iCEcube2 Tutorial

v1.2 – August 26, 2014

Copyright

Trademarks

Disclaimers

Contacting Technical Support

FAQs The first place to look. The Answer Database provides solutions to questions that many of our customers have already asked. Lattice Applications Engineers are continuously adding to the Database.

Telephone Support Hotline Receive direct technical support for all Lattice products by calling Lattice Applications from 5:30 a.m. to 6 p.m. Pacific Time.

For USA & Canada: 1-800-LATTICE (528-8423)
For other locations: +1 503 268 8001

In Asia, call Lattice Applications from 8:30 a.m. to 5:30 p.m. Beijing Time (CST), +0800 UTC. Chinese and English language only.

For Asia: +86 21 52989090

E-mail Support

techsupport@latticesemi.com
techsupport-asia@latticesemi.com

For Local Support Contact your nearest Lattice Sales Office

Revision History

The following table lists the revision history of this document.

Version	Revision
1.0	Release iCEcube2 2010.03
1.1	Updated for iCEcube2 2013.12. Added content to use stand-alone Diamond Programmer to program iCEblink40-HX1K Evaluation Board. Added Appendix A to describe Lattice Synthesis Engine.
1.2	Updated Trademarks.

Preface

About This Document

The *iCEcube2 Tutorial* provides iCE FPGA designers with an overview of the software tool and the design process using iCEcube2. This document covers the iCEcube2 tools for project setup, navigation, and physical implementation on the iCE40 FGPA device.

For detailed information of the iCEcube2 development tools. refer to the *iCEcube2 User Guide*, located in the <*icecube2_install_directory*>/doc directory.

For information on the Synopsys Synplify Pro software, refer to the Synplify Pro documentation provided in the <*icecube2_install_directory*>/synpbase/doc directory.

For information on the Aldec Active-HDL design tool, refer to the Active-HDL documentation provided in the <*icecube2_install_directory*>/Aldec/Active-HDL/Books directory.

Software Version

This tutorial is based on iCEcube2 Software Version 2013.12. For more information about acquiring the iCEcube2 software, visit the Lattice Semiconductor website: http://www.latticesemi.com.

Software Requirements

This tutorial requires two software programs:

- iCEcube2 Tool Suite version 2013.12 or higher.
- The stand-alone version of Lattice Diamond Programmer software, version 2.1 or higher.

The iCEcube2 software should be installed on a platform satisfying the following minimum requirements:

- Pentium 4 computer (500 MHz) with 256 MB of RAM, 256MB of virtual memory, and running one of the following operating systems :
  - Windows 7 OS, 32-bit / 64-bit
  - Windows XP Professional
  - Red Hat Enterprise Linux WS v4.0

For installation help, refer to the *iCEcube2 Installation Guide* located in the <*icecube2_install_directory*>/doc directory.

Programming Hardware Requirements

- iCEblink40-HX1K Evaluation Kit, which consists of the following:
  - iCEblink40-HX1K Board
  - USB Mini Cable to provide 5 V power to the board.
- Lattice HW-USBN-2A USB programming cable

Refer to http://www.latticesemi.com for more details on programming hardware.

Overview

iCEcube2 Tool Suite

The iCEcube2 Tool Suite is comprised of several integrated components, running under either the Microsoft Windows or the Red Hat Linux environments.

Figure 1 depicts the design flow using the iCEcube2 Tool Suite. The blue boxes indicate functionality supported by iCEcube2 software. The purple boxes indicate the functionality supported by Synopsys' Synplify Pro synthesis tools, Lattice Synthesis Engine (LSE), and Aldec Active-HDL simulation tool. The iCEcube2 software, Synopsys Synplify Pro, LSE, and the Aldec Active-HDL software constitute the iCEcube2 Tool Suite.

Note: The Aldec Active-HDL tool is available only in Windows environments.

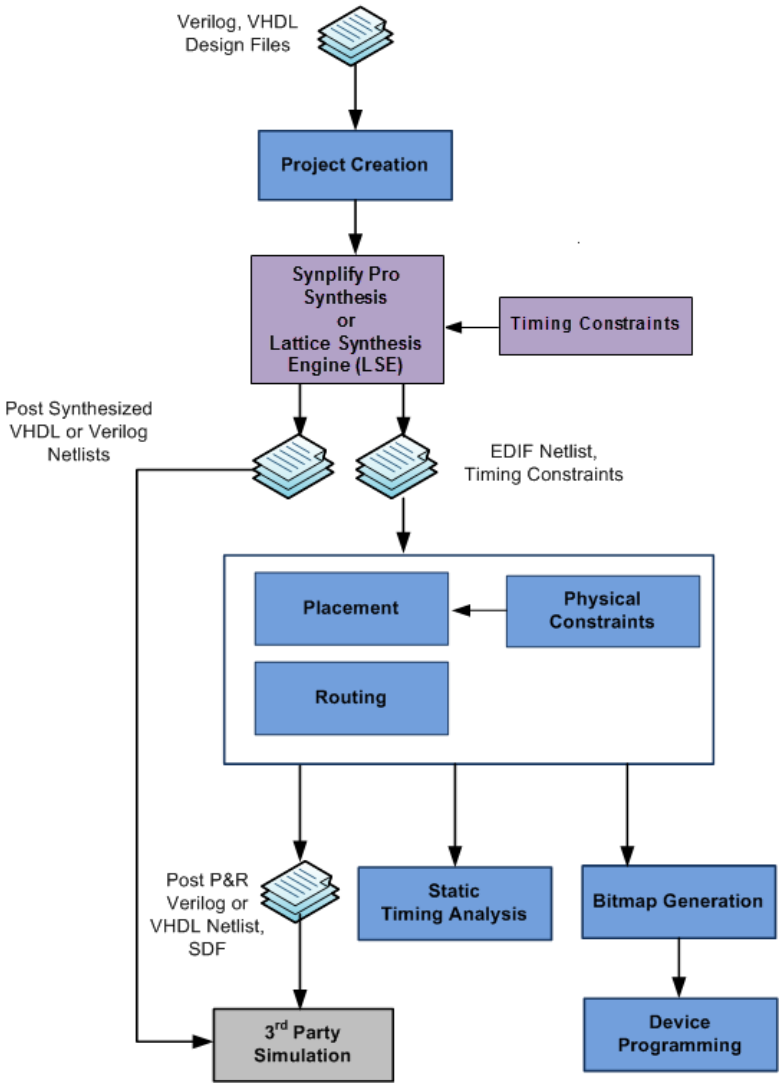

Figure 1: The iCEcube2 Design Flow

Design Flow

The following steps provide an overview of the design flow using the iCEcube2 Tool Suite.

- Create a new project in the iCEcube2 Project Navigator and specify a target device and its operating conditions. Add HDL (Verilog or VHDL) design files and constraint files to the project.
- Synthesize the design. iCEcube2 software supports both Lattice Synthesis Engine (LSE) and Synplify Pro design software. For the purposes of this tutorial, we will be using Synplify Pro. "Appendix A: Synthesizing the Design with Lattice Synthesis Engine" on page 26 provides steps that demonstrate the use of Lattice Synthesis Engine (LSE).
- Perform placement and routing using iCEcube2 Place and Route tools. iCEcube2 also supports the Floorplanner tool to allow you to manually place logic cells and I/Os.
- Perform timing simulation of your design using the Aldec Active-HDL simulation tool or any industry-standard HDL simulation tool. The files necessary for simulation are automatically generated by the iCEcube2 physical implementation tools, after the routing phase.
- Perform static timing analysis using the iCEcube2 Static timing analyzer.
- Generate device programming and configuration files from the iCEcube2 Physical Implementation tools.
- Program the device on the iCEblink40-HX1K board using Lattice Stand-Alone Programmer tool.

Create a Project

Start the iCEcube2 software. In Windows, from the Start menu, choose **LatticeiCEcube2 2013.12 > iCEcube2**. The iCECube2 main window appears, as shown in Figure 2.

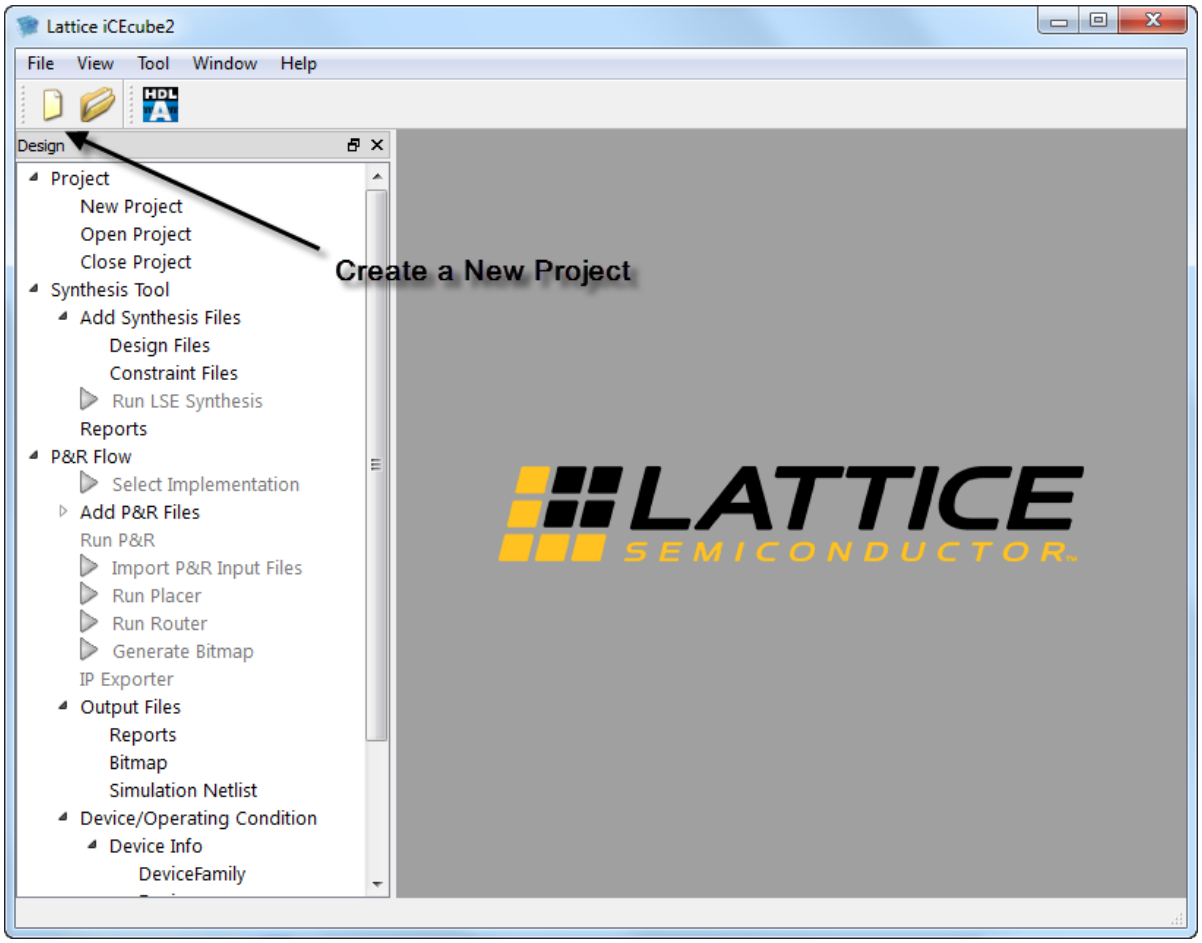

Figure 2: iCECube2 Main Window

To create a new project, choose **File > New**, or click the **Create a New Project** icon as shown in Figure 2. This will launch the New Project Wizard, a shown in Figure 3.

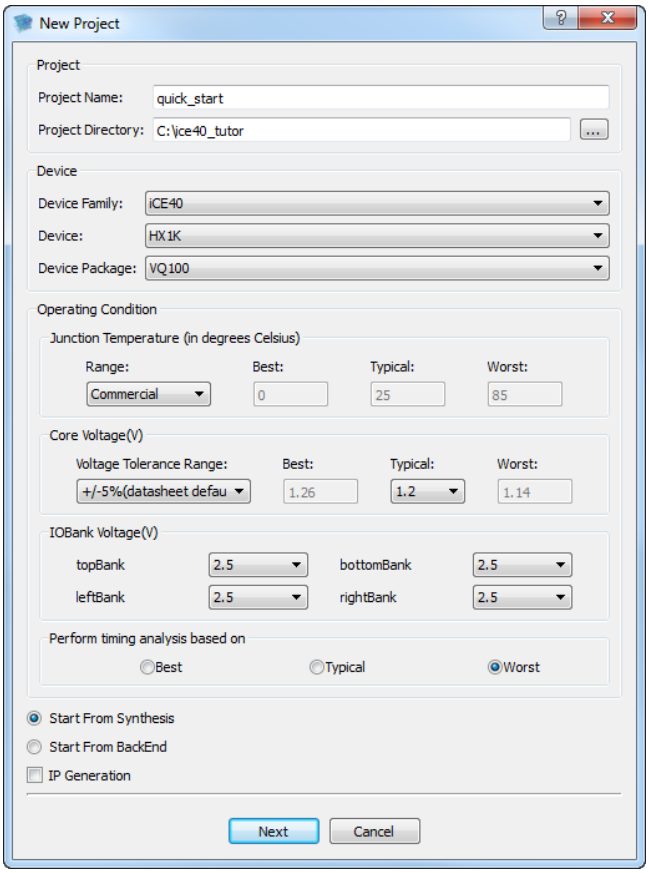

Figure 3: New Project Wizard

This example is targeted for an iCE40-HX1K device installed in the iCEblink40-HX1K Evaluation Board. Set up the project properties as follows:

- **Project Name**: Specify a project name (*quick_start*) in the Project Name field.
- **Project Directory**: Specify a directory in which to place the project. In this example, the directory C:\ice40_tutor was chosen.
- **Device Family**: Choose **iCE40** from the dropdown menu.
- **Device:** Choose **HX1K** from the dropdown menu.
- **Device Package:** Choose **VQ100** from the dropdown menu.
- **Operating Condition**: Leave all selections as default.
- **Start From Synthesis:** This option allows you to start the flow from synthesis. For this tutorial, select this option.
- Click **Next** to display the Add Files dialog box, as shown in Figure 4.

In the Add Files dialog box, in the Look in box, browse to:
< *icecube2_install_directory* >/examples/blinky.
Highlight the following files:

- iCElab.v
- iCElab.sdc

The SDC file is a Synopsys constraint file, which contains timing constraint information.

Click the **>>** button to add the selected files. The **>>>** button can be used to add all the files in the open directory. Files can be removed from the project using **<<** and **<<<**.

Click **Finish** to create the project.

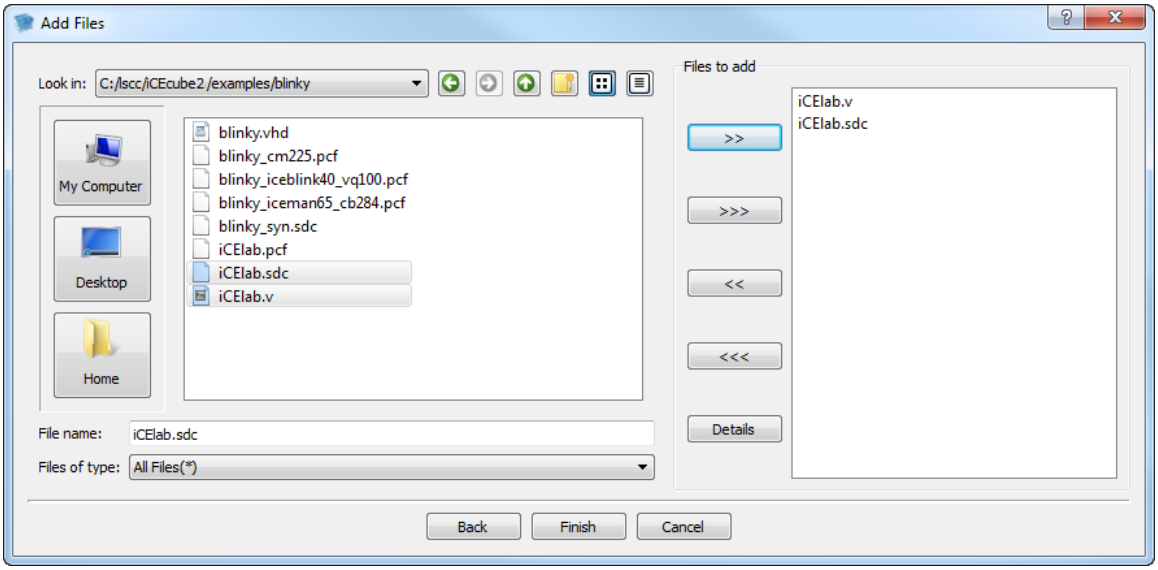

Figure 4: New Project Wizard – Add Files Dialog Box

After successfully setting-up your project, you will return to the following iCEcube2 Project Navigator, which should appear as shown in Figure 5.

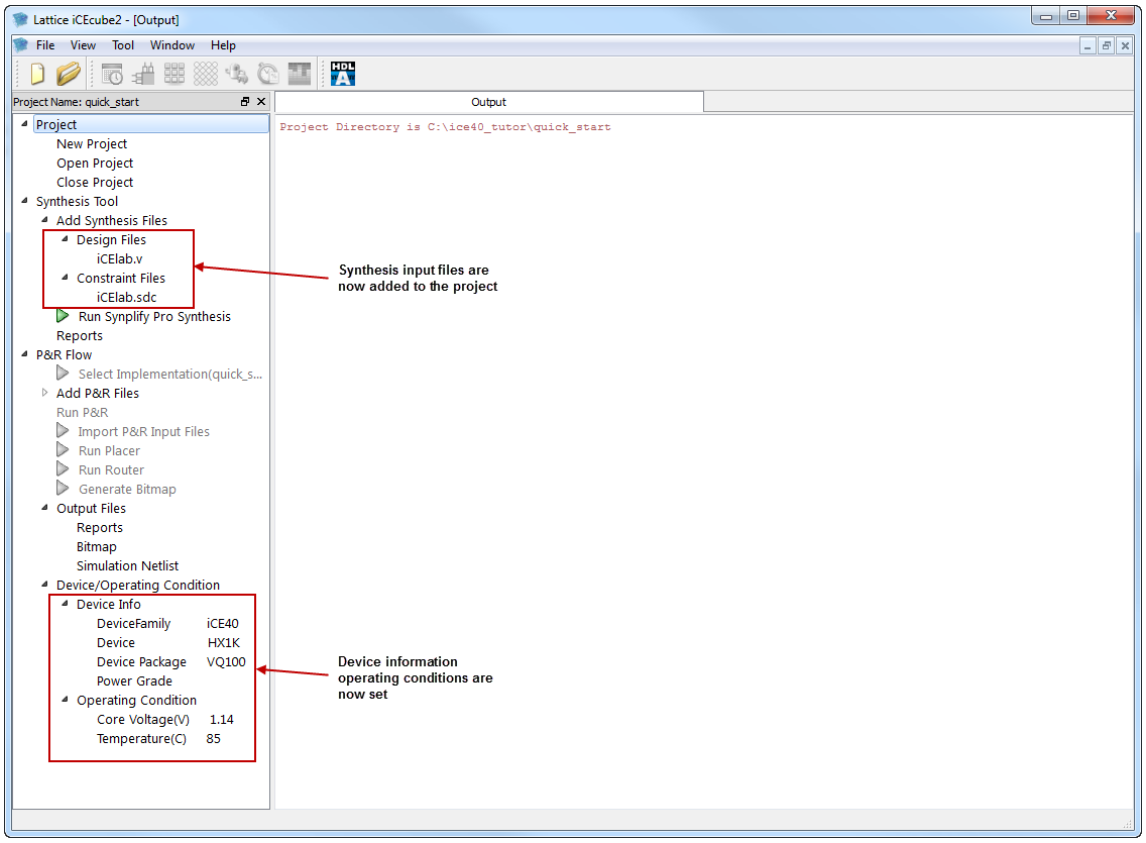

Figure 5: iCEcube2 Project Navigator View after Completing Project Set-up

Synthesize the Design with Synplify Pro Synthesis Tool

Synplify Pro is the default synthesis tool, but iCEcube2 also supports Lattice Synthesis Engine (LSE). For the purpose of this tutorial, we will use the default Synplify Synthesis Pro tool. To learn more about LSE, see "Appendix A: Synthesizing the Design with Lattice Synthesis Engine" on page 26.

To run synthesis, double-click **Run Synplify Pro Synthesis** in the project navigator window, as shown in Figure 6.

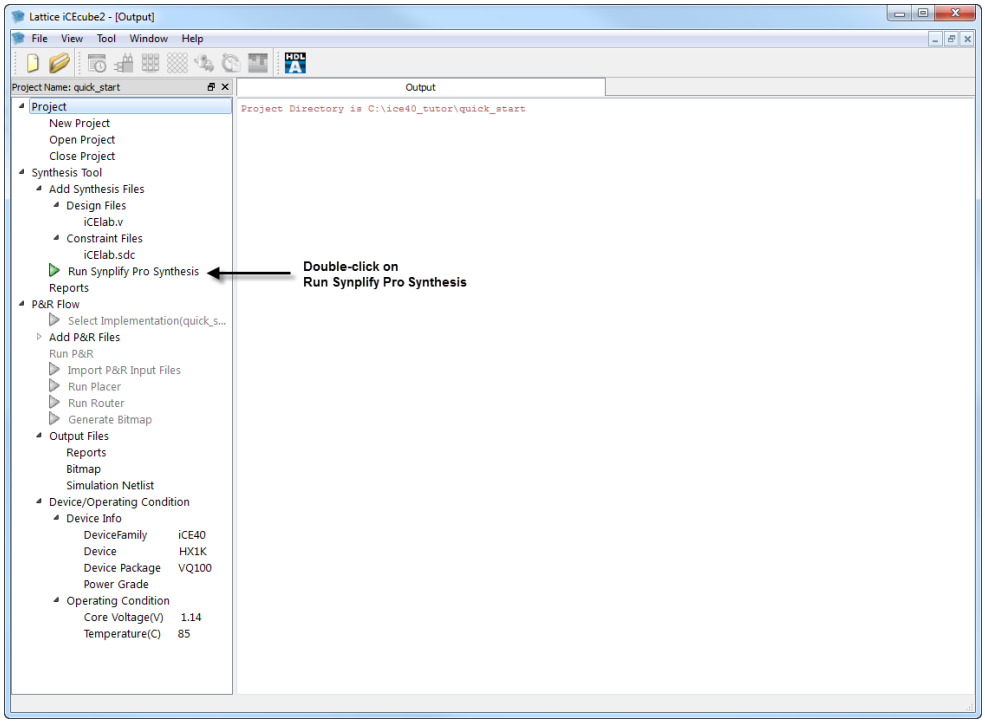

Figure 6: Launch Synthesis Tool

The synthesis is run in the background, and the results of the synthesis are displayed in the iCEcube2 output tab, as shown in Figure 7.

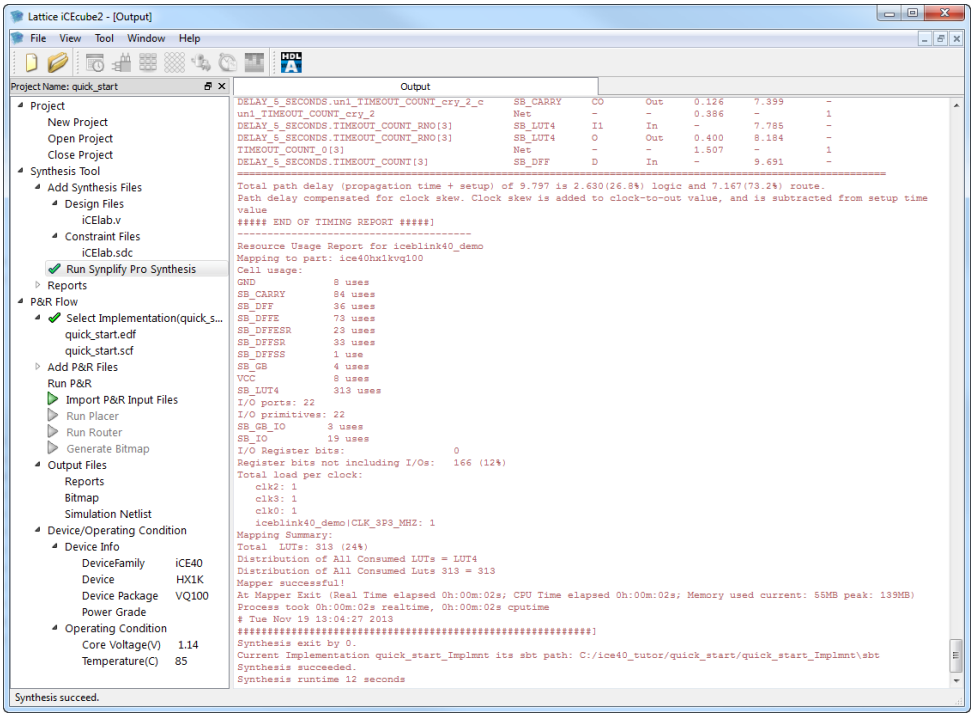

Figure 7: iCEcube2 Output Tab After Running Synthesis

Place and Route Flow

This section goes through the post synthesis physical implementation flow. This tutorial flow uses Synplify Pro synthesis tool.

Select Implementation

Double-click on **Select Implementation**. This sets the synthesis implementation to process for place and route in iCECube2. If you have different synthesis implementations, you will be able to select the synthesis implementation you wish to place and route. Since we only have one implementation, click **OK** when the Select Synthesis Implementation dialog box appears, as shown in Figure 8.

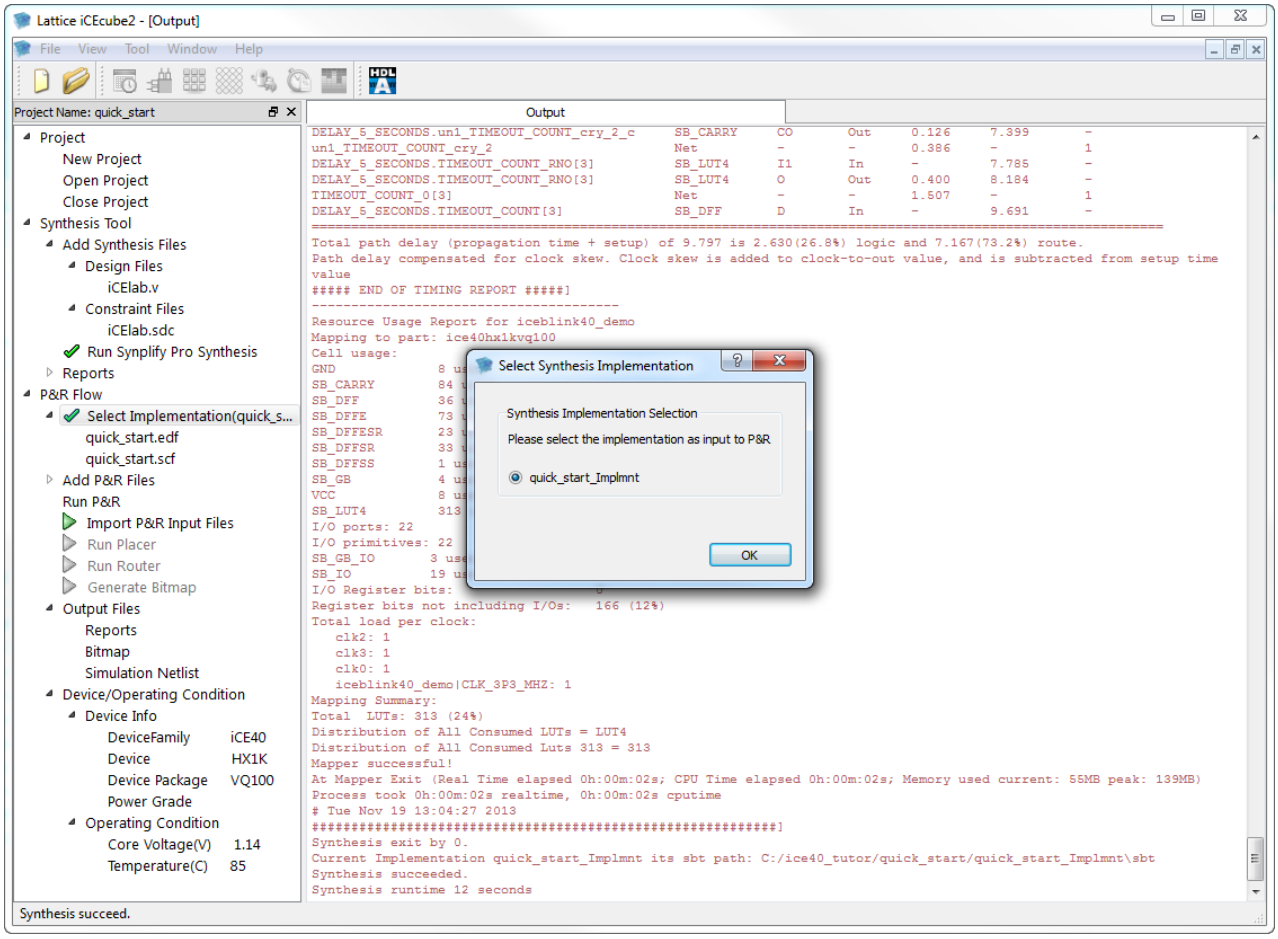

Figure 8. Select Synthesis Implementation

Import Physical Constraints

Physical constraints such as pin assignments are stored in a .PCF file (Physical Constraint File). You will now add the .PCF file to your project.

In the iCEcube2 Project Navigator, expand **Add P&R Files**. Right-click on Constraint Files, and select **Add Files** as shown in Figure 9.

Note: For information on importing physical constraints from iCEcube to iCEcube2, please refer to the **Importing Physical Constraints from iCEcube to iCEcube2** section in the *iCEcube2 User Guide*.

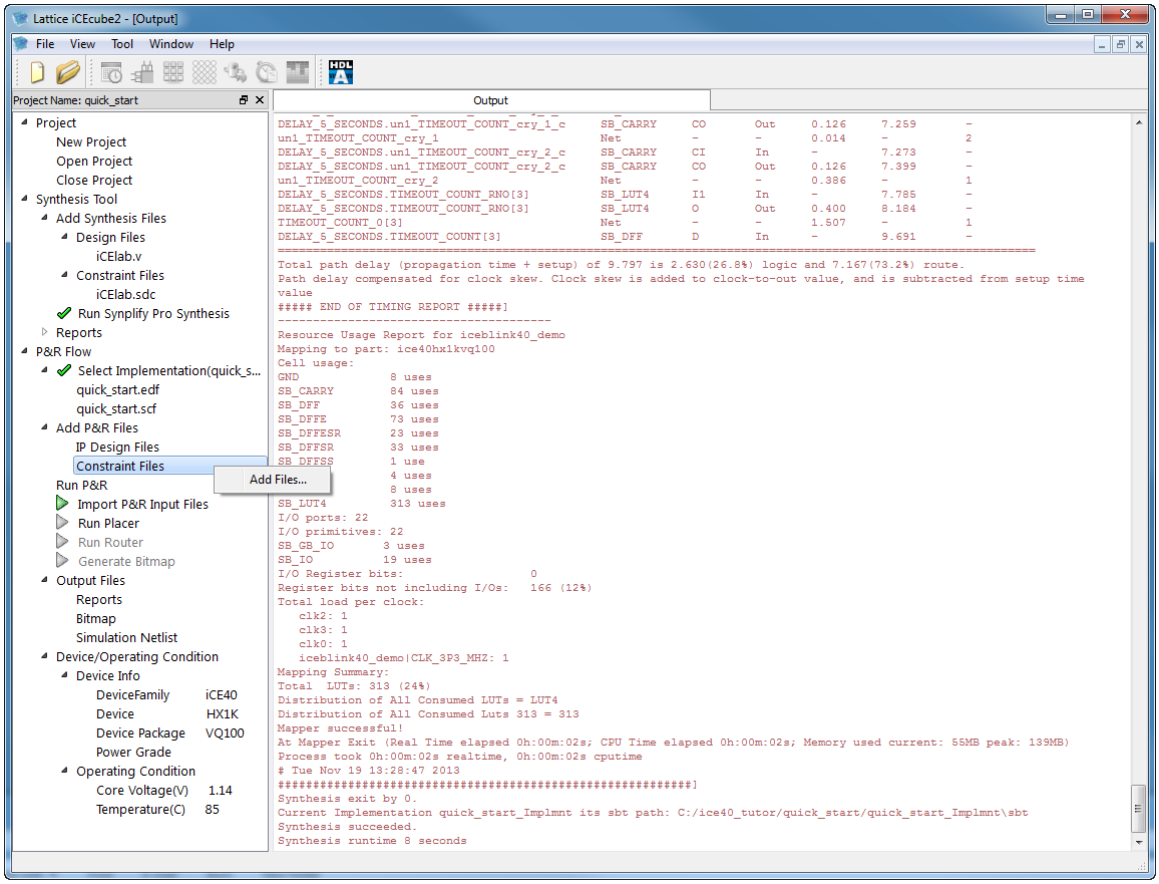

Figure 9: Specify Additional Files for Place and Route

Navigate to the < *icecube2_install_directory*>/examples/blinky directory and add the **iCElab.pcf** file as shown in Figure 10. Then, click **OK**.

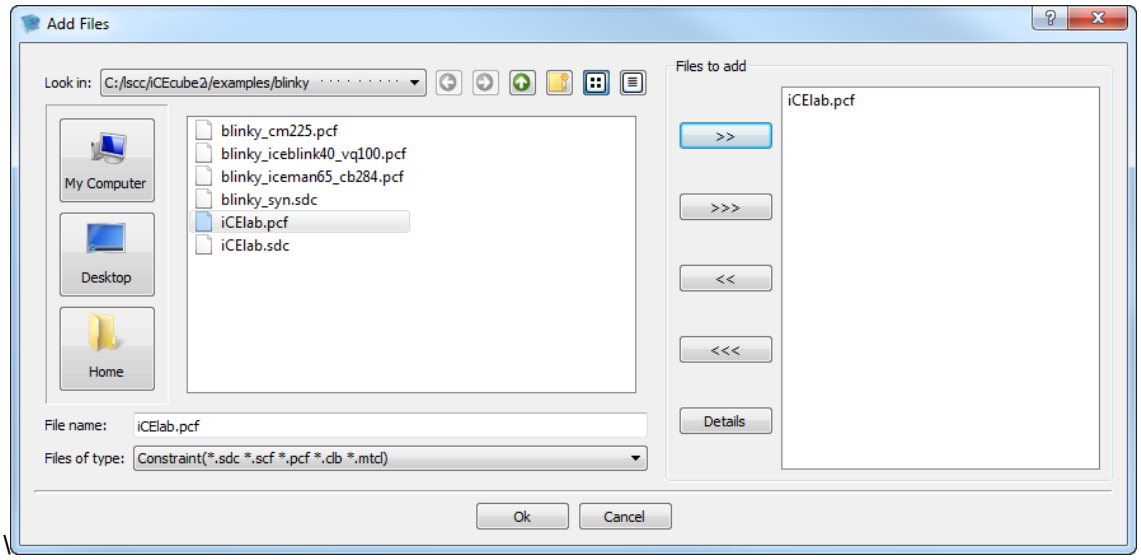

Figure 10. Add .pcf file

Import Place and Route Input Files

The next step is to import the files for Place and Route. Double-click on **Import P&R Input Files** in the Project Navigator. Once completed you will see a green check next to **Import P&R Input Files**, as shown in Figure 11.

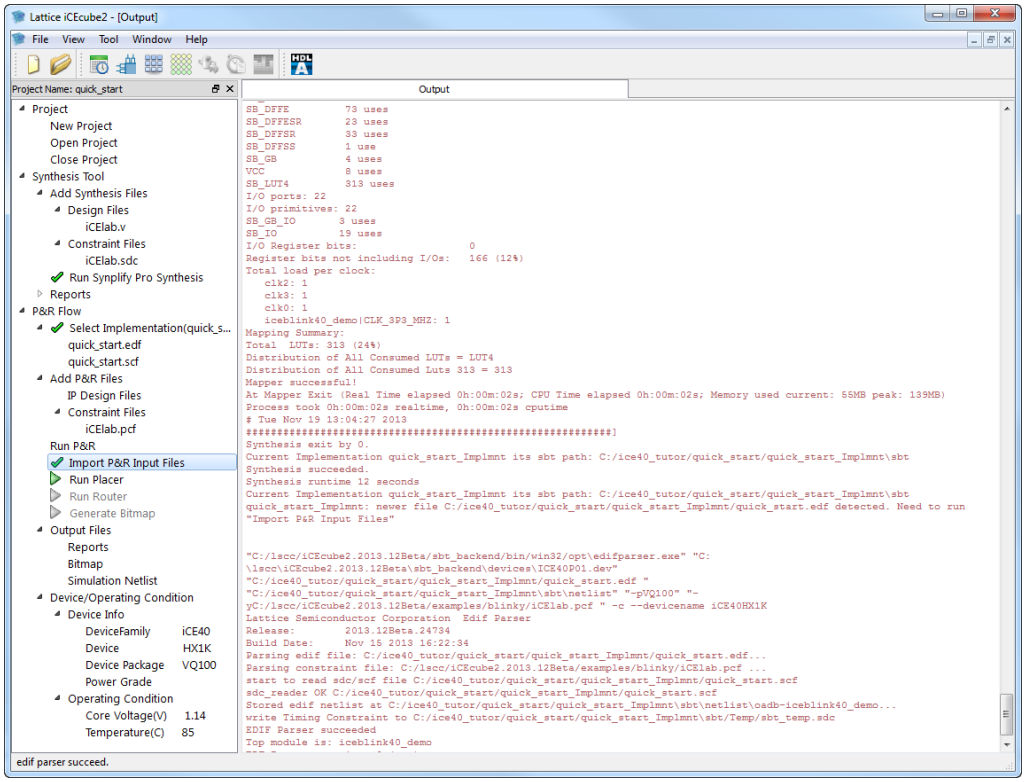

Figure 11: Successful Import of Place and Route Input Files

Place the Design

You will now run the Placer. Double-click on **Run Placer**.

Once placement is complete, a green check will appear and the Output window will show information about the placement of the design, as shown in Figure 12.

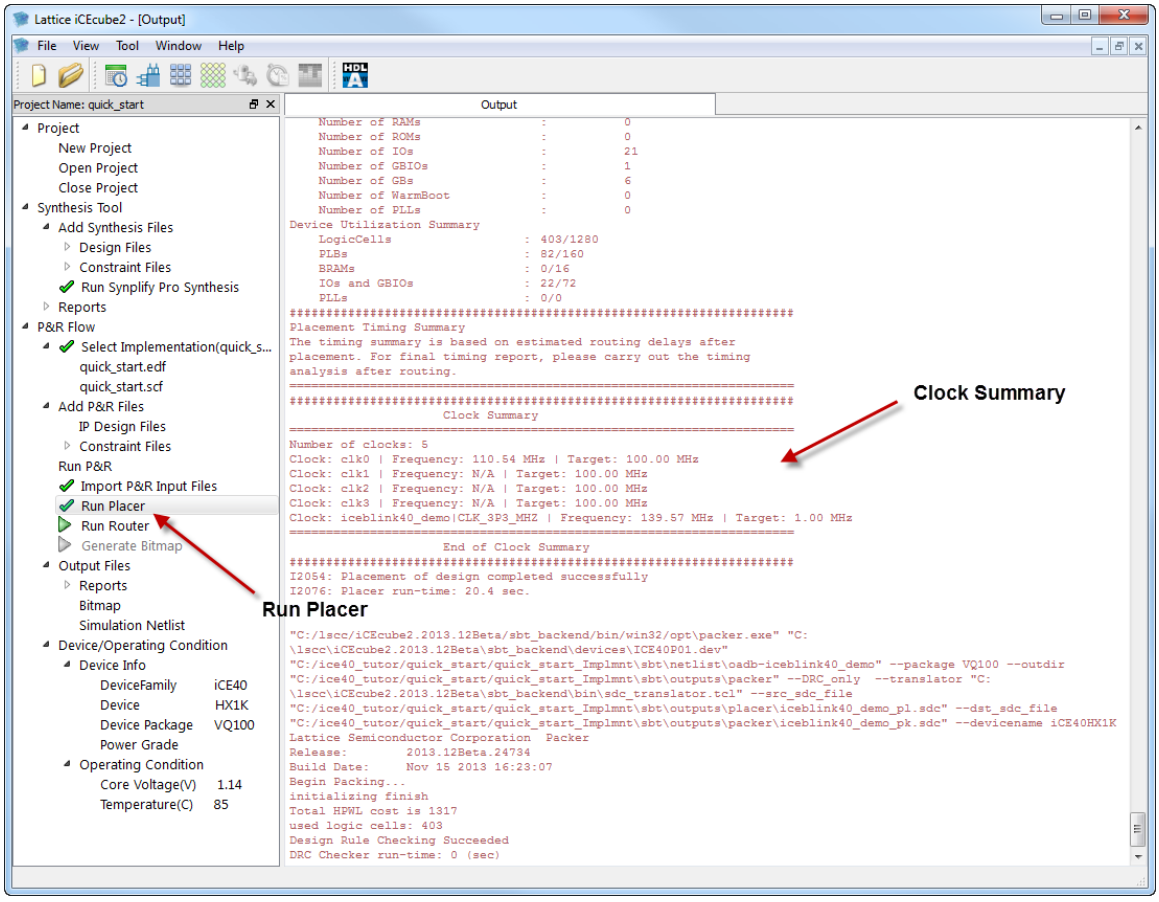

Figure 12: Run Placer

View Floorplanner

You can view the placement of the design by opening the Floorplanner.

Choose **Tool > Floor Planner,** or click the Floorplanner icon, shown in Figure 13Figur.

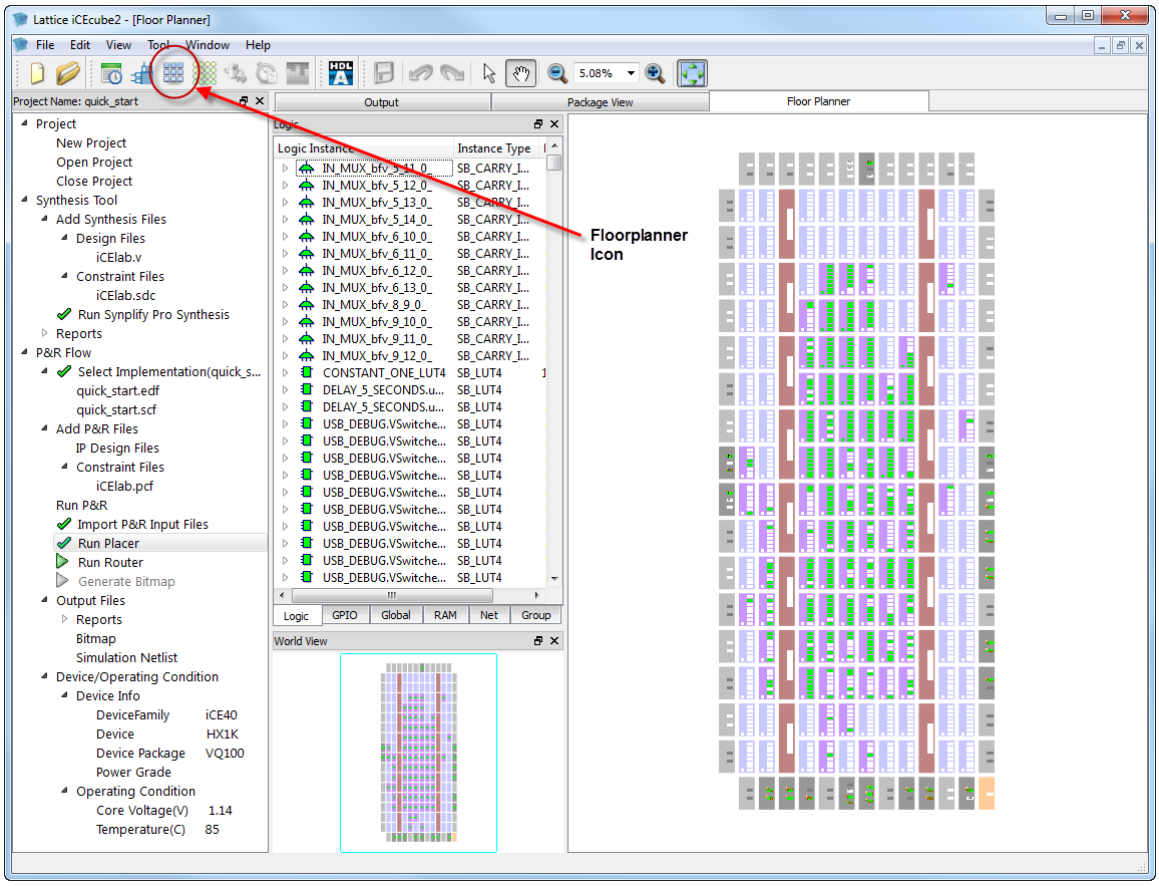

Figure 13. Floorplanner

View the Package Viewer

You can also see how pins were placed for your design by selecting the Package Viewer. You can select the package viewer by going to the menu and selecting **Tool > Package View** or click the Package View Icon, shown in Figure 14.

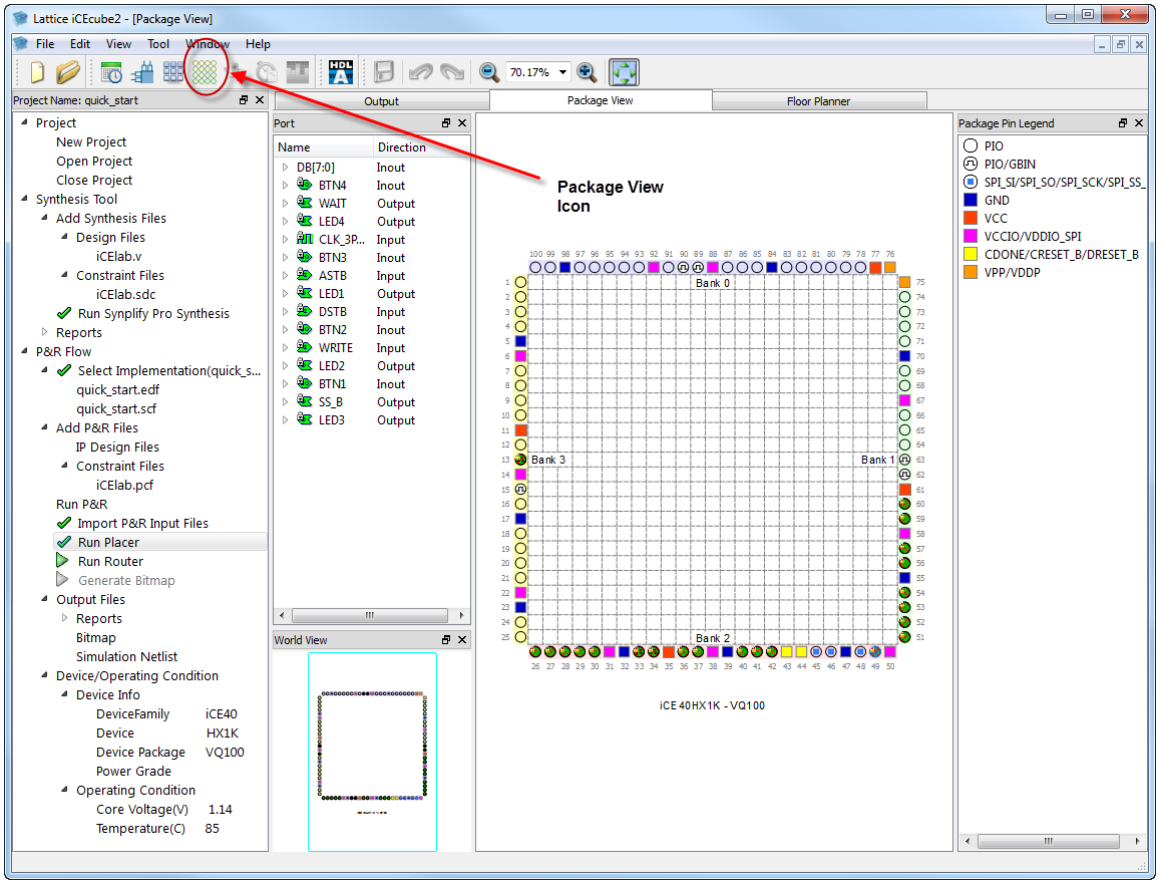

Figure 14: Package View

Route the Design

Double-click on **Run Router** in the project navigation window. Place and Route have been separated into different steps as to allow you to re-route the design after making placement modifications in the Floorplanner without having to re-run the placer.

Perform Static Timing Analysis

Now that after you have routed the design, you can perform timing analysis to check to see if the design is meets your timing requirements. To launch the Timing Analyzer, choose **Tool > Timing Analysis**, or click Timing Analysis icon. See Figure 15.

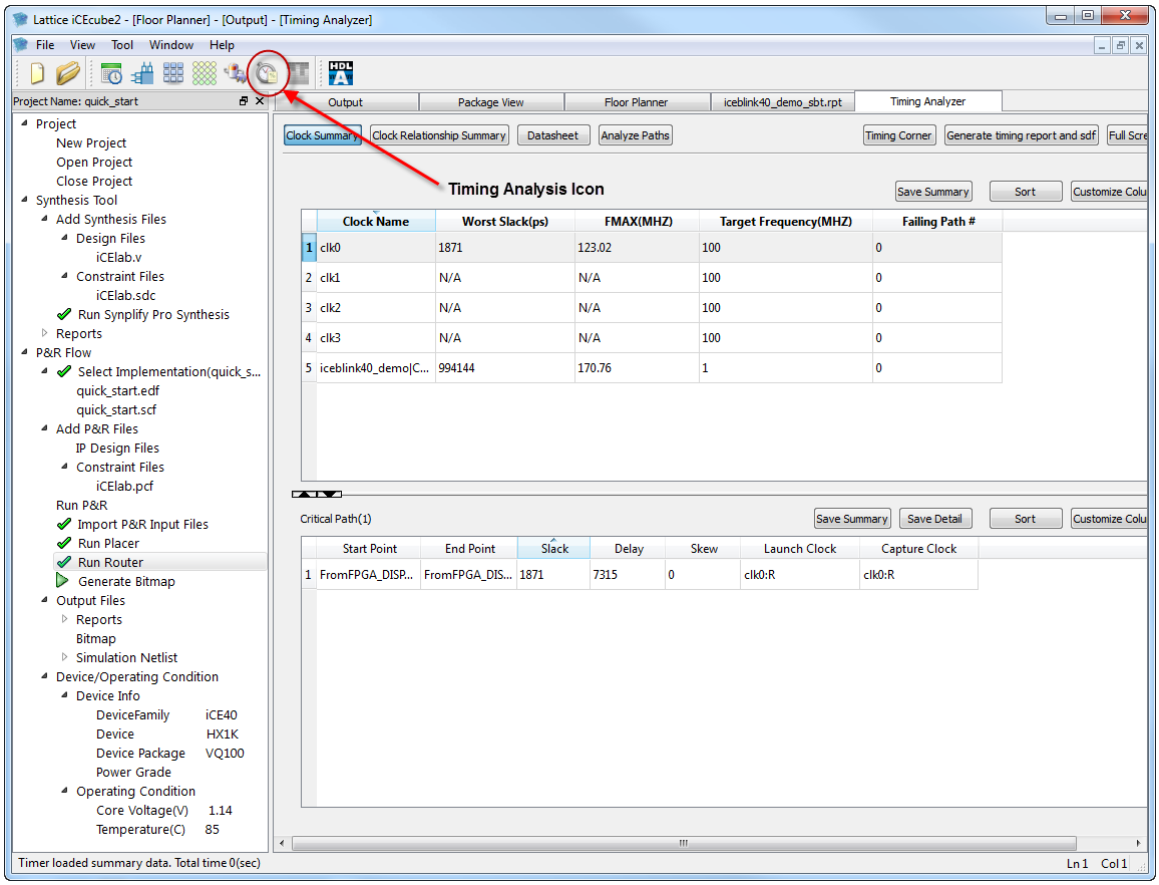

Figure 15. Timing Analysis Summary

Figure 15 shows the timing analysis of our 100 MHz design running at over 123.02 MHz and our 1 MHz clock is running at over 170.76 MHz (worst case timing). If we were not meeting timing, the timing analyzer will allow you to see your failing paths and do a more in-depth analysis. For this tutorial, we won't go into details on timing slack analysis. See the *iCEcube2 User Guide* for details on timing analysis.

Perform Power Analysis

iCEcube2 comes with power estimator tool. To launch the power Estimator, go to the menu and select **Tool > Power Estimator**, or click the Power Estimator icon, as shown in Figure 16.

There are multiple tabs in the Power Estimator tool including Summary, I/O, and Clock Domain. On the Summary tab, if it is not already set, change the **Core Vdd** to **1.2V** and make sure all **IO Voltages** are set at **2.5V**. Then click **Calculate**. The estimator will update with power information for both static and dynamic power. You can change voltages in the **IO Voltage** values in the respective drop down menus and re-calculate to see the different results. For more information on using the I/O and Clock Domain tabs, refer to the Power Estimator tool section of the *iCEcube2 User Guide*.

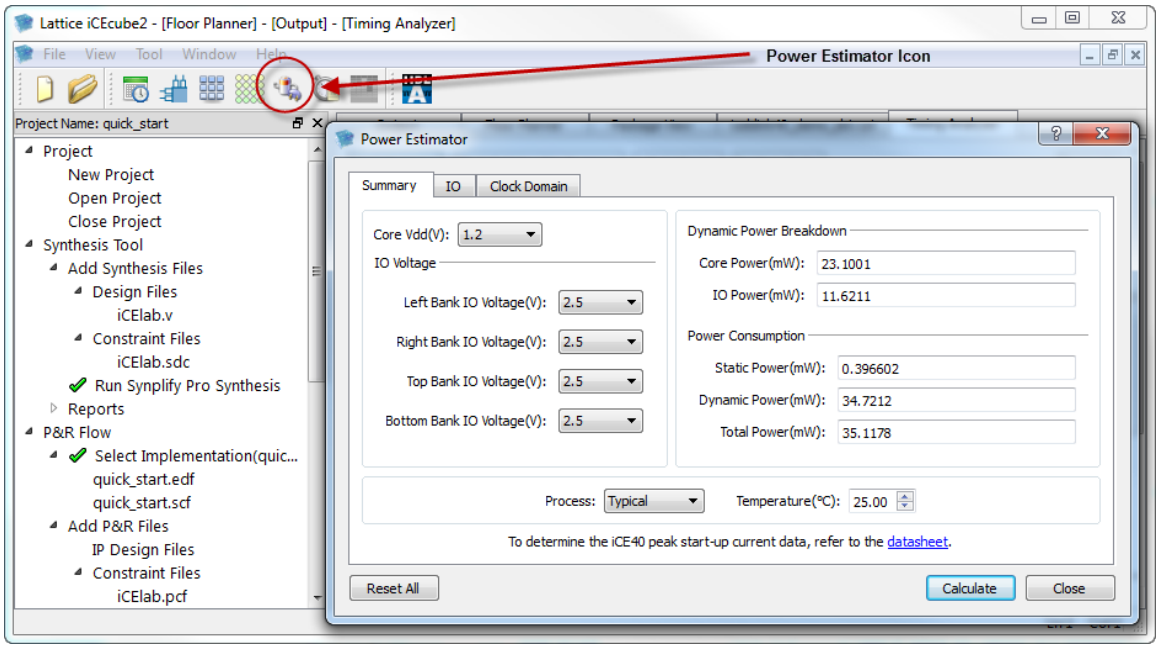

Figure 16. Power Estimator

Generate Programming Files with iCEcube2

In order to program a device, you will need to generate a programming file. In the project navigator, double click on **Bitmap**. This generates .bit and .hex files in the outputs directory.

In our example, the output directory is located here:

*<icecube2_install_directory>*examples\blinky\quick_start\quick_start_Implmnt\sbt\outputs\bitmap.

Save the iCEcube2 Project

Save the iCECube2 project and close iCEcube2 by choosing **File > Close Project**. Your project will be automatically saved in your project directory with a .project file name. In our example, the project file name is **quick_start_sbt.project**, saved in the C:\ice40_tutor\quick_start directory.

Program an iCE40 Device with Stand-Alone Diamond Programmer

The .hex files that you generated with iCECube2 software will be used for programming the external SPI Flash on Lattice iCEblink40-HX1K evaluation board. The iCE40 configures itself from the SPI flash.

For this tutorial, we will use the stand-alone version of Lattice Diamond Programmer. This is the preferred tool to program iCE40 devices **Do not use the Programmer tool that comes installed with iCEcube tool suite for this tutorial.**

Set Up the iCEblink40-HX1K Board

You must connect the iCEblink40-HX1K Evaluation Board to the computer using both the Lattice HW-USBN-2A USB programming cable and a USB Mini Cable..
Figure 17 is a schematic of the iCEBlink40 Evaluation Board.

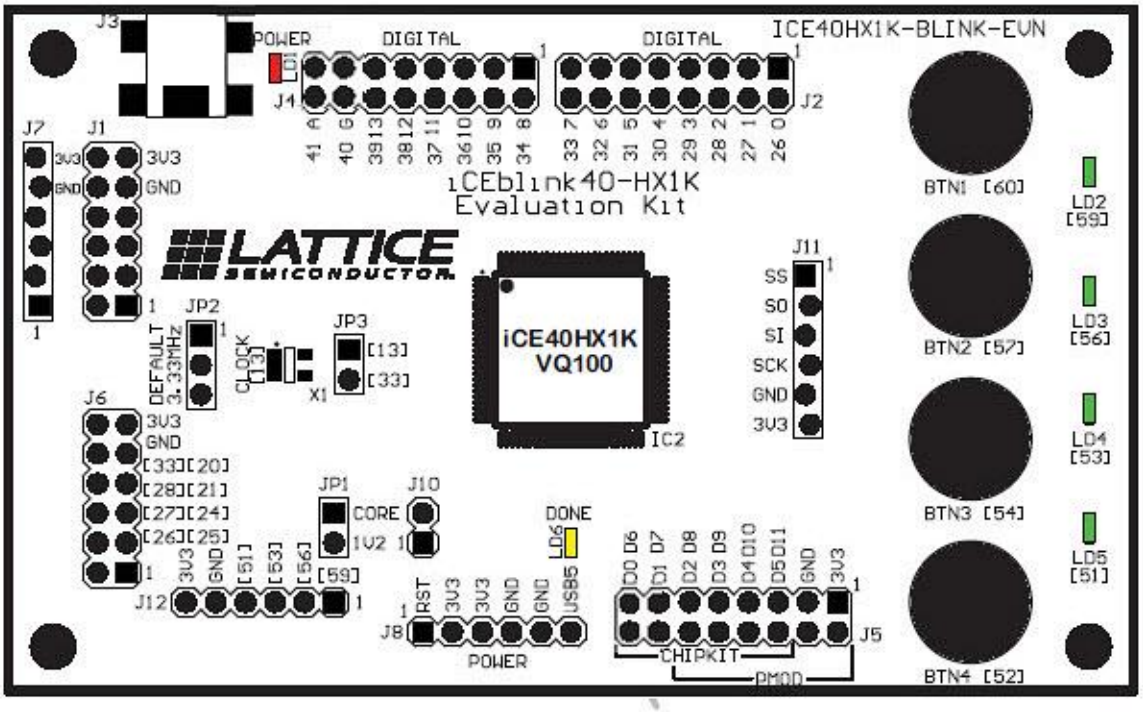

Figure 17: iCEblink40-HX1K Evaluation Board

1. Connect the fly wires of the Lattice HW-USBN-2A USB programming cable to the board as shown in Table 1.

Table 1. iCEblink40-HX1K Board to Programming Cable Connections

iCEBlink Board Connector	Lattice HW-USBN-2A USB Programming Cable
J11: SO	TDI
J11: SI	TDO
J11: SCK	TCK
J11: GND (Pin 5)	GND
J11: Vcc 3V3 (Pin 6)	VCC
J10: M25P10-SS_B (Pin 1)	ISPEN
J8 CRESET (Pin 1)	TRST

2. Connect the other end of the Lattice HW-USBN-2A USB programming cable to an available USB port on your computer.

3. Connect the USB Mini Cable the iCEblink40-HX1K board, and connect the other end to an available USB port on your computer. This cable provides 5 V power to the board.

The setup should appear as shown in the photo in Figure 18.

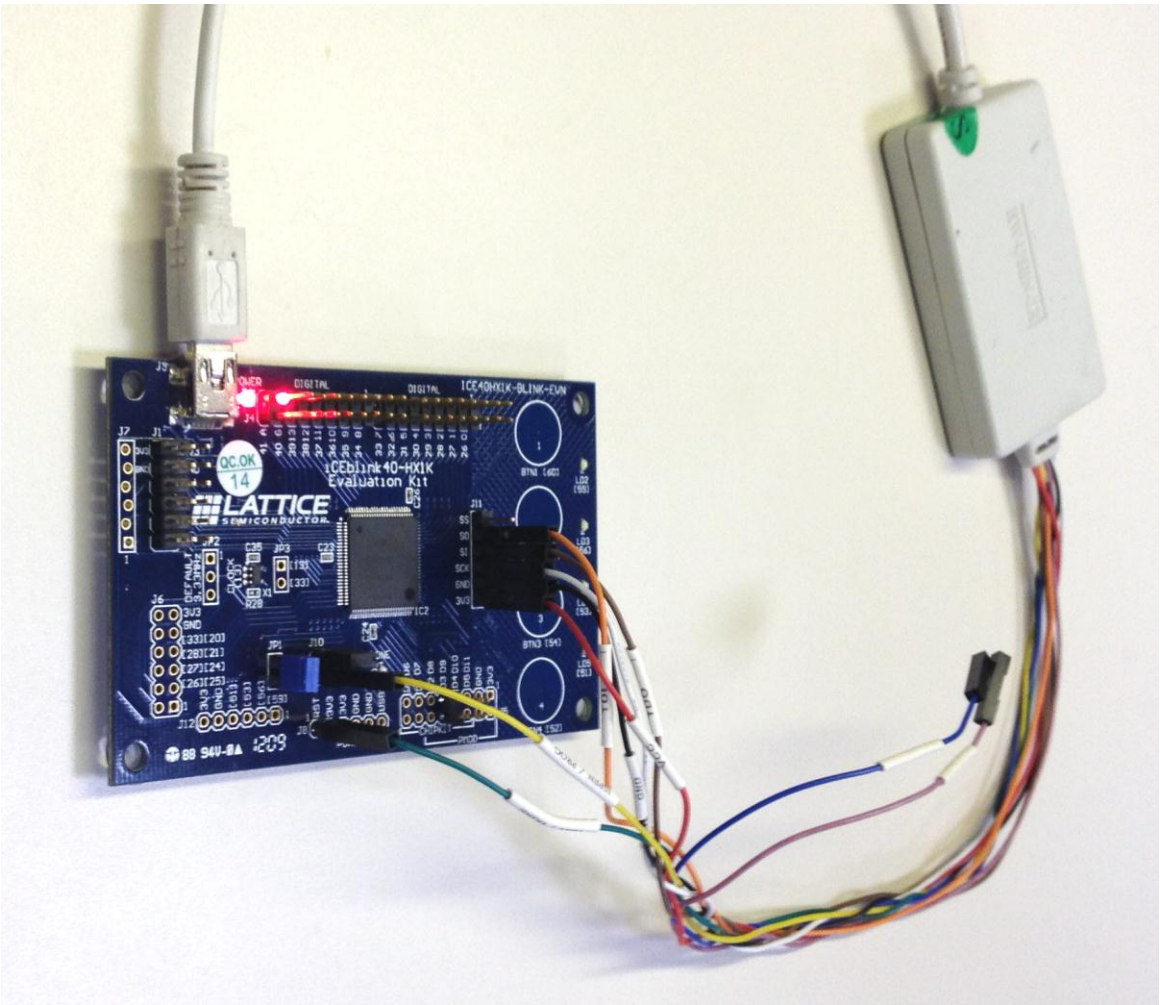

Figure 18: Cables Connected to Board

Program the Device With Stand-Alone Diamond Programmer

Use stand-alone Diamond Programmer to program the iCE40 device on the iCEbllink40-HX1K Evaluation Board.

Start the stand-alone Diamond Programmer. In Windows, from the Start menu, **choose Lattice Diamond Programmer <*version_number*> > Diamond Programmer**.

The Diamond Programmer Getting Started dialog box appears, as shown in Figure 19.

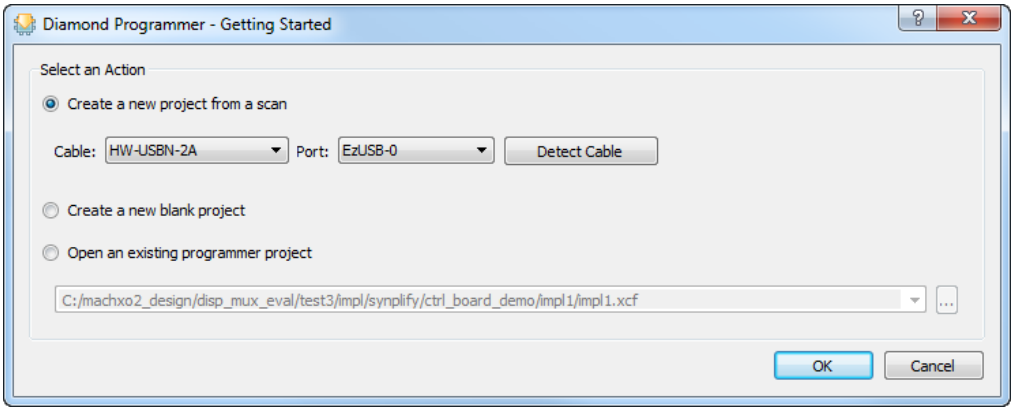

Figure 19: Diamond Programmer Getting Started Dialog Box

Choose **Create a New Project from a Scan** button and click **OK**.

The Diamond Programmer main window appears.

In the Cable Settings box in the upper right, click **Detect Cable**.

Diamond Programmer will indicate in the bottom output tab that the Lattice HW-USBN-2A USB programming cable was detected, as shown in Figure 20.

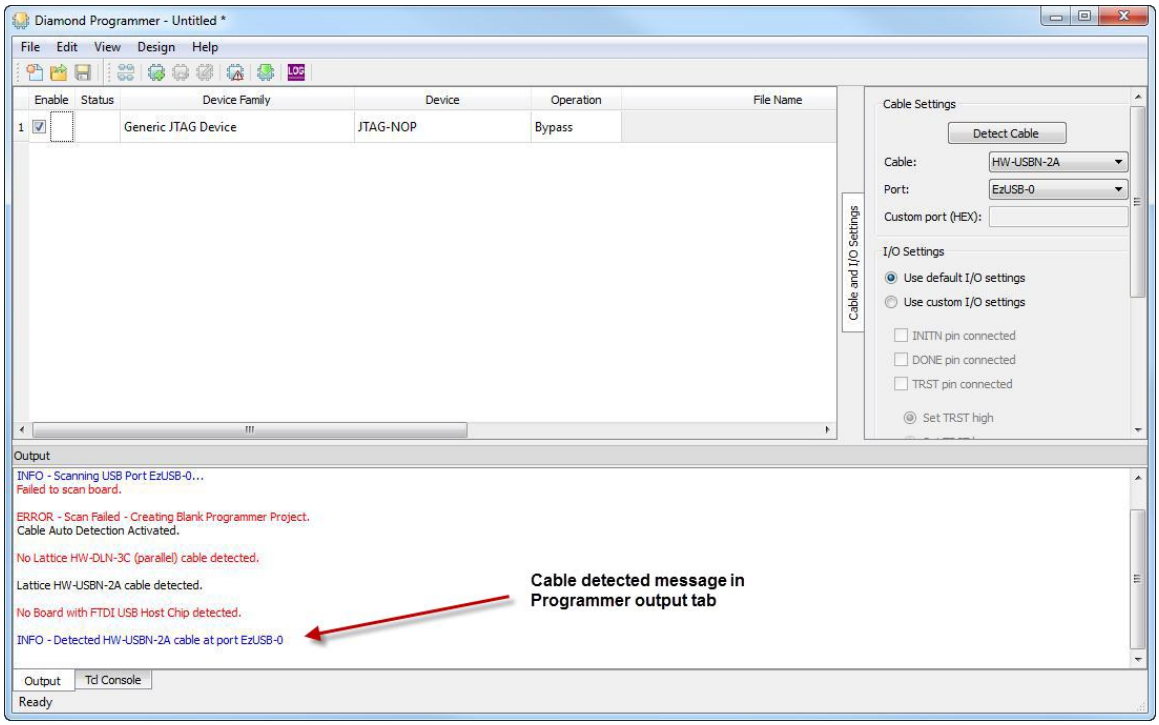

Figure 20: Diamond Programmer Main Window

In the Device Family field, click the Generic JTAG Device box and choose **iCE40** from the drop-down menu, as shown in Figure 21.

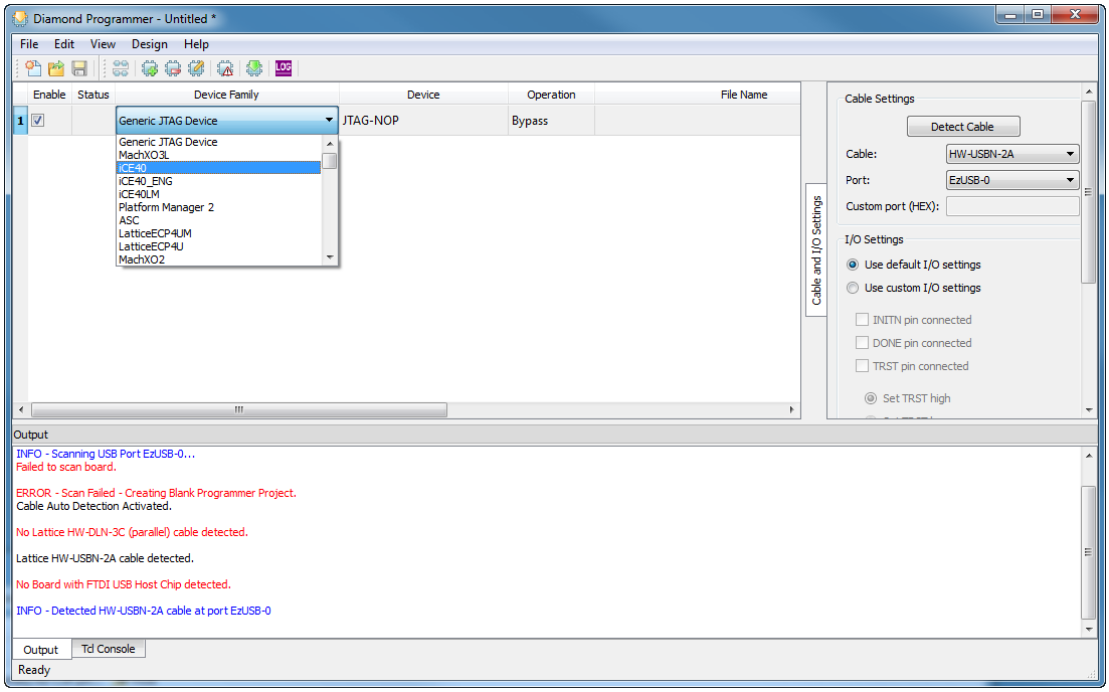

Figure 21: Choosing iCE40 Device Family

In the Device column, choose **iCE40HX1K** from the drop-down menu, as shown in Figure 22.

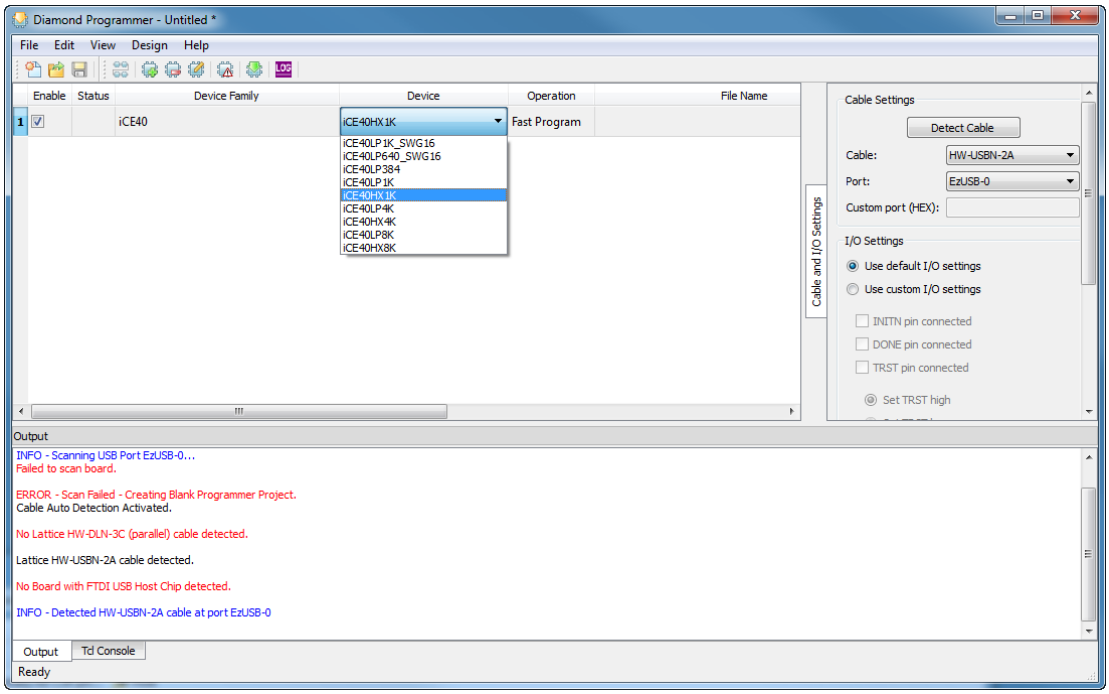

Figure 22: Choosing iCE40HX1K Device

Choose **Edit > Device Properties**, or double-click the Operation box to display the Device Properties dialog box, as shown in Figure 23.

In the Device Properties dialog box, set options as follows:

- **Access Mode**: SPI Flash Programming
- **Operation**: SPI Flash Erease,Program,Verify

In the Programming File box, browse to the **iceblink40_demo_bitmap.hex** file you generated with iCEcube2. In our example, the .hex file is located in the: C:\ice40_tutor\quick_start\quick_start_Implmnt\sbt\outputs\bitmap directory.

In the SPI Flash Options box, choose the following options:

- **Family**: SPI Serial Flash
- **Vendor**: STMicro
- **Device**: SPI-M25P 10-A
- **Package**: 8-pin SOIC

The Device Properties dialog box should be configured as shown in Figure 23.

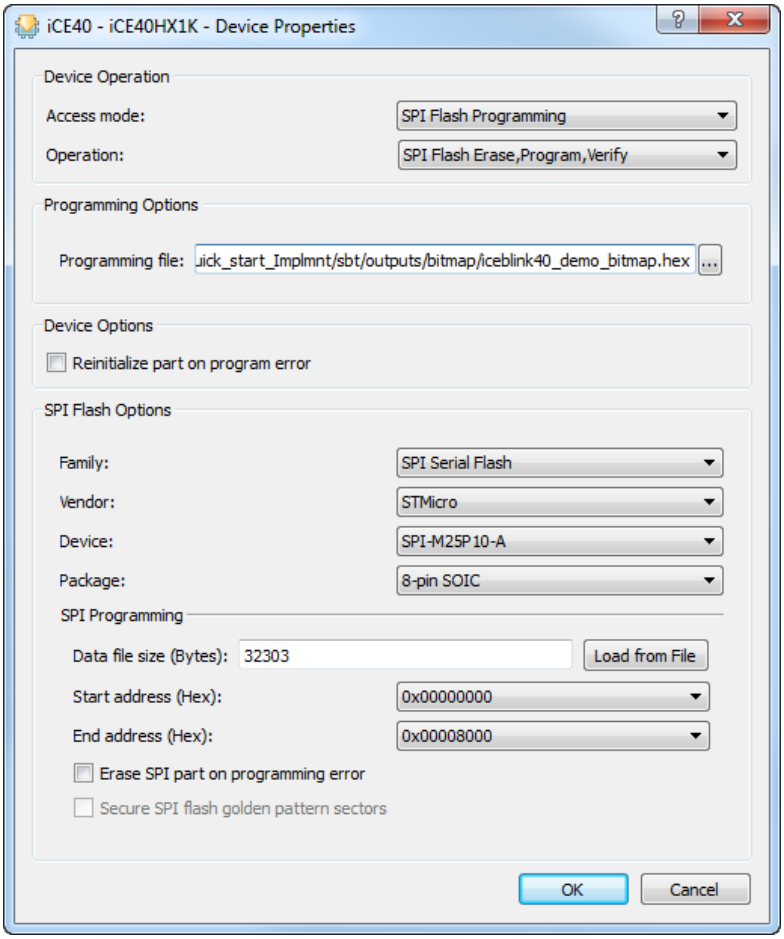

Figure 23: Device Properties Dialog Box

In the Device Properties dialog box, click **OK**.

In the Diamond Programmer main window, choose **Design > Program**, or click the Program icon in the toolbar, as shown in Figure 24. The output tab in the lower left portion of Diamond Programmer should indicate **Operation successful**.

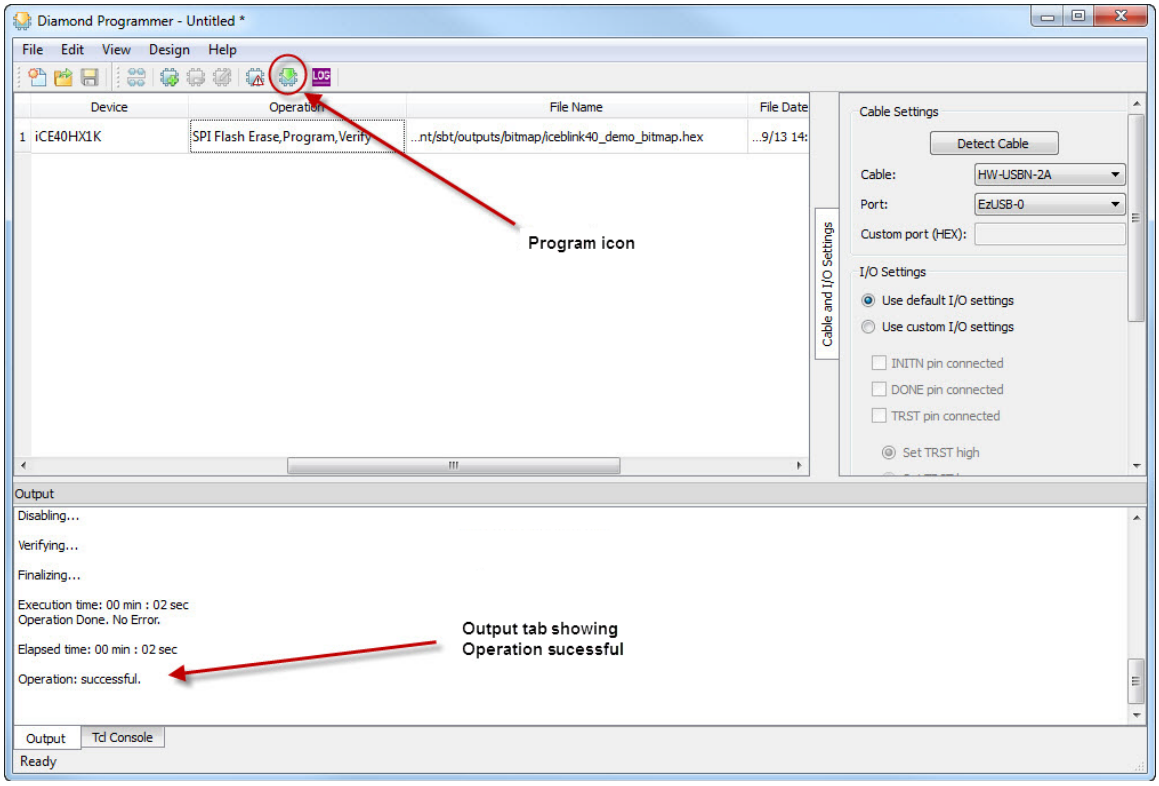

Figure 24: Diamond Programmer Main Window – Successful Program Operation

The external SPI Flash on the Lattice iCEblink40-HX1K evaluation board has been programmed, and the iCE40 is configured from the SPI flash.

The four green LEDs on the iCEblink40-HX1K board should flash in sequence from top to bottom. Congratulations, you have successfully programmed your design onto an iCE40 device.

Appendix A: Synthesizing the Design with Lattice Synthesis Engine

You can use Lattice Synthesis Engine (LSE) instead of Synopsys Synplicity Pro synthesis software to synthesize your design.

This appendix demonstrates using LSE to synthesize your design.

In iCECube2, open the project you saved in "Save the iCEcube2 Project" on page 19. Choose **File > Open Project**. Browse to the project file. In our example the project file name is **quick_start_sbt.project**, located in the C:\ice40_tutor\quick_start directory. In the Open File dialog box, highlight the project file and click **Open**.

In the iCEcube2 Project Navigator, right-click **Synthesis Tool**, and then click the **Select Synthesis Tools** pop up. In the **Select Synthesis Tool** dialog box, choose **Lattice LSE** as shown in Figure 25, and then click **OK.**

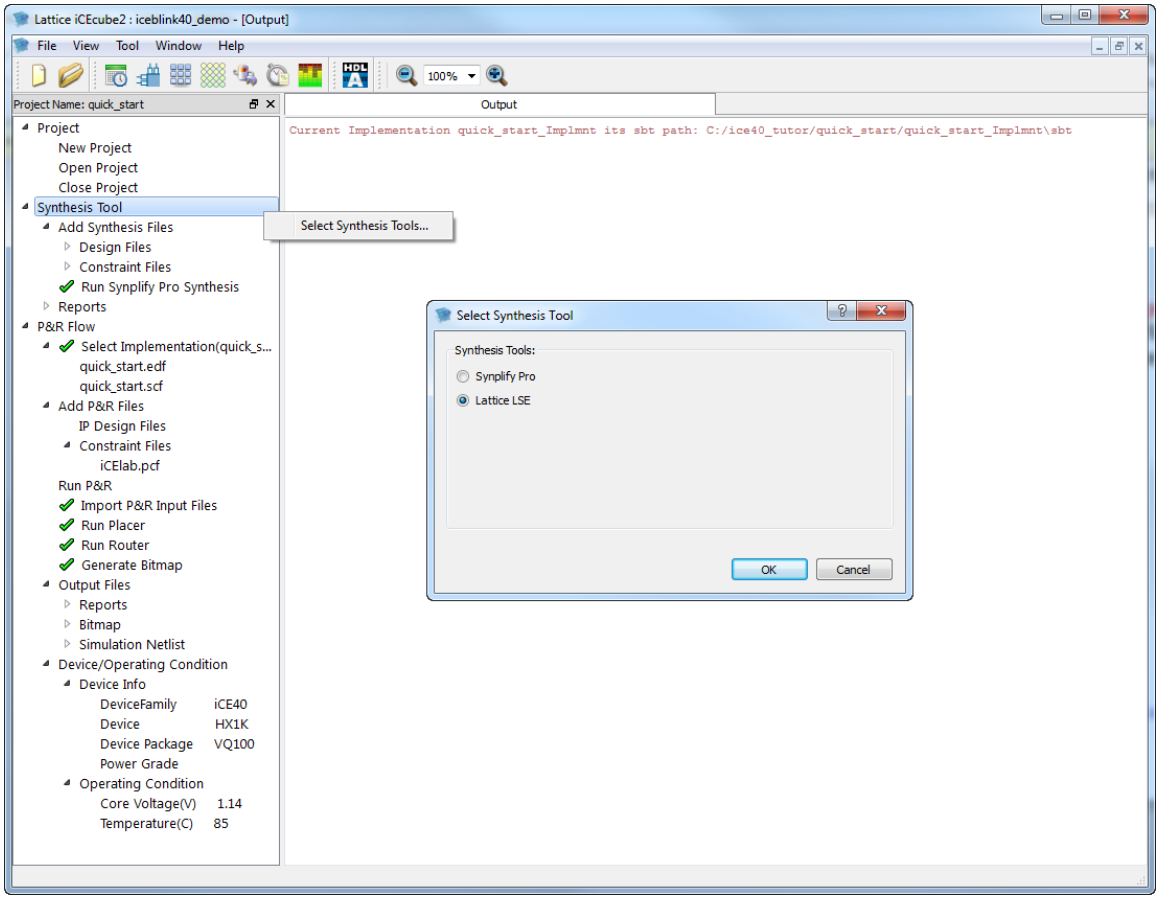

Figure 25: Select Synthesis Tool Dialog Box

Double-click **Run Lattice LSE Synthesis**, as shown in Figure 26.

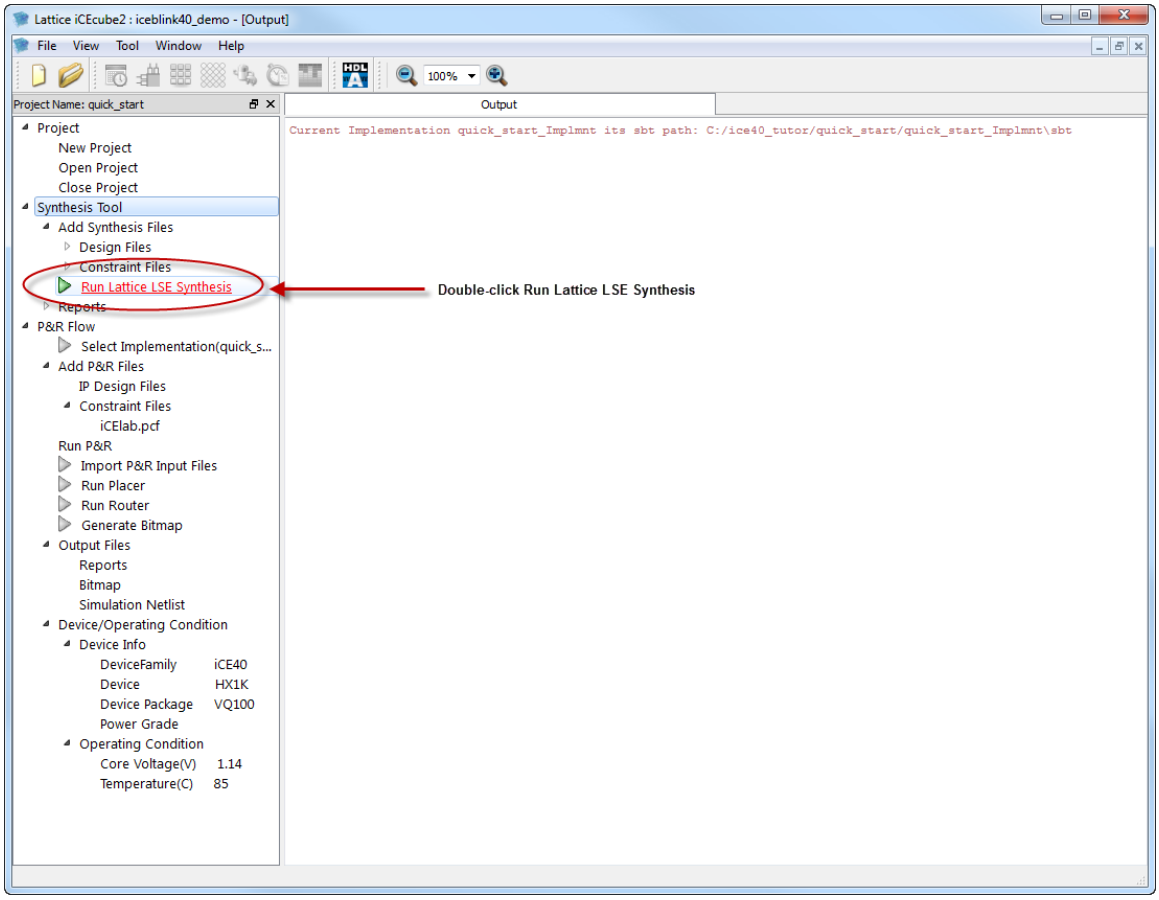

Figure 26: Run Lattice LSE Synthesis

Your design has now been synthesized with LSE. You can perform the same tasks as were performed with the design that was synthesized using Synopsis Synthesis Pro, including:

- Place the Design (page 15)
- View Floorplanner (page 16)
- View the Package Viewer (page 17)
- Route the Design (page 17)
- Perform Static Timing Analysis (page 18)
- Perform Power Analysis (page 19)
- Generate Programming Files with iCEcube2 (page 19)
- Save the iCEcube2 Project (page 19)
- Program an iCE40 Device with Stand-Alone Diamond Programmer (page 20)