TUTORIAL ON USING XILINX ISE DESIGN SUITE 14.6: Design Entry using VHDL (Full Adder) for 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. Another objective is to show the basic concepts of VHDL. VHDL stands for (**V**ery High-Speed-Integrated-Circuit **H**ardware **D**escription **L**anguage. VHDL is a very powerful digital design language. It allows designers to specify a digital circuit at a high level of abstraction.

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 VHDL editor.
3. How to synthesize, implement your design.
4. 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.

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

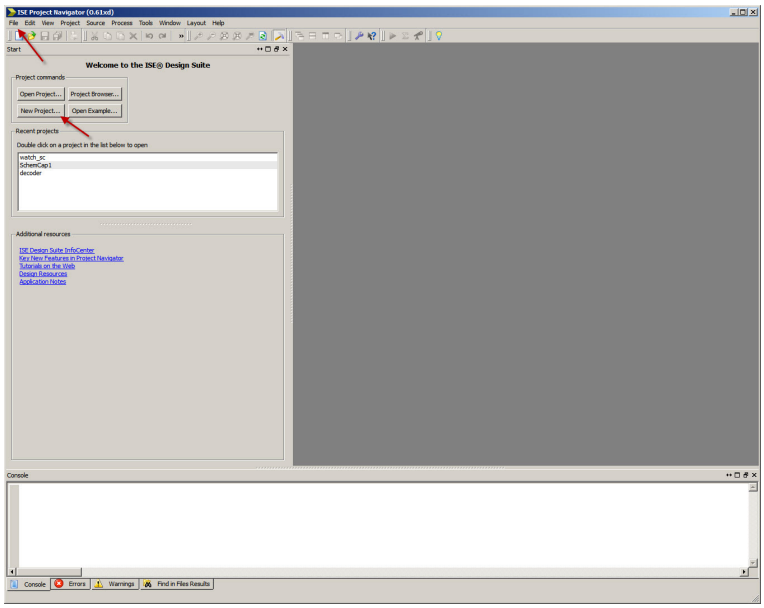
This circuit will be designed using VHDL.

4 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.

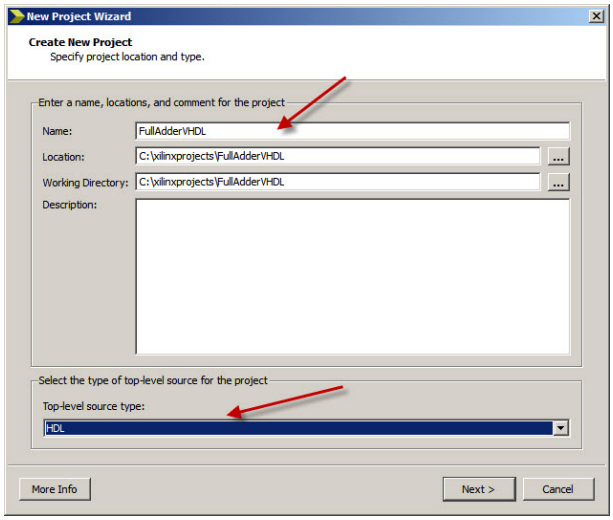
1. Load the Project Navigator from the Start → **All Programs** → **Xilinx ISE Design Suite 14.6** → **ISE Design Tools** → **Project Navigator**.

2. The Project Navigator window will appear.

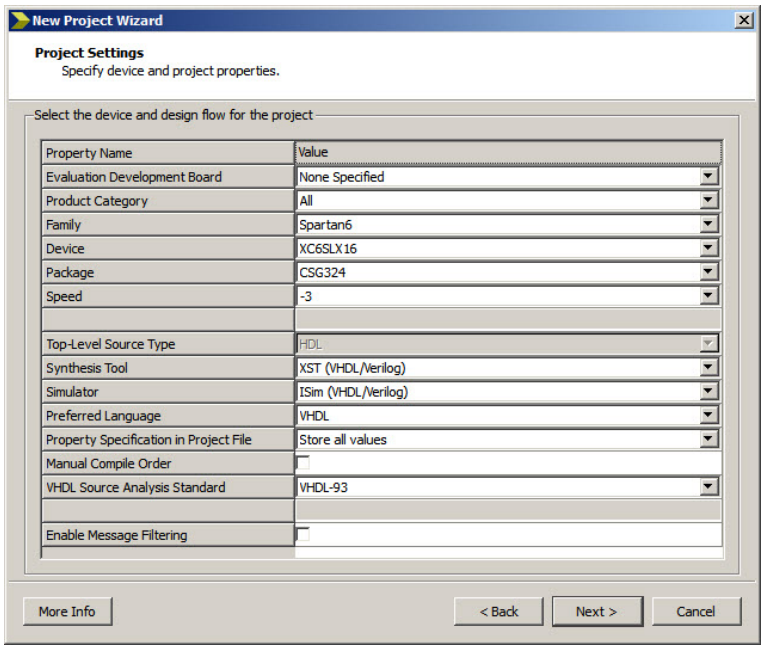


3. Select either **File** → **New Project** or click on the **New Project** tab.

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 "FullAdderVHDL". In the **Top-Level Source Type** section select **HDL** and click **Next**.



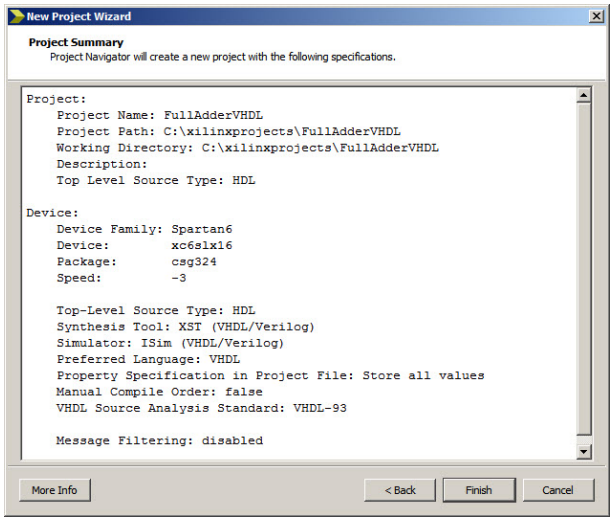
5. Another New Project Wizard dialog box will appear prompting you for device, synthesis and simulation settings for the project.



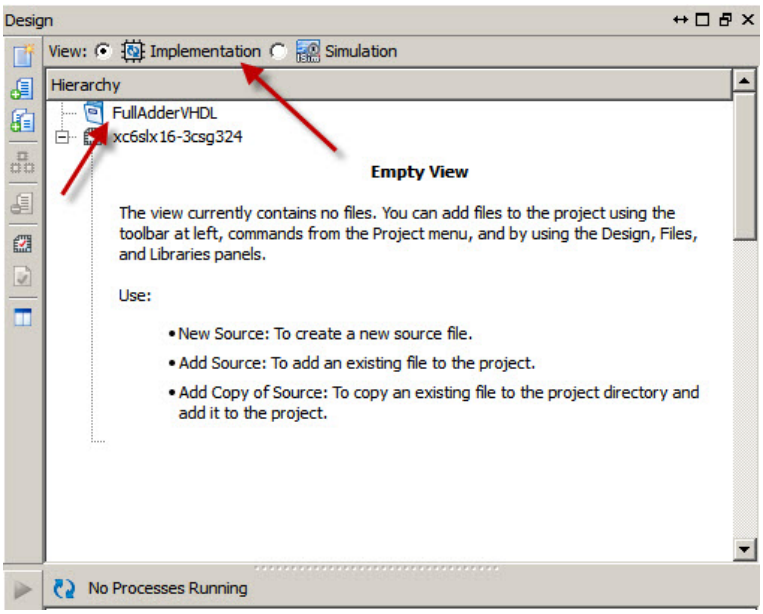
In this dialog box verify the following settings:

- **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**.

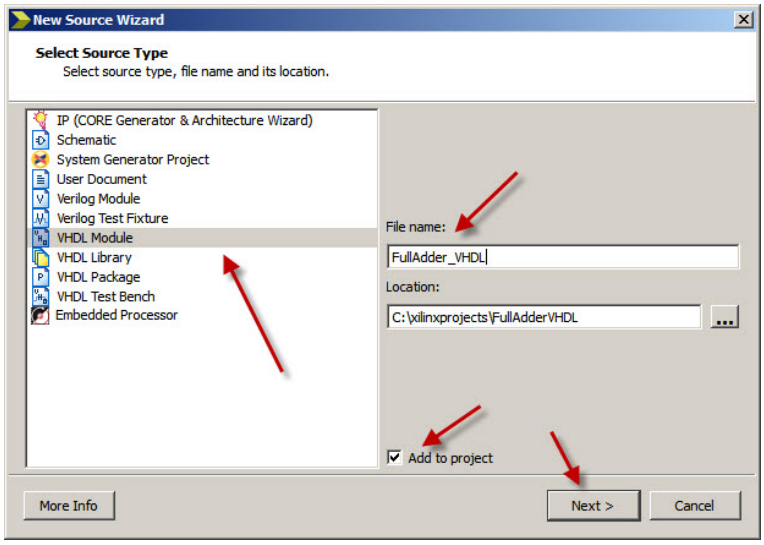


7. The next dialog box will allow you to create a new source file and add it to the project.



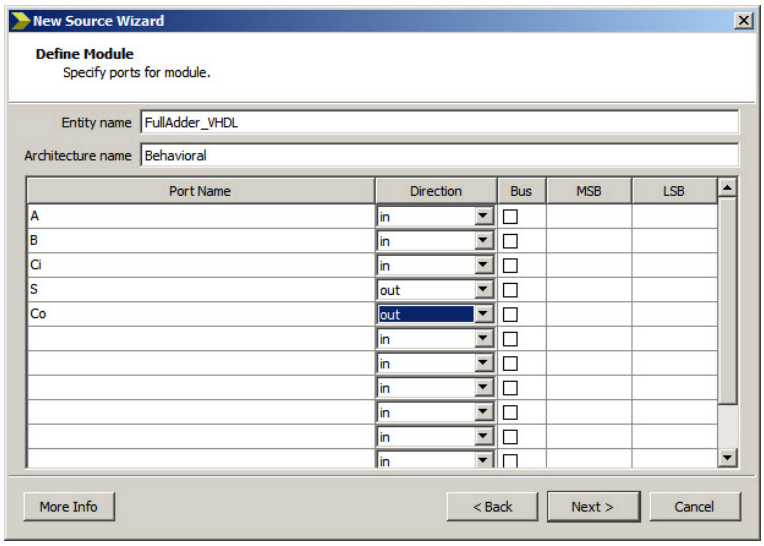
Click on the **New Source** button.

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



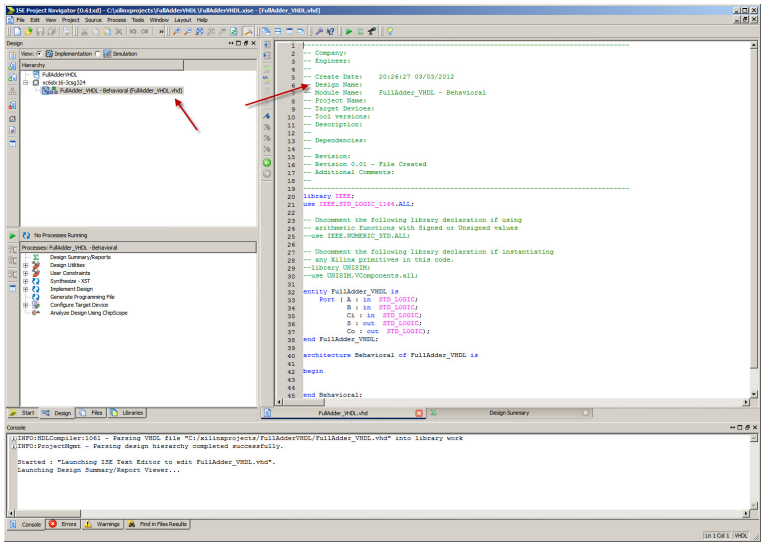
The default location is the current project directory and can be left as is. Ensure the **Add to Project** box is selected and click the **Next** button.

9. In the new dialog box that appears, you will be asked to specify ports for the module you are creating. Since a Full Adder has 3 inputs and 2 outputs, you will add ports A, B and Ci as inputs and add ports S, Co as outputs.



click **Next** and verify the information in the next dialog box and click **Finish**.

10. You will now return to the **Project Navigator** and the "FullAdder_VHDL.vhd" source template that you have created will now be listed.



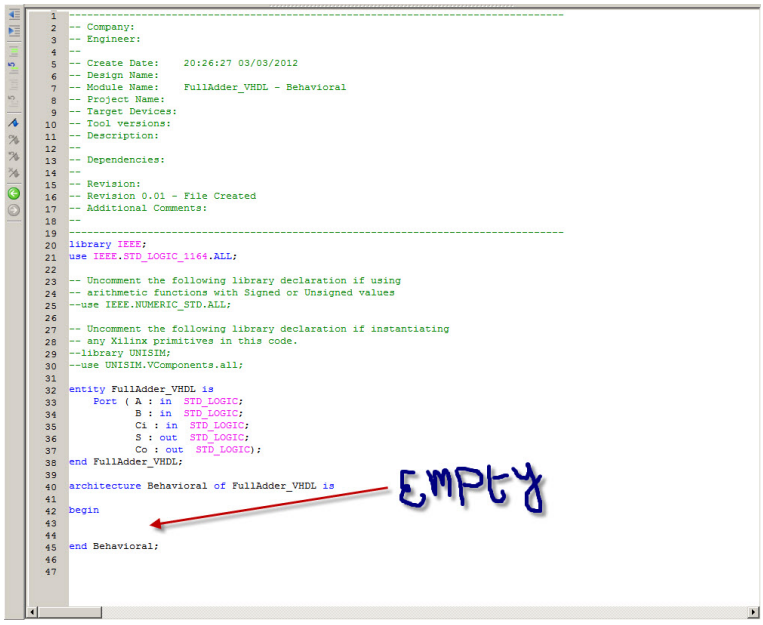
11. You will need to edit the FullAdder_VHDL.vhd to add VHDL statements to realize the functionality of a Full Adder circuit.

12. **NOTE: 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.**

5 Using the HDL Editor in the Project Navigator

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

1. The VHDL editor window should show the following code:



2. Between **begin** and **end Behavioral** add the following lines:

```
S <= ((A xor B) xor CI);
CO <= ((A and B) or (CI and (A xor B)));
```

3. We must now check that the syntax of the VHDL code is correct. In the *Project Navigator* window highlight the VHDL source in the **Sources in Project** pane. Expand Synthesize - XST if it is not already and double click Check Syntax. A process will be spawned to verify the VHDL code for syntax errors.

   - If there are none, a green check mark will appear next to the **Check Syntax** process.
   - If errors where found, A red cross will appear next to the **Check Syntax** process. The errors will be listed in the **Errors** tab of the transcript window located at the bottom of the Project Navigator window. In it you will see a listing of the errors and the lines on which they occur. If you double click on an error message a red dot will appear in the VHDL editor pane next to (or close to) the line where the error was found.

4. Once you have eliminated all of the errors, save your work.

5. Next, we will assign the pins in our design to actual pins on the FPGA so that we can test it (See Section "Creating UCF File").

6 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 VHDL source **FullAdder_VHDL -Behavioral(FullAdder_VHDL.vhd** 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 'A' to Switch 2 (SW2), 'B' to Switch 1 (SW1), 'Ci' to Switch 0 (SW0). We will also connect 'S' to LED 1 (LD1) and 'Co' LED 0 (LD0). For further information check the link "NEXYS 3 Board Pins (FOR UCF FILE ASSIGNMENT) on the web.

—Design pin	—FPGA pin	—Description
A	V9	SW2
B	T9	SW1
Ci	T10	SW0
S	V16	LD1
Co	U16	LD0

6. Enter the following statements in the **Workspace** panel:
   NET A LOC=V9;
   NET B LOC=T9;
   NET Ci LOC=T10;
   NET S LOC=V16;
   NET Co LOC=U16;

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.

7 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.

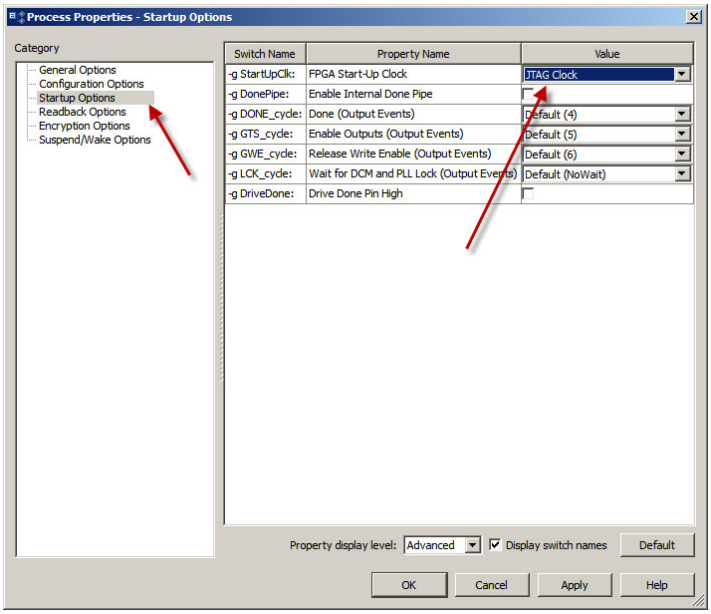
1. Now go back to the Project Navigator window. Highlight **FullAdder_VHDL - Behavioral(FullAdder_VH** in the **Hierarchy** pane. Left double click Synthesize - XST in the **Processes** pane to synthesize the design.

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:



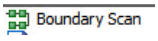
   Highlight **startup options** and change the switch name "-g StartUpClk" value to JTAG clock. Press **Apply** then **OK**.

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).

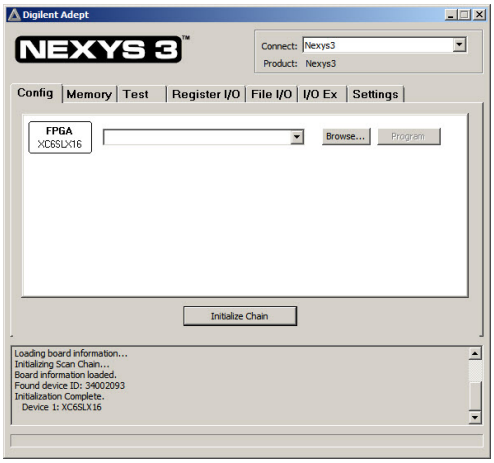
8 Downloading the design to the FPGA

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.



9 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 our Sum (S) bit to **LEDs LD1**. The carry-out bit (Co) is displayed on **LED LD0**.

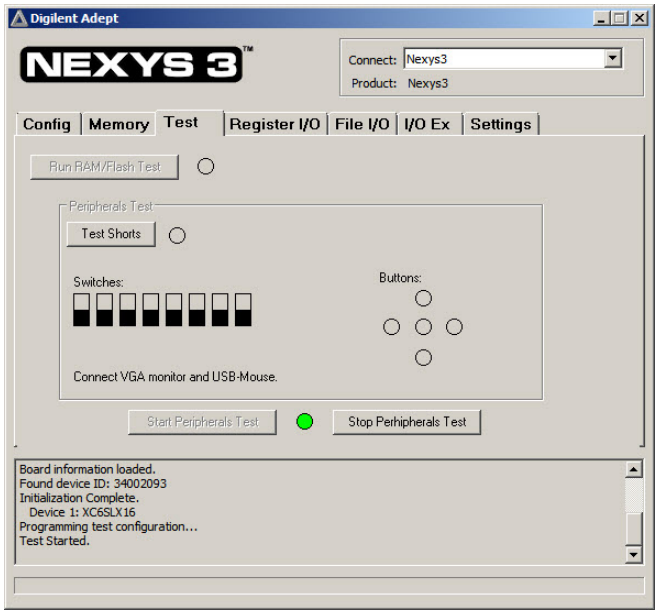
We are using the slide switches for our A, B and Ci inputs. The A inputs is assigned to **Switch SW2**. The B input is assigned to **Switch SW1**. Finally, the Ci is assigned to **Switch SW0**. Moving a switch to the **ON** position puts a 1 on the input. Moving a switch to the **OFF** position puts a 0 on the input. Try different combinations of inputs and verify that the circuit is working correctly.

10 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.

11 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.

11.1 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

11.2 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)

11.3 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)

11.4 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)