Xilinx ISE 7.1 Tutorial

This is a step-by-step procedure to use the Xilinx ISE 7.1i software for simulating and prototyping of your designs. We will be using an adder circuit as an example.

Start the Xilinx software. (Start, All Programs, Xilinx ISE 7.1i, Project Navigator)
Notes:

1. You can stop your Xilinx session at any time. Save all source files that are open, and exit the software. When you exit, the project file is automatically saved with the most recent changes you made. When you restart the software, you will see the content of your project with the last saved changes.

2. You can access Help at any time during a session. Press **F1** to view the help for the specific tool or function your are currently using. Design flow-based help is called ISE Help and is accessible from the **Help** menu in Project Navigator. This help package contains information about creating and maintaining your complete design flow in ISE.

1. Create a New Project**:**

A **project** in ISE is a collection of all files necessary to create and download a design to the selected device. The project we will be creating will be targeted to use our FPGA and allow us to draw a schematic as the main way of entering the design. Here are the steps for creating a new project:

1. Select **File, New Project**.
2. In the **New Project Wizard** dialog box, type the desired location in the **Project Location** field, or browse to the directory under which you want to create your new project directory using the browse button next to the **Project Location** field. You will need to save your project on your network drive, Z: or some removable media such as a USB drive. Do not save your files on the local machine.
3. Enter "Lab3" in the **Project Name** field. When you enter "Lab3" in the **Project Name** field, a Lab3 subdirectory is automatically created in the directory path in the **Project Location** field. For example, for the directory path Z:\COE203, entering the Project Name "Lab3" modifies the path as Z:\COE203\Lab3.
4. Use the pull-down arrow to select **Schematic** from the **Top-Level Module Type** field. Click in the field to access the pull-down list. (NOTE: you can apply the fundamentals learned from this tutorial to either an HDL or schematic design containing both schematic and/or HDL sources.)
5. Click **Next**.

6. In the **New Project Wizard Device and Design Flow** dialog box, use the pull-down arrow to select the **Value** for each **Property Name**. Click in the field to access the pull-down list. Make sure the values are as follows:

    - Device Family: **Spartan3**
    - Device: **xc3s200**
    - Package: **ft256**
    - Speed Grade: **-4**
    - Top-Level Module Type: **Schematic**
    - Synthesis Tool: **XST**
    - Simulator: **ModelSim**
    - Generated Simulation Language: **Verilog**.

When the table is complete, your project properties should look like the following:

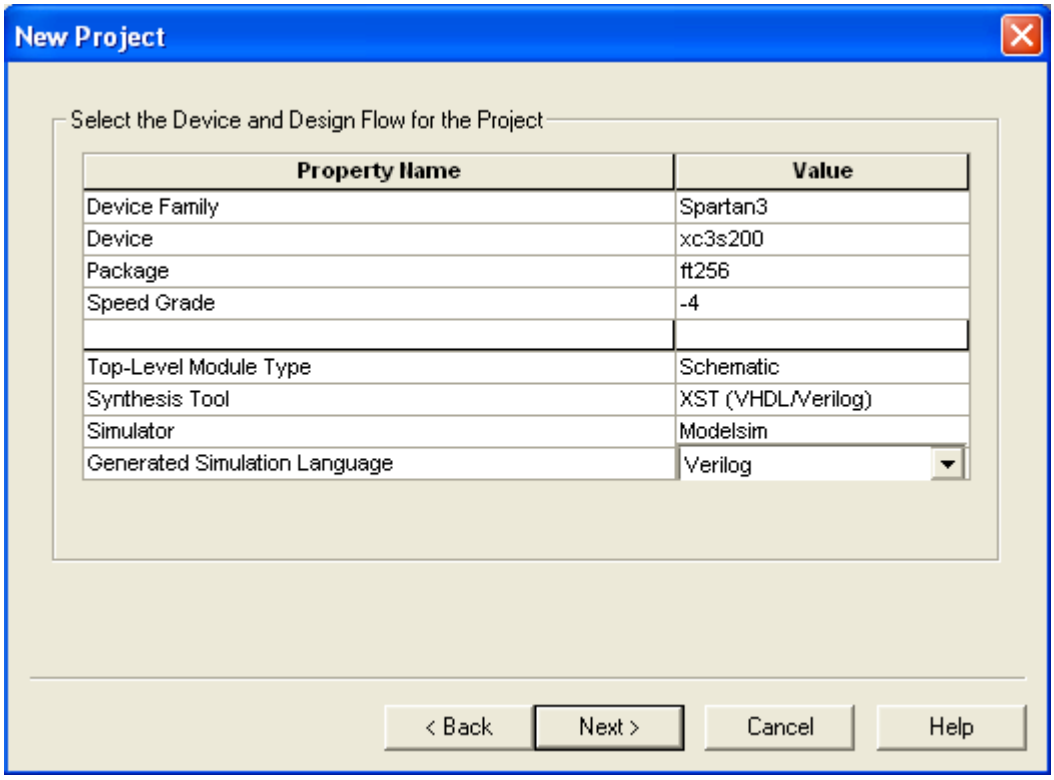

Figure B.1: Project Properties

7. Click **Next**
8. In the **Create a New Source** dialog box, click the **New Source** button. Select **Schematic** from the box on the left, and type in a file name for your project such as "lab3". Click **Next**. Click **Finish**. Click **Next**
9. In the **Add Existing Sources** dialog box, click **Next**.
10. In the **New Project Information** dialog box, click **Finish**.

ISE creates and displays the new project in the **Sources in Project** window, and opens the lab3.sch file in the Xilinx tool for creating and editing schematic diagrams, Engineering Capture System (ECS).

2. Schematic Design Entry

This section demonstrates how to create a schematic that contains logic gates. It describes how to wire them together, add net names to the wires, and add I/O markers to show where signals enter or exit the schematic. You may refer to Figure B.2 in this section as a reminder of what you need to build.

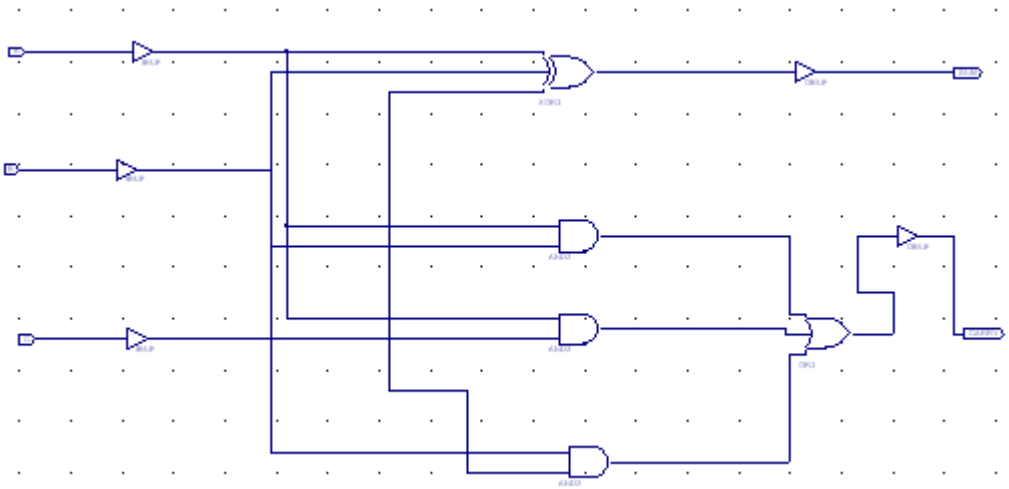

Figure B.2: The circuit to be designed

1. Create a Top-Level Schematic.

   ECS is already launched and a blank sheet is open in an ECS schematic window. If ECS is not launched, then double click on the schematic in the Sources in Project window. In ECS, you will create a schematic diagram from scratch.

2. Add a 2 input AND gate.
   - Select **Add, Symbol** or click the **Add Symbol** icon in the Tools toolbar (looks like a gate with a resistor below t).
   - Select **Logic** from the list of **Categories**.
   - Select and2 from the list of **Symbols**.
   - Place one AND gate on the schematic. Click the left mouse button to place the gate on the schematic where the cursor sits.
   - Press Esc to exit **Add Symbol** mode and restore your select tool.

3. Add any other gates (xor3, or3) you might need to make your schematic.

4. If you like, adjust your view using the **Zoom** option (**View, Zoom, In**) and the scroll bars in ECS.
5. Now we need to wire the schematic. When wiring the schematic symbols, some wires interconnect the modules and others are extended and left hanging for I/O.

    - To activate the drawing tool, select **Add, Wire** or select the **Add Wire** icon from the Tools toolbar (looks like a pencil drawing a wire).
    - To add a hanging wire or to extend the wire:
        - Click and hold the mouse button at the vertex of a pin or simply click on the pin. When your mouse is over a pin, a box will appear.
        - Drag the mouse to extend the wire to the desired length or click at the point you wish to connect to.
        - Release the mouse button at the location you want the wire to terminate or double click.
    - When finished wiring, press **Esc** to exit **Add Wire** mode.

6. Add I/O buffers

    IO buffers are used by the schematic capture tool to understand which internal signals are connected to the pins of the physical device that will be used. You have three input wires and one output wire. You will need an input buffer for each input and an output buffer for the output.

    - Select **Add** -> **Symbol** or click the **Add Symbol** icon from the **Tools** toolbar.
    - Select **IO** from the menu
    - Select **ibuf** and place 3 input buffers on the left side of your schematic
    - Select **obuf** and place 1 output buffer on the right side of your schematic
    - Add a hanging wire to the left side of each input buffer and the right side of each output buffer
    - Wire the inputs and outputs into your logic

7. Add Net Names to Wires:

    After wiring the schematic symbols, you are ready to add net names to the wires. Net names should only be added to nets that will be directly connected to I/O pins. (That is, before input buffers and after output buffers.) Note: net names are only for you to keep track of nodes easily and will not affect the behavior of your design. Adding net names to the IO will make it easier for you to assign IO pins later.

    - Select **Add, Net Name** or click the **Add Net Name** icon from the Tools toolbar.
    - To create and place a net name for each hanging wire:
        - Type the net name in the text box in the **Options** tab, located to the left of the screen.
            - Note: leave the default options as **Name the branch** and **Keep the name**.
        - Place the cursor, which now displays the net name, at the end of the hanging wire.

- Click the left mouse button.

- Name A B, and C as inputs and Z as the output.

8. Add I/O Markers

   IO markers are needed by the design tool to synthesize the design. They give a logical connection for the synthesis tool to understand that the internal signal will be passed outside either the chip or schematic. It is very important that the correct type of IO marker be used. Putting an input IO marker on an output buffer will cause an error.

   - Select **Add, I/O Marker** or click the **Add I/O Marker** icon from the Tools toolbar.
   - Add input markers to the A, B, C inputs:
     a) Select the **Add an input marker** radio button on the **Options** tab.
     b) Place the cursor, which now displays the input graphic, at the end of the input wire.
     c) Click the left mouse button to add the marker. (Note: You can also label all of your inputs or outputs at once by drawing a box around the nodes you wish to label.)

     The input graphic is added to the end of the wire, around the net name.
   - Add an output marker to the Z output:
     a) Select the **Add an output marker** radio button on the **Options** tab.
     b) Place the cursor, which now displays the input graphic, at the end of the output wire.
     c) Click the left mouse button to add the marker. The output graphic is added to the end of the wire, around the net name.

Your schematic is complete. Save the schematic diagram using **File, Save**. Then exit ECS.

3. Behavioral Simulation

ISE provides an integrated flow with the ModelTech ModelSim simulator that allows simulations to be run from the Xilinx Project Navigator graphical user interface (GUI).

In this section, we will introduce the concept of test bench and show how to verify the function of our circuit by behavioral simulation.

What is a test bench?

A test bench supplies stimuli to the design, observes the outputs of the design, and compares the observed outputs with the expected values. If any mismatch happens, the test bench issues certain messages signifying that there are errors in the design. Figure B.3 shows the concept of test bench.

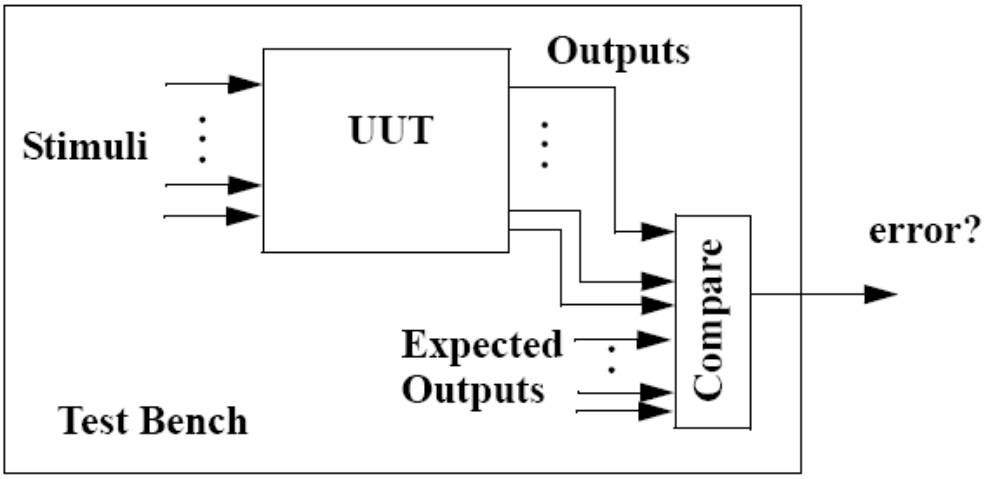

Figure B.3: A conceptual diagram of the test bench

1. In your **Project Navigator** window, click on your schematic file ***lab3 (lab3.sch)*** to make it active. Now select **Project → New Source**. In the window that opens up select the option **Test Bench Waveform**. Specify a name for the waveform in the **File Name** field and click on **Next**. In the following window click on **Next** and then finally click **Finish**.

2. In the **Initialize Timing** window (Figure B.4), select the option **Combinatorial Design**. In the input boxes after **Check outputs and Assign Inputs**, enter the value 25, **Initial Length of Test Bench** to 1000 ns and set the time scale to **ns** as shown below and click **OK**. This will open the **HDL Bencher** window (Figure B.5).

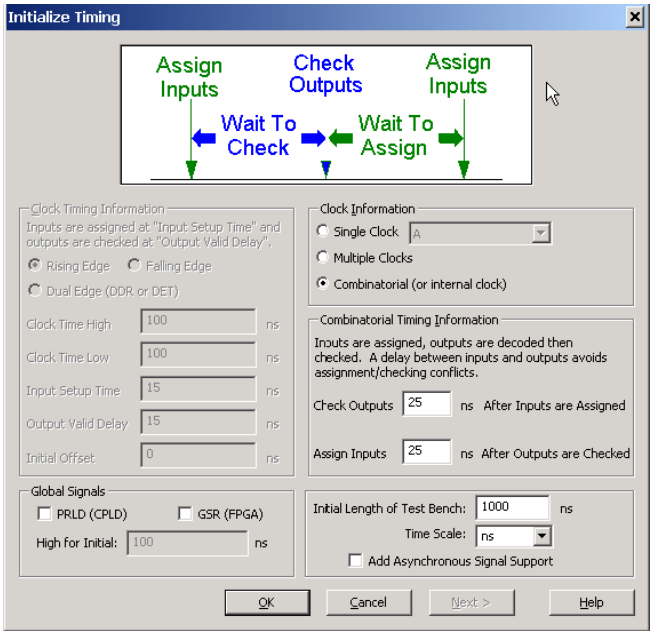

Figure B.4: Initialize Timing Window

3. Click **OK** and you'll see the waveform window of the test bench. The three input signals are marked cyan while the two output signals are marked yellow. By directly clicking on the waveform you can change the values of the signal. Just play around a little to get familiar with it. Now specify the waveforms of the three inputs as shown in Figure B.5. Notice that the waveforms of *A*, *B* and C cover all possible 8 combinations. For each of the eight combinations, draw the expected outputs on the waveforms of *CARRY* and *SUM*. Recall that the outputs are supposed to be 25ns later than the inputs as we specified in the previous window. Save the waveforms after you're done.

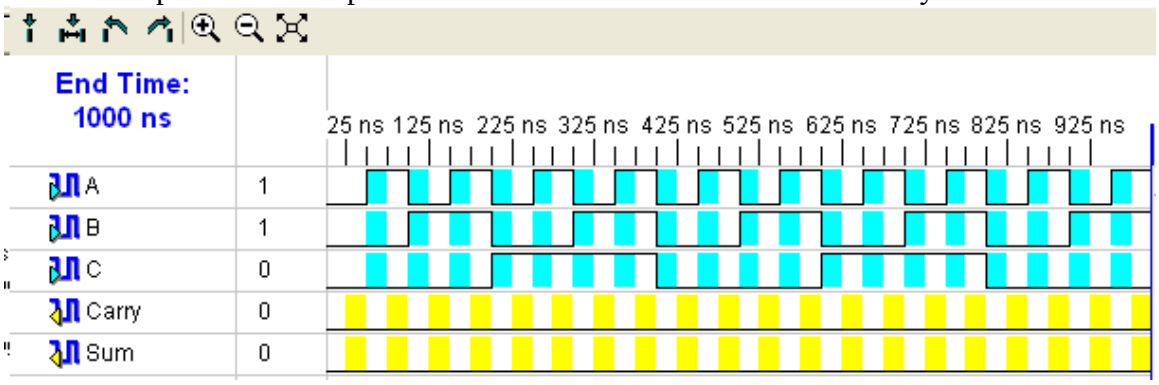

Figure B.5: Test Bench Waveforms

4. Now go back to the ***Project Navigator*** windows, and make sure that you have the test bench waveform you just created selected in ***Sources in Project*** window. In the ***Process View*** window, double click ***Simulate Behavioral Model***. This will open up ModelSim simulation windows and run the test bench simulation. If ModelSim fails to start, you need to go back to check the license.

5. Right-click on the waveforms and select ***Zoom Range***. Choose ***Start*** as 1 ns and ***End*** as 1000 ns.

6. Check the waveforms to see whether there are any errors. In particular, pay attention to signal ***tx_error*** in the ***Objects*** window. ***tx_error*** counts how many errors are detected in the simulation. In this case, ***tx_error*** is 0 meaning everything looks fine.

4. Assigning Package Pins

We will use the switches and LEDs that are already on the Spartan-3 Board to implement the design.

1. Choose input and output pins on your Digital Logic Board. See appendix A or for the pinout diagram. Refer to Appendix B to find which pins on the FPGA are connected to the switches and LEDs on board. Pick two LED pins (output) and three switch pins (inputs).
2. Click on **Assign Package Pins** under **User Constraints.** This will launch the Pin-out Area Constraint Editor (PACE).

3. In PACE, select the **Package View Tab** to open the Package Pins window. This window shows the graphical representation of the device package.
4. The **Design Object List** window shows all the IOs in the design.
5. In the Design Object List window, click and drag the input and output signals to the specific locations which you have assigned in Step 1. A sample assignment is shown in Figure B.6.

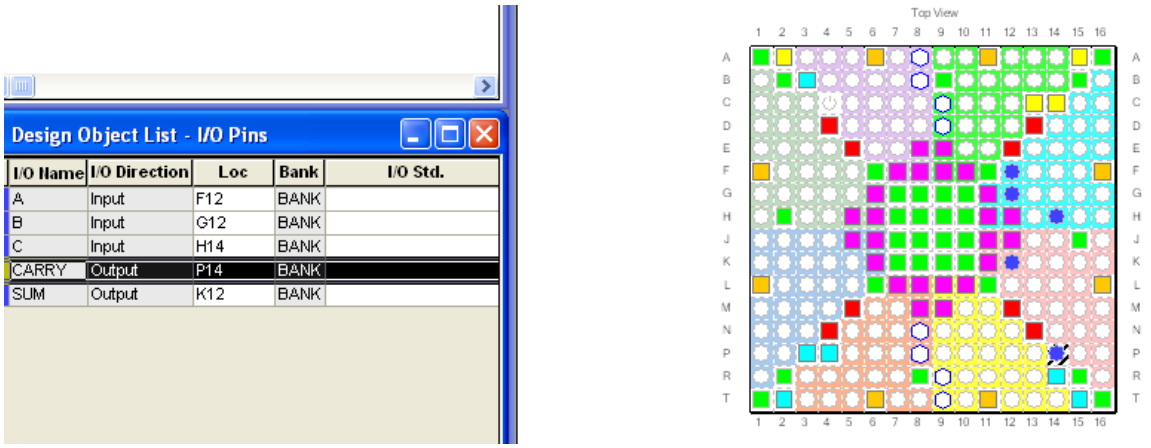

Figure B.6: Assigning pins in PACE

6. Once the pins are locked down, select **File → Save**. The changes made in PACE are saved in the *lab3.ucf* file in your current working directory.
7. Exit PACE.

5. Design Implementation

Design implementation covers running the Implement Design process in **Project Navigator**.

Note: For more information about implementing a design, see ISE Help. Select **Help, ISE Help Contents**, expand either the FPGA or CPLD hierarchy in the left pane and expand the **Implementing a Design** hierarchy.

1. Run Implement Design

   First, run all processes (Synthesis through Place & Route) associated with the design. To do so, run Implement Design on the schematic file:

   - Select lab3.sch in the **Sources in Project** window.
   - Double-click **Implement Design** in the **Processes for Source lab3** window. This runs all processes. Be patient – this takes a while!

A check in the **Processes for Source** window denotes a process that was run successfully. An exclamation indicates that the process was run and that there is a warning for the process. More information about warnings can be obtained in the **Transcript** window.

6. Timing Analysis

To see the timing report, go to ***Implement Design->Place&Route->Generate Post-Place&Route StaticTiming->Text-based Post-Place&Route Static Timing Report***. The timing report will be shown in the right window similar to Figure B.7. From the timing report, we see that the critical path (i.e. worst case delay) is from C to Carry with a delay of 7.851ns.

```
18
19  INFO:Timing:2698 - No timing constraints found, doing default enumerat
20  INFO:Timing:2752 - To get complete path coverage, use the unconstraine
21     option. All paths that are not constrained will be reported in the
22     unconstrained paths section(s) of the report.
23
24
25  Data Sheet report:
26  -----------------
27  All values displayed in nanoseconds (ns)
28
29  Pad to Pad
30  ---------------+---------------+---------+
31  Source Pad     |Destination Pad|  Delay  |
32  ---------------+---------------+---------+
33  A              |Carry          |    7.817|
34  A              |Sum            |    7.802|
35  B              |Carry          |    7.596|
36  B              |Sum            |    7.575|
37  C              |Carry          |    7.851|
38  C              |Sum            |    7.745|
39  ---------------+---------------+---------+
40
41  Analysis completed Wed Feb 22 15:15:45 2006
42  --------------------------------------------------------------------------
```

Figure B.7: Timing Report of the circuit

7. Timing (post-place-and-route simulation)

Timing simulation uses the block and routing delay information from a routed design to give a more accurate assessment of the behavior of the circuit under worst-case conditions. For this reason, timing simulation is performed after the design has been placed and routed.

In order to simulate the design, a test bench is needed to provide stimulus to the design. You should use the same test bench that was used to perform the behavioral simulation.

To set the simulation process properties:
1. In the Sources in Project window, select the test bench file.
2. In the Processes for Source window, click the + next to ModelSim Simulator to expand the process hierarchy.

To start the timing simulation, double-click **Simulate Post-Place and Route Verilog Model** in the Processes for Source window. ISE will run NetGen to create the timing simulation model. ISE will then call ModelSim and create the working directory, compile the source files, load the design, and run the simulation for the time specified.

To view signals during the simulation, you must add them to the Wave window. ISE automatically adds all the top-level ports to the Wave window. Additional signals are displayed in the Signal window based on the selected structure in the Structure window.
There are two basic methods for adding signals to the Simulator Wave window.

• Drag and drop from the Signal/Object window.

• Highlight signals in the Signal/Object window and then select **Add → Wave → Selected Signals**.

Right-click on the waveforms and select ***Zoom Range***. Choose ***Start*** as 1 ns and ***End*** as 1000 ns. Zoom in on the waveforms and find the exact delay on the longest path.

8. Programming the Digilent Board

1. Turn on your Digilent Board and make sure that the cable is properly connected.
2. Double-click **Generate Programming File** to create a bitstream of this design.
3. The BitGen program creates the design_name.bit bitstream file (in this design, the lab3.bit file). The bitstream file contains the actual configuration data.
4. Double-click on **Configure Device.** This lunches the iMPACT software.
5. Select **Boundary Scan Mode** in the Configure Devices dialog box.
6. Click **Next**.
7. Select **Automatically connect to cable and identify Boundary-Scan chain** in the Boundary Scan Mode Selection dialog box.
8. Click **Finish**.

The configuration file is used to program the device. When the software prompts you to select a configuration file for the device XC3S200:

1. Select the BIT file from your project working directory.
2. Click **Open**.
3. If the software gives the option of assigning a new configuration file, Click **Cancel.**
4. Right-click on the XC3S200 device
5. Select **Program** from the right-click menu.
6. Uncheck **Verify**. Click **OK.**

When the Program operation completes, a large blue message appears showing that programming was successful.

Your design has now been programmed. The board should now be working and should allow you to use the switches for providing the inputs to the adder. Two LEDs should show the CARRY and SUM bits.