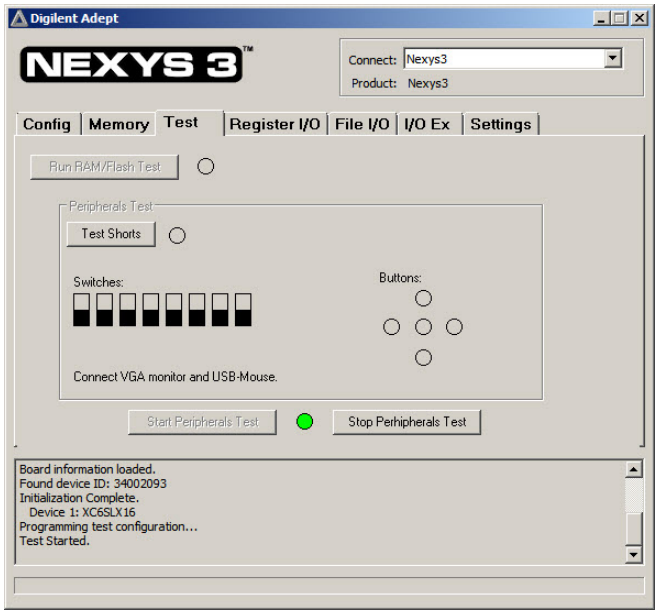
Moving a switch to the **ON** position puts a 1 on the input. Moving a switch to the **OFF** position puts a 0 on the input.
Try different combinations of inputs and verify that the circuit is working correctly.

9 Appendix A - Setting-up and Testing the NEXYS3 board

This is intended to allow the student to quickly set up the NEXYS 3 board for this tutorial. It does not attempt to explain the configuration and is in no way a substitute for the documentation provided with the board. It will allow you to use the slide switches as input and the LEDs as outputs.

1. Connect the USB cable to the NEXYS 3 board.
2. Connect the host computer to your USB cable.
3. When the power switch to the board is on a small yellow LED labeled *Done* should glow.

You can test if the Digilent NEXYS3 Board is operational by using the Digilent Adept Tool. Double click on the  icon and you will see the Digilent Adept GUI on your screen. Press the **Test** icon. A new menu will appear. Press the "Start Peripherals Test".



The test will display different values on the 7-segment display. You can also test the switches and light emitting diodes by sliding the switches to the on-off position. Once a switch is turned on the corresponding LED will glow. You will also notice that the switches on the Digilent Adept tool will change value. You can also test the push buttons by pressing on them. You will see the color of the corresponding button on the Adept tool change from transparent to black. Once you are satisfied that the FPGA board is operational you can press the "Stop Peripherals Test". By pressing the "Reset Button" on the FPGA you will reset the board to the factory setting where it tests all other modules on the PCB board. Power off the board using the slide switch found at the top left part of the board.

10 Appendix B - LEDs, 7-Segments and Switches

The following sections explain the connection and location of the DIP switches and LEDs of the Digilent NEXYS 3 Board.

10.1 LEDs

The Digilent NEXYS 3 Board provides a series of eight LEDs (LD0–LD7) for use. All of these LEDs are **Logic Active High** meaning that an LED segment will glow when a logic-high is applied to it. The following table show the connection from the NEXYS 3 Board to LEDs expressed as UCF constraints.

—Description	—Location
NET LD0	LOC=U16
NET LD1	LOC=V16
NET LD2	LOC=U15
NET LD3	LOC=V15
NET LD4	LOC=M11
NET LD5	LOC=N11
NET LD6	LOC=R11
NET LD7	LOC=T11

Table 1: NEXYS 3 (Light Emitting Diodes) LEDs

10.2 Seven Segment Displays

The Digilent NEXYS 3 Board provides four multiplexed 7-segment displays for use. The following tables show the connection from the NEXYS 3 Board to the 7-segment displays expressed as UCF constraints.

—Description	—Location
NET CA	LOC=T17;
NET CB	LOC=T18;
NET CC	LOC=U17;
NET CD	LOC=U18;
NET CE	LOC=M14;
NET CF	LOC=N14;
NET CG	LOC=L14;
NET DP	LOC=M13;
NET AN0	LOC=N16;
NET AN1	LOC=N15;
NET AN2	LOC=P19;
NET AN3	LOC=P17

Table 2: NEXYS 3 (7-Segment display)

10.3 Slide Switches

The Digilent NEXYS 3 board has a bank of eight slide switches which are accessible by the user.

When closed or ON, each DIP switch pulls the connected pin of the NEXYS 3 Board to ground. When the DIP switch is open or OFF, the pin is pulled high through a $10K\Omega$ resistor.

The table below shows the connections from the Digilent NEXYS 3 Board to the switches expressed as UCF constraints.

—Description	—Location
NET SW0	LOC=T10
NET SW1	LOC=T9
NET SW2	LOC=V9
NET SW3	LOC=M8
NET SW4	LOC=N8
NET SW5	LOC=U8
NET SW6	LOC=V8
NET SW7	LOC=T5

Table 3: NEXYS 3 (Slide Switches)

10.4 Push Buttons

The Digilent NEXYS 3 board has five pushbuttons (labeled BTNS through BTNR) which are accessible by the user.

When pressed, each pushbutton pulls the connected pin of the NEXYS 3 Board to ground. Otherwise, the pin is pulled high through a $10K\Omega$ resistor. The table below shows the connections from the the Digilent NEXYS 3 Board to the push buttons expressed as UCF constraints.

—Description	—Location
NET BTNS	LOC=B8
NET BTNU	LOC=A8
NET BTNL	LOC=C4
NET BTND	LOC=C9
NET BTNR	LOC=D9

Table 4: NEXYS 3 (Pushbuttons)