Copyright Information

Copyright©2007-2012 Lattice Semiconductor Corporation. All rights reserved. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. Synopsys and Synplify Pro are trademarks of Synopsys, Inc. Aldec and Active-HDL are trademarks of Aldec, Inc. All other trademarks are the property of their prospective owners. All specifications are subject to change without notice.

Notice of Disclaimer: This software is provided to you "as-is," without any express or implied warranty.

Contact Information

Lattice Semiconductor Corporation
5555 N.E. Moore Court
Hillsboro, Oregon 97124-6421
United States of America
Tel: +1 503 268 8000
Fax: +1 503 268 8347
www.latticesemi.com

Revision History

The following table lists the revision history of this document.

<table>
<thead>
<tr>
<th>Version</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Release iCEcube2 2010.03</td>
</tr>
</tbody>
</table>
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 FPGA device.

For information on the Synopsys Synplify Pro software, please refer to the Synplify Pro documentation provided in the synpro/doc directory in the iCEcube2 software installation (<icecube2_install_dir>/synpro/doc), and on the Lattice website.

For information on the Aldec Active-HDL design tool, please refer to the Active-HDL documentations available at <icecube2_install_dir>/Aldec/Active-HDL/BOOKS or Active-HDL product help which can be invoked from the help menu in the software tool.

For detailed information of the iCEcube2 development tools, please refer to the *iCEcube2 User Guide*.

Software Version
This tutorial is based on iCEcube2 Software Version 2012.06

For more information about acquiring the iCEcube2 software, please visit the Lattice Semiconductor website: [http://www.latticesemi.com](http://www.latticesemi.com).

Platform Requirements
The iCEcube2 software can be installed on a platform satisfying the following minimum requirements.
A 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 iCEcube2 Install overview document, please visit the software documentation webpage.

Programming Hardware
Here are the following ways to program iCE FPGA devices:

- A third party programmer, using the programming files generated by the iCEcube2 Physical Implementation Tools. Consult the third party programmer user manual for instructions.
- The iCEblink and iCEman evaluation Boards, which not only serves as a vehicle to evaluate iCE FPGAs, but also includes an integrated device programmer. This programmer can be used to program devices on the evaluation boards, or it can be used
to program devices in a target system. Please visit Lattice Semiconductor website: http://www.latticesemi.com for additional information on the Evaluation Boards.

- Digilent USB cables
- The iCE Programming hardware : iCEcable, iCEprog (Programmer base module) and iCEsab(socket adaptor). Refer to lattice website: 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. Please refer to Platform Requirements for additional information on supported operating systems.

The Figure 1 below depicts the design flow using the iCEcube2 Tool Suite. The components in blue signify functionality supported by Lattice Semiconductor’s proprietary iCEcube2 software, and the components in purple indicate the functionality supported by Synopsys’ Synplify Pro synthesis tools and the Aldec Active-HDL simulation tool. The iCEcube2 software, Synopsys Synplify Pro 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](image-url)
Design Flow
The following steps provide an overview of the design flow using the iCEcube2 Tool Suite.

1. Create a new project in the iCEcube2 Project Navigator and specify a target device and its operating conditions. Add your HDL (Verilog or VHDL) design files and your Constraint files to the project.
2. Synthesize your design using the Synplify Pro design software. This software has been provided as part of the iCEcube2 Tool Suite, and can be invoked from the iCEcube2 Project Navigator. Within the Synopsys design environment, assign your Logic Synthesis, Timing and Pin constraints.
3. Perform Placement and Routing using the iCEcube2 place and route tools. iCEcube2 also supports physical implementation tools such as floor planning, allowing users to manually place logic cells and IOs.
4. 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.
5. Perform Static Timing Analysis using the iCEcube2 static timing analyzer.
6. Generate the device programming and configuration files from the iCEcube2 Physical Implementation tools.
7. Program your device using the device programming hardware provided by Lattice.

Tutorial
This chapter provides a brief introduction to the iCEcube2 design flow. The goal of this chapter is to familiarize the user with the fundamental steps needed to create a design project, synthesize and implement the design, generate the necessary device configuration files, and program the target device.

Creating a Project
Starting the iCEcube2 software for the first time, you will see the following interface shown in Figure 2.
The first step is to create a new design project and add the appropriate design files to your project. You can create a new project by either selecting File→New Project from the iCEcube2 menu, or by clicking the Create a New Project icon as seen in Figure 2. The New Project Wizard GUI is displayed in Figure 3.
This example is targeted for iCE40 family device. Follow the following steps to setup the project properties.

1. **Project Name** Field: Specify a project name (*quick_start*) in the Project Name field.
2. **Project Directory** Field: Specify any directory where you want to place the project directory in the Project Directory field.
3. **Device Family** Fields: This section allows you to specify the Lattice iCE device family you are targeting. For this example, change the Device Family to *iCE40*.
4. **Device** Fields: This section allows you to specify the Lattice device and package you are targeting. For this example, change the Device to *HX1K* and change the device package to the VQ100.
5. **Operating Condition** Fields: This section allows you to specify the operating conditions of the device which will be used for timing and power analysis. The **IO Bank Voltage** option shown in Figure 3 is not available for iCE65 family devices.

6. **Start From Synthesis**: This option allows you to start the flow from Synthesis using Synopsys Synplify Pro tool. For current example, select this option.

7. **Start From BackEnd**: This option allows the user to start from Post Synthesis flow.

8. Click Next to go to the Add Files dialog box shown in Figure 4. You will be prompted to create a new project directory. Click Yes.

9. In the Add Files dialog box, navigate to: `<iCEcube2 installation directory>/examples/blinky`

Highlight the following files:

- `blinky.vhd`
- `blinky_syn.sdc`

Select each file and click `>>` to add the selected file, or click `>>>` to add all the files in the open directory (files can be removed using `<<` and `<<<`) to your project. Click **Finish** to create the project.

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

![Figure 4: New Project Wizard – Add Files dialog box](image)

After successfully setting-up your project, you will return to the following iCEcube2 Project Navigator screen shown in Figure 5.
Synthesizing the design

After a successful project setup, **Double-Click on the Synthesis Launch Synthesis Tool** icon in the project navigator window. See Figure 6. This will bring-up the Synopsys Synplify Pro synthesis tool’s graphical user interface. See Figure 7.
Hit the Run Button to synthesis your design. Once synthesis is complete, you will see a Done message. See Figure 8.
**View Timing Constraints**

**Double Click** on the `blinky_syn.sdc` file under the Constraint folder. See Figure 9. It will open the timing constraints for the project shown in Figure 10.
Figure 9: Open the SDC file to View Timing Constraints

Figure 10: View Timing Constraints
Viewing Hierarchical View of Synthesis Results

Under the **HDL-Analyst menu**, **Select RTL > Hierarchical View**. You will see a hierarchical RTL view of the design just synthesized. See Figure 11.

If you double click on one of the blocks, it will take you to the RTL for that block. See Figure 12.
P&R Flow
This section goes through the post synthesis physical implementation flow.

Select Implementation

In order to ensure that the synthesized design can be successfully imported into iCEcube2, exit the Synplify Pro GUI.

Return to the iCEcube2 Navigator and Double-click on Select Implementation. See Figure 13. This will tell iCEcube2 which synthesis implementation to process for place and route. 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, select OK when the Select Synthesis Implementation dialog box appears.

Figure 12: Double-clicking on a block will reveal its HDL code in HDL Analyst
Importing Physical Constraints

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

In the iCEcube2 Project Navigator, **Right Click** on **Specify Additional Files**. Select **Add Files**… See Figure 14.

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 software user guide.
Navigate to the <ICEcube2 Installation Directory>/examples/blinky and Add blinky.pcf file.  See Figure 15.

Import Place & 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.  **See Figure 16.**  Once completed you will see a green check next to **Import P&R Input Files.**  **See Figure 17.**
Figure 16: Import P&R Input files

Figure 17: Successful Import of P&R Input Files
Place the Design

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. See Figure 18.

View Floorplanner

At this point, since placement has been completed, you can view the placement of the design by opening the Floorplanner. You can open the Floorplanner by going to the menu and selecting Tool > Floorplanner or you can also select the Floorplanner Icon. See Figure 19.
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 Viewer or you can also select the Package Viewer Icon. See Figure 20.
Route the Design

Double-click on Route 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 you have routed the design, you can perform timing analysis to check to see if the design meets your timing requirements. To launch the timing analyzer, go to the menu and select Tool > Timing Analysis. You can also select the Timing Analysis Icon. See Figure 21.
You can see from the timing analysis that our 32 KHz design is running at over 395.62 MHz and our 32 MHz clock is running at over 222.36 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 iCEcube2 software usage guide for details on timing analysis.

Perform Power Analysis

iCEcube2 also comes with power estimator tool. To launch the power estimator, go to the menu and select Tool >> Power Estimator. You can alternatively select the power estimator icon. Figure 22. There are multiple tabs in the Power Estimator tool including Summary, IO, and Clock Domain. On the Summary tab, change the Core Vdd to 1.2V and make sure all IO voltages are at 2.5V. Then hit calculate. The estimator will update with power information for both static and dynamic power. For more information on using the IO and Clock Domain tabs, please refer to the detailed section on the Power Estimator tool in the iCEcube2 software userguide.
Programming the Device

In order to program a device, you will need to generate a programming file. In the project navigator, **double click on Bitmap**. Expand the Bitmap section in the Output Files. The section is now populated with .hex and .bin files. The .hex files are used for programming the external SPI Flash on Lattice iCE evaluation kit. The iCE40 configures itself from the SPI Flash.

You are now ready to program an iCE40 FPGA device with the generated bitmap. Invoke the programmer from the **Programming** icon which is now enabled in the Project Navigator. Alternately, you may invoke it from the **Tool>Programmer** menu item.

To program the Lattice iCEblink40 board, select iCEblink40 from the Programming Options dropdown menu and select M25P10A for the External Serial Flash PROM. See Figure 23. For more details on programming the iCEblink40 evaluation kit, refer to Evaluation Kit’s User’s Guide. Click on the image file settings button to ensure you latest .hex file is selected, as shown in Figure 24.

Additional details on programming a device are provided in a separate section **Programming the Device** in section *iCEcube2 Physical Implementation Tools* of the iCEcube2 software usage guide.
Figure 23: Programmer Graphical User Interface

Figure 24: Image File Settings