



LatticeMico32 Development Kit User's Guide

Lattice Semiconductor Corporation
5555 NE Moore Court
Hillsboro, OR 97124
(503) 268-8000

December 2006

Copyright

Copyright © 2006 Lattice Semiconductor Corporation.

This document may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Lattice Semiconductor Corporation.

Trademarks

Lattice Semiconductor Corporation, L Lattice Semiconductor Corporation (logo), L (stylized), L (design), Lattice (design), LSC, E²CMOS, Extreme Performance, FlexiMAC, FlexiPCS, GAL, GDX, Generic Array Logic, HDL Explorer, IPexpress, ISP, ispATE, ispCLOCK, ispDOWNLOAD, ispGAL, ispGDS, ispGDX, ispGDXV, ispGDX2, ispGENERATOR, ispJTAG, ispLEVER, ispLEVERCORE, ispLSI, ispMACH, ispPAC, ispTRACY, ispTURBO, ispVIRTUAL MACHINE, ispVM, ispXP, ispXPGA, ispXPLD, LatticeEC, LatticeECP2, LatticeECP2M, LatticeECP, LatticeECP-DSP, LatticeMico8, LatticeMico32, LatticeSC, LatticeXP, MACH, MachXO, MACO, ORCA, PAC, PAC-Designer, PAL, Performance Analyst, PureSPEED, Silicon Forest, Speedlocked, Speed Locking, SuperBIG, SuperCOOL, SuperFAST, SuperWIDE, sysCLOCK, sysCONFIG, sysDSP, sysHSI, sysI/O, sysMEM, The Simple Machine for Complex Design, TransFR, UltraMOS, and specific product designations are either registered trademarks or trademarks of Lattice Semiconductor Corporation or its subsidiaries in the United States and/or other countries. ISP and Bringing the Best Together are service marks of Lattice Semiconductor Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Disclaimers

NO WARRANTIES: THE INFORMATION PROVIDED IN THIS DOCUMENT IS "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF ACCURACY, COMPLETENESS, MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL LATTICE SEMICONDUCTOR CORPORATION (LSC) OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION PROVIDED ON THIS SITE, EVEN IF LSC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF CERTAIN LIABILITY, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

LSC may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. LSC makes no commitment to update this documentation. LSC reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. LSC recommends its customers obtain the

latest version of the relevant information to establish, before ordering, that the information being relied upon is current.

Type Conventions Used in This Document

Convention	Meaning or Use
Bold	Items in the user interface that you select or click. Text that you type into the user interface.
<i><Italic></i>	Variables in commands, code syntax, and path names.
Ctrl+L	Press the two keys at the same time.
<i>Courier</i>	Code examples. Messages, reports, and prompts from the software.
...	Omitted material in a line of code.
.	Omitted lines in code and report examples.
[]	Optional items in syntax descriptions. In bus specifications, the brackets are required.
()	Grouped items in syntax descriptions.
{ }	Repeatable items in syntax descriptions.
	A choice between items in syntax descriptions.

Contents

Chapter 1	Introduction	1
Chapter 2	Getting Started with the Development Kit	3
	Before You Start	3
	Development Kit Contents	3
	Development Tools	4
	ispLEVER Design Environment	4
	LatticeMico32 System Software	4
	Installing the LatticeMico32 Development Tools	5
	System Requirements	6
Chapter 3	Using the LatticeMico32/DSP Development Board	7
	Setting Up the Development Board	8
	Troubleshooting USB Driver Installation	10
	Verifying Correct Board Operation	11
	Checking the Power Supply	11
	Ensuring Board/PC Communications	11
	Restoring the Factory Configuration	14
Chapter 4	Running LatticeMico32 System Software	15
	Creating an ispLEVER Project	15
	Creating a New Platform	16
	Generating an .msb File	17
	Adding Components to Your Platform	17
	Generating Component Addresses	18
	Assigning Component Interrupt Priorities	19
	Performing Design Rule Checks	19
	Saving Your Platform	19
	Generating Your Platform	19
	Generating the Microprocessor Bitstream	20

Importing the Verilog File	20
Connecting the Microprocessor to FPGA Pins	20
Generating the Bitstream	21
Downloading the Hardware Bitstream to the FPGA	21
Creating the Software Application Code	22
Debugging the Software Application Code	23
Downloading Executable Code to Flash Memory	23
Recommended References	24
Index	25

Introduction

This guide describes how to start using the LatticeMico32 Development Kit, which includes a low-cost platform for evaluating the LatticeMico32 FPGA embedded microprocessor for your specific application. Along with the LatticeMico32/DSP development board and accessories, this kit includes all the hardware development tools, software development tools, and documentation that you need to begin developing designs using a LatticeMico32 embedded microprocessor.

In addition, this document reviews the contents of the LatticeMico32 Development Kit and walks you through the process of setting up the LatticeMico32 System development environment.

It also refers you to more detailed information on specific board features that are outside of the scope of this guide. Because this document only provides an overview of major software tasks you perform in this flow, it also refers you to other LatticeMico32 System software documentation as needed.

This document shows you how to do the following:

- ◆ Install all applicable development tools.
- ◆ Set up the LatticeMico32/DSP development board and ensure that it is operating correctly.
- ◆ Connect the development board to the PC host to establish communication.
- ◆ Download the SoC platform design to the FPGA on the board.
- ◆ Become familiar with the software development tools and major design flow steps.
- ◆ Download the executable code to flash memory.
- ◆ Learn about other existing documentation to use in conjunction with this guide.

Getting Started with the Development Kit

This chapter provides an inventory of all the items that come with your kit and all the necessary software that is included. It also provides installation information for the software development tools for the LatticeMico32 System software tool chain. Make sure that you familiarize yourself with the contents of this chapter before you set up the board and run the software tools.

Before You Start

Review this section to understand what is included in this kit.

Development Kit Contents

The following items are included in your LatticeMico32 Development Kit:

- ◆ LatticeMico32/DSP development board pre-configured with the Lattice Semiconductor reference design
- ◆ Standard USB cable
- ◆ AC adaptor cord
- ◆ Latest compatible version of Lattice Semiconductor's ispLEVER design software CD-ROMs
- ◆ LatticeMico32 System CD-ROM
 - ◆ LatticeMico32 System software tools
 - ◆ *LatticeMico32 Development Kit User's Guide* (this document)
 - ◆ *LatticeMico32/DSP Development Board User's Guide*

If you are missing any of the items just listed, contact Lattice Semiconductor Technical Support. Documentation is also available on the Lattice Semiconductor Web site.

In addition to its inclusion with this development kit, the LatticeMico32/DSP development board is also available as a stand-alone product that includes just the board, the USB cable, and the AC adaptor. Please contact Lattice Semiconductor Technical Support or go to the Lattice Semiconductor Web site at www.latticesemi.com for more details.

Development Tools

The kit includes the Lattice Semiconductor ispLEVER software development environment, which contains the following development tools:

- ◆ LatticeMico32 System software
- ◆ ispLEVER design software
- ◆ ispVM System software

See <https://www.latticesemi.com/products/intellectualproperty/ipcores/latticemico32> for the latest release of the LatticeMico32 System software available for download.

ispLEVER Design Environment

To use this kit, you must have a licensed version of the ispLEVER 6.1 version or later of the FPGA design software. The Lattice Semiconductor ispLEVER software set consists of three CD-ROMs that include all of the design tools, available devices, and the ispVM System software for device configuration. For instructions on ispLEVER software licensing and installation, refer to the *ispLEVER 6.1 Installation Notice*.

The ispLEVER software is a comprehensive environment for designing with the integrated LatticeMico32 System development tool chain. Using the ispLEVER software, you can develop hardware design files, synthesize the netlist, define the I/O pins, apply timing constraints, perform timing analysis, and output a bitstream file to configure your target FPGA. You can access the ispVM System software through ispLEVER's Project Navigator toolbar or download it at www.latticesemi.com/ispvm.

See "Installing the LatticeMico32 Development Tools" on page 5 for more information on downloading a design for evaluation purposes.

LatticeMico32 System Software

You must install the LatticeMico32 System software to access the embedded processor design flow. This software is installed separately on top of the ispLEVER design software. The LatticeMico32 System software set consists of the following items:

- ◆ CD-ROM containing the LatticeMico32 System software suite
- ◆ *LatticeMico32 Development Kit User's Guide* (this guide)
- ◆ *LatticeMico32/DSP Development Board User's Guide*

You can install the LatticeMico32 System software from the CD-ROM included in the development kit, or you can download it from the Lattice Semiconductor Web site at <https://www.latticesemi.com/products/>

[intellectualproperty/ipcores/latticemico32](#). This CD-ROM includes the following tools, along with other components:

- ◆ Mico System Builder (MSB), which provides an integrated development environment for creating your microprocessor platform. MSB defines the the CPU component, its peripherals, and their connections.
- ◆ C/C++ Software Project Environment (C/C++ SPE), which enables you to create the code to configure the functionality of the microprocessor.
- ◆ The Debugger, which enables you to analyze the code to help you identify problems and correct them before and after deployment.
- ◆ LatticeMico32 microprocessor, which is a 32-bit microprocessor core in an IP module. There are several implementations of this module available. See “Installing the LatticeMico32 Development Tools” on page 5.

This development tool chain contains the industry-standard GNU-based tools for source code compilation, such as an assembler, C/C++ compiler, and linker.

Installing the LatticeMico32 Development Tools

The section provides brief instructions for installing the LatticeMico32 System software development tools on your PC. See the *LatticeMico32 Development Tools Installation Guide* that is available in the documentation set for more details on installation, requirements, and help system browser proxy settings.

To install the required software, perform the following steps:

1. If it is not installed on your system already, install all necessary Lattice Semiconductor ispLEVER CD-ROM disks, as directed in the *ispLEVER 6.1 Installation Notice* or later. Use the ispLEVER version that is compatible with the version of the LatticeMico32 System software that you are installing. The version numbers should match.

If the ispLEVER 6.1 or later software is already installed, you can skip this step. By default, the ispLEVER software installation also installs the ispVM System software necessary to download the bitstream for device configuration.

1. Insert the LatticeMico32 System CD-ROM into your CD-ROM drive. To begin installation, click **Next** on the first page of the LatticeMico32 Setup dialog box that automatically opens.
2. Follow the installation wizard options to accept the licensing agreement, install the tools in the default directory, and start copying necessary files that will be integrated with ispLEVER tools.

By default, the LatticeMico32 System software is installed in your ispLEVER installation directory's .\ispTOOLS folder. It is assumed that you have installed ispLEVER as instructed and that you are not attempting to install the software as a stand-alone tool, which is not recommended for this kit.

3. Click **Finish** in the final page of the installation wizard.

Note

If you need to re-install the LatticeMico32 System software, you should first uninstall the previous installation to ensure that you are using the tool with the correct version of ispLEVER. For example, LatticeMico32 System version 6.1 is intended for use with ispLEVER 6.1 SP1. To install the LatticeMico32 system software manually, execute the setup.exe on the CD.

System Requirements

This section lists the minimum system requirements that your PC must have to ensure proper functionality and successful installation. Software requirements are listed as well for purposes of convenience.

- ◆ Intel Pentium or Pentium-compatible PC running at 400 MHz or faster
- ◆ Microsoft Windows 2000 or Windows XP Professional
- ◆ 512 MB memory required; 1 GB memory recommended for FPGAs
- ◆ Approximately 4.62 gigabytes of free disk space
- ◆ Graphics display of 1024 X 768
- ◆ CD-ROM drive (2X or above) or DVD-ROM drive
- ◆ Microsoft-compatible mouse and mouse driver
- ◆ USB port for use with the LatticeMico32/DSP development board

Following are the software requirements:

- ◆ ispLEVER 6.1 software or later installed on your system (included)
- ◆ LatticeMico32 System software, version 6.1 or later. The version must match the version of the ispLEVER software (included) installed.
- ◆ ispVM System software for downloads (included)

Note

Refer to the *LatticeMico32 Development Tools Installation Guide*, which is available in the documentation set for system requirements and alternative installation options. This document provides complete installation scenarios and any possible issue that you might confront, for example, proxy settings to ensure proper function of the online help.

Using the LatticeMico32/ DSP Development Board

This chapter describes how to use the development board for the LatticeMico32 embedded microprocessor.

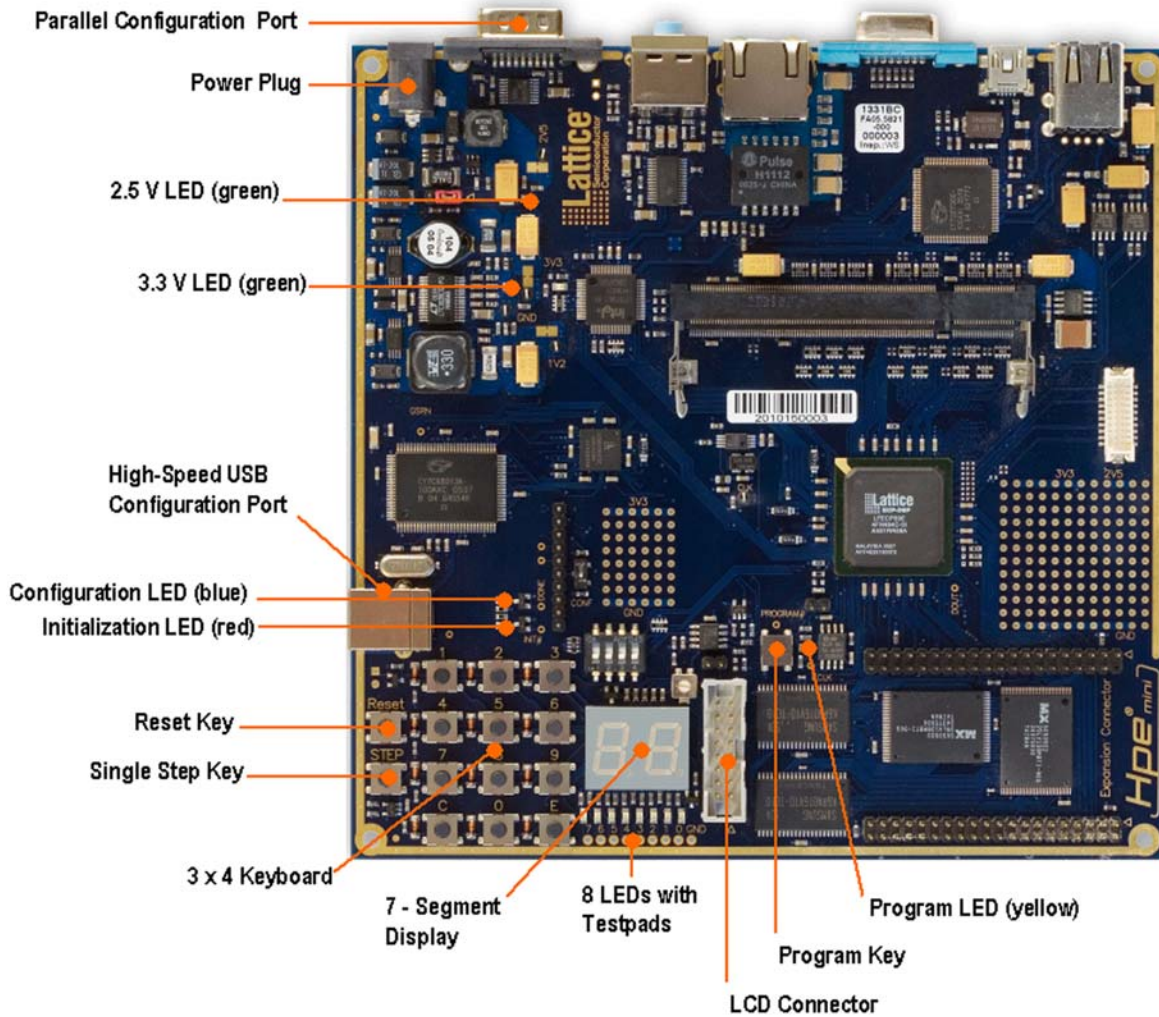
The LatticeMico32/DSP development board can be used as a basis for prototyping complex embedded systems and for programming the microprocessor architecture and functionality.

The LatticeMico32/DSP development board is equipped with a LatticeECP33 FPGA DSP-enabled device and a MachXO CPLD device. The board is pre-configured with the Lattice Semiconductor reference design.

Figure 1 shows the various LED indicators, test pads, keyboards, and other on-board interfaces that you will encounter during an evaluation of the LatticeMico32 development kit. This diagram provides a quick overview of these features.

To learn more about the board design and features, refer to the *LatticeMico32/DSP Development Board User's Guide*, which includes much more detail about the hardware components and their features.

Figure 1: Interface Layout of the LatticeMico32/DSP Development Board



Setting Up the Development Board

This section describes the steps required to set up the board properly to get started. The board comes with a bitstream pre-configured with the Lattice Semiconductor reference design.

To set up the LatticeMico32/DSP development board, do the following:

1. Remove the LatticeMico32/DSP development board and accessories from their anti-static packaging and compare the contents to the packaging list. Ensure that you avoid situations that would create electrostatic discharge (ESD) during setup or use of the board.

1. Position the board in front of you so that the keyboard is at the bottom left, as shown in Figure 1.
2. Plug the AC power supply into an outlet, then plug the AC connector into the power jack on the board.
3. If you already have any other download cables hooked up to the development board, disconnect them before proceeding, uninstall any ispDOWNLOAD USB drivers, and reboot your machine. To uninstall drivers, open ispVM and choose **ispTools > Install/Uninstall LSC Win98/2000/XP USB Driver** to open a dialog box with install and uninstall options.

Note

If you have multiple cable hook-ups, you may not be able to download or debug on one or both cables. This will be apparent later when using the Debugger tool in LatticeMico32 System software, because the program will issue error messages.

4. Connect the standard USB cable provided to an available USB port on your PC and to the USB port on the lower left side of the board, as it is oriented in Figure 1.

You should see a message on your PC that a new hardware component was detected.
5. Cancel out of any hardware component prompts in Windows, and choose **Start > Programs > Lattice Semiconductor > ispVM System** to open the ispVM System software.

You will install your USB driver using this download software. You must be the PC administrator or have administrator privileges to install this driver.
6. In ispVM, choose **ispTools > Install/Uninstall LSC Win98/2000/XP USB Driver**.

A popup dialog box appears that informs you that you must be an administrator to install the driver.
7. Click **Yes** to indicate that you have administrator privileges, and click the **Close** button to exit the pop-up dialog box.
8. In the Install dialog box, click the **Install** button, and click the **Close** button to exit the dialog box when installation is completed.

If the driver already exists, you will be prompted to overwrite the present driver file. Click **Yes** if this occurs.
9. Click **Yes** in the dialog box that prompts you to restart your PC after installing the driver.
10. If you experience any USB driver installation issues, see “Troubleshooting USB Driver Installation” on page 10 to resolve them.

11. Ensure proper operation of the development board, as described in "Verifying Correct Board Operation" on page 11.

Note

In addition to the configuration method described for the purposes of using this kit, it is also possible to configure Lattice Semiconductor devices using an ispDOWNLOAD cable. A connector X3 is available to connect the ispDOWNLOAD cable, which is used to program IEEE 1532-compliant programmable devices. See the "ispDOWNLOAD Cable Connector" section in the *LatticeMico32/DSP Development Board User's Guide* for more details.

Troubleshooting USB Driver Installation

If you have any issues with your USB driver installation, review the following items to determine a solution.

- ◆ To ensure that your USB driver has been set up properly, do the following:
 - a. In your Windows 2000 Control Panel, double-click on **System** to access the System Properties dialog box.
 - b. Click on the **Hardware** tab.
 - c. Click on the Device Manager button to open the Device Manager dialog box.
 - d. In the list box, expand the Universal Serial Bus Controllers list item until you see **Lattice USB ISP Programmer** beneath it.
 - e. Right-click on the item and select **Properties**.
 - f. In the Driver tab, you should see a driver provider named, "%Tril%" in the dialog box.

Accessing the system control panel may be different for other supported Windows operation systems, for example, Windows XP.

- ◆ You must have the correct files provided in the ispVM System software to ensure proper driver set-up and installation. The correct driver is located in the following file path:

```
<install_dir>\ispvmsystem\trilw2k.inf
```

Another required file called the ezusb.sys file is automatically copied to your C:\WINNT\system32 folder from the ispvmsystem folder during the ispVM System installation. This path may vary, depending upon your operating system, .

- ◆ If you attempted to perform the setup through the Windows hardware component wizard that appears when your PC detects a new device, the ezusb.sys file was not likely copied into your system32 folder. You can manually copy it to that location; however, it is recommended that you carefully follow the instructions in the previous section.
- ◆ If you are running WindowsXP on a system that has the USB 1.1 host, you will encounter a warning message in the ispVM System download

software that informs you that you are installing a high-speed device onto a full-speed hub. You can safely cancel out of these messages and proceed. IspVM System ideally expects the system to be equipped with a USB 2.0 host, which allows for faster download speeds.

Verifying Correct Board Operation

This section lists the checks that you should make to ensure proper functioning of the board. You can also refer to the documentation related to this board in the *LatticeMico32/DSP Development Board User's Guide*. For part references, see Figure 1.

Checking the Power Supply

To verify that the power supply is functioning properly, examine the two green power-on LED lights in the upper left of the board as it is positioned in Figure 1:

- ◆ The 2.5-V LED at the top left of the board should be lit.
- ◆ The 3.3-V LED just below the 2.5 V LED should be lit.

Note

All boards leave the manufacturer fully tested. In addition, the LatticeMico32/DSP development board is supplied with test programs so that you can check the operability of the unit at any time. See the *LatticeMico32/DSP Development Board User's Guide* for details.

Ensuring Board/PC Communications

To ensure that the development board and the PC can communicate with each other properly, perform the following four procedures. This set of procedures involves downloading a predefined, low-level sample configuration bitstream to the board and verifying that the software tools recognize it.

This procedure assumes that you have set up the necessary hardware and software, as described in this chapter and the previous chapter.

Note

The steps that you performed to set up your USB cable and install your drivers in the “Setting Up the Development Board” on page 8 should have already established board-PC communication.

Reset your Factory Board Configuration

Reset your board configuration to its factory default, as described in “Restoring the Factory Configuration” on page 14.

Download the Sample Hardware Bitstream

Use ispVM System download software as follows to download the low-level bitstream program to configure the device:

1. From your PC desktop, choose **Start > Programs > Lattice Semiconductor > Accessories > ispVM System** to start the ispVM System software. For Windows XP, the path will begin with **Start > All Programs**.
1. In ispVM System, choose **ispTools > Scan Chain** or the Scan toolbar button. The software opens a new configuration file, scans the board connected to the PC, and lists the devices.

Note

If you have not run this before, ispVM System assumes a parallel cable, and this operation will fail. To avoid this, first set the cable type to USB by selecting Cable + I/O Setup, and configure this setup either manually or automatically. After the cable is configured, you can properly perform the **ispTools > Scan Chain** command.

2. In the Scan Configuration Setup dialog box, click on the device in the Device List column and select **LFECP33E** from the pop-up menu.
3. Double-click within the empty cell in the **FileName/IR-Length** column. The Device Information dialog box opens.
4. In the Device Information dialog box, click the **Browse** button for Data File and navigate to the `<isplever_installdir>\micosystem\platforms\PlatformA\Platform.bit` file.
5. Change the Operation setting to **Fast Program**.
6. Click **OK**.
7. Choose **Project > Download** or the **GO** button to program the device. The ispVM System software begins programming your device. As it is being programmed, the dual number display on the board cycles through 0-99 and the red LED lights below them cycle on and off.
8. When you notice the activity in the dual number display and the red LED lights have ceased, the programming is complete.

Create a Hardware Platform

You must establish a hardware platform using the LatticeMico32 System software so that you can later use the sample microprocessor application code to test the board.

1. Open the LatticeMico32 System with the MSB perspective open in one of the following two ways:
 - ◆ From the Windows desktop Start menu, choose **Start > Programs > Lattice Semiconductor > Accessories > LatticeMico32 System**.

- ◆ From within the Project Navigator, select **Tools > LatticeMico32 System** to open the software.
1. Accept the default workspace.
When the LatticeMico32 System opens, the MSB perspective will be available by default as indicated on the title bar.
 2. Choose **File > New Platform**.
 3. In the New Platform Wizard dialog box, select a platform name, for example, "**PlatA**," in the Platform Name text box.
 4. In the Directory text box, enter the file path, *<ispLEVER_dir>\micosystem* examples.
 5. In the Platform Templates list box, choose **PlatformA**.
 6. Click **Finish**.
 7. Choose **Platform Tools > Run Generator** or the **Run Generator** toolbar button.

Download the Executable from a Software Project

Finally, you must create a software project to download the sample application code to verify that the board is working properly.

1. In the LatticeMico32 System software, click on the **C/C++ icon** at the upper left beneath the toolbar to switch to the C/C++ perspective.
1. Choose **File > New > Mico32 Managed Make C Project**.
The New Project Wizard dialog box opens.
2. In the Project Name text box, enter **BoardTest**.
3. In the MSB System text box, browse to the .msb platform description file that you created in previous procedure and saved in the *<ispLEVER_dir>\micosystem* examples\PlatA\soc\PlatA.msb file.
4. In the Select Project Templates list box, click **LED7SegsTest** and click **Finish**.
5. In the Projects Pane on the left, right-click on the **BoardTest** project folder and select **Rebuild Project** from the pop-up menu.
6. Choose **Run > Run**.
7. Click on the **mico32 hardware** in the left panel, then click on **BoardTest**.
8. In the bottom left, click **New**.
9. Click **Run**.

This procedure should download the program into the board and start executing it. If you have correctly established board-to-PC communication, you should notice the LED lights on the board sequentially blinking on and off.

Note

The procedures that guide you through the design flow here to test board-PC communications are discussed in more detail in Chapter 4, "Running LatticeMico32 System Software" on page 15.

Restoring the Factory Configuration

You can restore the LatticeMico32/DSP development board to its original factory-programmed configuration by reprogramming both the ECP33 FPGA and the flash memory on the embedded processor module.

To reprogram the FPGA and the embedded processor module, perform the following steps:

1. Set up your PC to download to the development board, as described in “Setting Up the Development Board” on page 8.
1. Go to the following the URL on the Lattice Semiconductor Web site:
<https://www.latticesemi.com/products/intellectualproperty/ipcores/latticemico32/>
2. Right-click on the LatticeMico32/DSP development board bitstream (mico32.bit), and select **Save Link As** to save it on your local system.
3. Open the LatticeMico32 System software, and in the MSB perspective, select **Tools > Flash Programming**.
4. Follow the instructions for this operation described in “Downloading Executable Code to Flash Memory” on page 23, but navigate to the location to which you saved the development board bitstream file just mentioned.

When you have completed downloading this bitstream file to flash memory, the board will be reprogrammed back to its original state. If you have any problems with this procedure, contact Lattice Semiconductor Technical Support.

Running LatticeMico32 System Software

This chapter summarizes the major steps involved in using the software development tools for the LatticeMico32 Development Kit. This overview is intended to give you a quick tour of the interface and to show you how to perform major design steps within it. The *LatticeMico32 System Tutorial* provides detailed, step-by-step instructions on creating and programming a sample processor, and it is highly recommended that you begin with this tutorial.

Once you have connected the LatticeMico32/DSP development board to your computer, you can start the LatticeMico32 System integrated development environment from the ispLEVER software. This chapter assumes that you are familiar with the ispLEVER software interface and can navigate its various perspectives to perform tasks in major functional areas.

Creating an ispLEVER Project

Before starting a LatticeMico32 System project, you must first create an FPGA project in the Project Navigator in ispLEVER. Do not worry about adding source files or any details beyond what is described here. You add your Verilog source file that you generate using the LatticeMico32 System software later in the flow.

To create a new ispLEVER project for use with a LatticeMico32 project:

1. From your PC desktop, choose **Start > Programs > Lattice Semiconductor > ispLEVER Project Navigator**.
1. Choose **Environment > Options** and make sure **Use Project Wizard to Create New Design** is selected the Advanced tab in the Environment Options dialog box.
2. In the Project Navigator, create an ispLEVER project by choosing **File > New Project**.

The Project Wizard opens.

3. In the first dialog box of the Project Wizard, name your project, designate a file location, select **Verilog HDL** as your design entry type, select **Synplify** as your synthesis tool, and click **Next**.
4. In the Project Wizard - Select Device dialog box, select the following device options for use with this kit and click **Next**:
 - a. Family: **LatticeECP-DSP**
 - b. Device: **LFCEP33E**
 - c. Speed grade: **-4**
 - d. Package Type: **FPBGA484**
 - e. Operating Conditions: **Commercial**

Note

Currently, the LatticeMico32 platform is supported for use with LatticeEC, LatticeECP-DSP, LatticeECP2, LatticeECP2m, LatticeXP, LatticeSC, and LatticeSCM devices.

5. In the Project Wizard - Add Source dialog box, click **Next**.
6. In the Project Wizard - Project Information box, click **Finish**.

For more information on creating projects in ispLEVER, see the Project Navigator online Help system accessible from the Help menu.

Creating a New Platform

After you have created a new project in the Project Navigator, you must create a new microprocessor platform in the Mico System Builder (MSB). A platform generically refers to the hardware processor configuration that includes the CPU component, its peripheral components, and the interconnectivity between them.

To create a platform in MSB, you must generate the .msb file and add your components, as detailed in the following procedures.

Generating an .msb File

The first step in creating a new platform is generating an .msb file in MSB. This file will eventually contain a complete definition of your microprocessor platform.

To create a new processor platform description file:

1. In the Project Navigator, select **Tools > LatticeMico32 System**.

Note

You can also run the LatticeMico32 System software tools as a stand-alone application directly from the Windows desktop Start Menu by choosing **Start > Programs > Lattice Semiconductor > Accessories > LatticeMico32 System**.

1. Accept the default workspace if you are prompted to choose a workspace. This prompt appears first time that the system has been opened. A workspace defines how the underlying Eclipse software displays the LatticeMico32 system interface. You can create custom workspaces.
2. In the upper left-hand corner of the graphical user interface, select **MSB**, if it is not already selected, to open the MSB perspective.

A perspective in the LatticeMico32 System software is a separate, prearranged, and predefined combination of views, menus, commands, and toolbars in a given graphical user interface window that enable you to perform a set of particular tasks. The MSB perspective is one of three default perspectives in the software to help you to perform major functions. The C/C++ and Debug perspectives are the other two.

3. Choose **File > New Platform**.
4. In the New Platform Wizard dialog box, enter a platform name in the Platform Name box.
5. In the Directory box, browse to the directory in which you want to store your project files and click **OK**.
6. Click **Finish**.

You now have created an .msb file, which will hold the contents of your platform: a CPU, its peripherals, and the interconnections between them. Currently, the platform description contains no components.

Adding Components to Your Platform

There are component definitions for the processor and peripherals in MSB that you must add to your platform description.

To add the microprocessor core to the design:

1. Under CPU in the **Available Components** view, click on **LatticeMico32** to view the information available about the LatticeMico32 microprocessor.

Information about the LatticeMico32 microprocessor, including the parameters that you can set for it, now appears in the Component Help, or LatticeMico32, view and in the Component Attributes view in the lower

third of the screen. If you click on the  icon in the Component Help view, you can view a complete description of this component in a PDF file.

Similarly, if you click on this icon for a memory or a peripheral component, you can view information about that component.

1. Double-click on **LatticeMico32** to add the microprocessor component to your platform description. If you want to see information about it before you place it in the Editor, click on it.

The Add LatticeMico32 dialog box appears. The parameters in this dialog box correspond to those shown in the table in the Component Help view.

2. Set the options in the Add LatticeMico32 dialog box and click **OK**.

You also must add peripherals, for example, UARTs, GPIO, and others in the same way you added the microprocessor component. These, too, are added to your platform description.

For example, to add a UART peripheral component to the design:

1. Under IO in the Available Components view, double-click on **uart**.
1. In the Add UART dialog box, click **OK** to accept the default settings.
2. After you have added the last peripheral, specify the connections between the master and slave ports by clicking on the appropriate circles in the Connection column.

See the *LatticeMico32 System Tutorial* that accompanies this development kit for step-by-step instructions on these procedures.

Generating Component Addresses

The next step is to generate component addresses.

1. To change a component address, click on the appropriate address in the Base column and change the address.


Note

You can only edit the Base address. You cannot edit the End address. The value of the end address is equivalent to the value of the base address plus the value of the size.

1. To lock any addresses so that MSB will not automatically assign new addresses to the locked components, select the box for the appropriate components in the Lock column.

Note

To assign an address for only one component, lock all other components.

2. To automatically assign component addresses, choose **Platform Tools > Generate Address** or click the Generate Address toolbar button . You can also right-click in the Editor view and choose **Generate Address** from the pop-up menu.

3. After you have added the last peripheral, specify the connections between the master and slave ports by clicking on the appropriate circles in the Connection column.


Note

You can also create custom components to include in your platform description. See the *Creating Components in LatticeMico32 System* document for more details.

Assigning Component Interrupt Priorities

Assign an interrupt request priority (IRQ) to all components that feature a dash in the IRQ column of the Editor view. You cannot assign interrupt priorities to components lacking this dash in the IRQ column, such as memories and CPUs.

To assign interrupt priorities for all components other than memories and the CPU:

- ◆ Choose **Platform Tools > Generate IRQ** or click the Generate IRQ toolbar button . You can also right-click in the Editor view and choose **Generate IRQ** from the pop-up menu.

Performing Design Rule Checks

To ensure that your design conforms to the design rules for a given device, run a design rule check (DRC).

To perform a design rule check and verify the addressing:

- ◆ Choose **Platform Tools > Run DRC** or click the Run DRC toolbar button . You can also right-click in the Editor view and choose **Run DRC** from the pop-up menu.

Saving Your Platform

After you do a number of tasks to set up your platform, you should save your changes.

To save your platform changes in MSB:

- ◆ In the MSB perspective, choose **File > Save**. This operation specifically saves any changes you made to your .msb file.


Generating Your Platform

When you generate your platform, MSB updates the .msb file and does the following:

- ◆ Assigns addresses to components without locked addresses
- ◆ Assigns interrupt lines
- ◆ Performs design rule checking (DRC)
- ◆ Generates platform Verilog structural implementation

- ◆ Generates platform Verilog instantiation file
- ◆ Copies hardware/software implementation support files for the instantiated components

To generate your processor platform in MSB:

- ◆ In the MSB perspective, choose **Platform Tools > Run Generator** or click the Run Generator toolbar button . You can also right-click on your project in the Project view and choose **Run Generator** from the pop-up menu to perform the same operation. You can leave this program open or save your changes and close it.

Generating the Microprocessor Bitstream

In this flow, you would now return to ispLEVER to import the Verilog file generated by MSB and specify the connections from the microprocessor to the chip pins by importing an .lpf file. You can optionally perform functional simulation and timing simulation. Finally, you will build the database; map, place, and route the design; and generate the bitstream to download to the chip on the board.

Importing the Verilog File

First, you must import the Verilog file output by MSB.

To import the Verilog file output by MSB:

1. In the ispLEVER Project Navigator, choose **Options > Environment**.
2. Click on the **General** tab.
3. If the “Copy source file to project directory for Source-Import command” is selected, turn it off, and click **OK**.
4. Choose **Source > Import**.
5. In the Import File dialog box, browse to the location of your .v file and click **Open**.
6. In the Import Source Type dialog box, select **Verilog Module** and click **OK**.
7. In the Choose Top Module dialog box, select the platform name from the drop-down menu and click **OK**.

Connecting the Microprocessor to FPGA Pins

You have two options for connecting the microprocessor to the FPGA pins:

- ◆ Manually create the pin constraints and import them into ispLEVER.
- ◆ Import a pre-created constraints file that is part of the platform templates into ispLEVER.

The *LatticeMico32 System Tutorial* allows you to import the pin constraints specified for the LatticeMico32 PlatformE template platform into ispLEVER. MSB copies the .lpf file associated with a platform template, if it exists, into the ..\soc directory of your LatticeMico32 project. A detailed explanation of how you assign pin constraints is beyond the scope of this document but an import procedure is provided here.

To import the .lpf file:

1. In the Project Navigator, choose **Source > Import Constraint/ Preference File**.
2. In the Import Preference File dialog box, browse to the .lpf file, and click **Open**.
3. Click **OK** at the prompt dialog boxes that tell you where the logical preference file is being saved and indicate whether you wish to reset the project update status.

Generating the Bitstream

Now you generate a bitstream to download the processor to the FPGA. If you did not perform timing simulation, the bitstream generation process automatically builds a database and synthesizes, maps, places, and routes the design before it generates the bitstream.

To generate a bitstream (.bit) file:

1. In the Sources in Project pane of the Project Navigator, highlight the part name, for example, **LFCEP33E-4F484C**.
1. In the Processes for Current Source pane, double-click **Generate Bitstream Data**.

The ispLEVER software now generates the bitstream in a programming (.bit) file that is ready for downloading onto the device.

Downloading the Hardware Bitstream to the FPGA

This section describes how to download the hardware bitstream to the FPGA on the board to program your device. The instructions in this procedure are specific to the design used in the kit, and you should refer to the tutorial provided to perform step-by-step tasks.

To download the hardware design reference file to the device:

1. Start the ispVM System software by choosing **Tools > ispVM** in the Project Navigator. Alternatively, you can choose **Start > Programs > Lattice Semiconductor > Accessories > ispVM System**.

Note

If you do not have ispVM installed, click on the ispVM icon in the Project Navigator toolbar or go to the Lattice Semiconductor Web site to download the program.

1. In ispVM, choose **File > New**.
2. Choose **Edit > Add Device**.
3. In the Device Information dialog box, click the **Select** button in the Device section to open the Select Device dialog box.
4. In the Select Device dialog box, select **LatticeECP** in the Device Family box.
5. In the Device box, select your target device, **LFCEP33E**.
6. In the Package Type box, select your package, **484-ball fpBGA**.
7. Click **OK**.
8. In the Data File section of the Device Information dialog box, use the Browse button to navigate to the bitstream (.bit) file, select the bitstream file, and click **Open**.
9. In the Operation box, choose, for example, **Fast Program**.
10. In the Device Access Options box, choose an option, for example, **JTAG 1532 Mode**.
11. Click the **Go** button to initiate the download.
A processing status bar shows you the progress of the download.
12. Check the Chain Configuration status to see if the download passed.

At the end of this process, the FPGA is loaded with the microprocessor hardware configuration.

Creating the Software Application Code

To create and debug the software application code that programs the LatticeMico32 processor, you return to the LatticeMico32 System interface and activate the C/C++ SPE tool and the Debugger. Refer to the *LatticeMico32 System Tutorial* and the *LatticeMico32 Software Developer User's Guide* for more explicit instructions to guide you through this development flow.

To create a new software project:

1. From the MSB perspective, click the **C/C++** button on the upper left.
The C/C++ perspective opens.
1. In the C/C++ perspective, choose **File > New > Mico32 Managed Make C++ Project**. You can choose either C or C++ here, depending on which language you are using to write your base application code.
The New Project dialog box opens.
2. In the Project Name box, enter the project name.
3. In the MSB System text box, browse to the location of your project's .msb file, if it is not already selected by default, select it, and click **Open**.

4. In the Select Project Templates box, select the template for the application code.
5. Click **Finish**.
Your software project has been created.
6. Click on the project name to select it in the C/C++ Projects view (left-hand pane)
7. Choose **Project > Build Project**.

You can perform subsequent builds by right-clicking the project name and selecting **Rebuild Project**.

Debugging the Software Application Code

You can debug your software application code using the Debugger's Instruction Set Simulator (ISS), which is available in both the C/C++ perspective and the Debug perspective. The instructions in this procedure are specific to the design used in the kit, and you should refer to the tutorial provided to perform step-by-step tasks.

To run a debug session using the ISS:

1. In the C/C++ Perspective's Projects view, right click on the project you just created and choose **Run > Debug** from the pop-up menu.
2. In the Debug dialog box, click **Debug**.
3. In the Confirm Perspective Switch box, click **Yes**.

This procedure activates the Debug perspective and downloads the .elf file into the FPGA. In addition, the Debug perspective allows a more robust environment for all aspects of debugging the application code and testing it before it is ready for download.

See the online Help included in the LatticeMico32 System interface for further instructions on debugging. You can also refer to the *LatticeMico32 Software Developer User's Guide* and the *LatticeMico32 System Tutorial* for further guidance.

Downloading Executable Code to Flash Memory

After creating your program executable code for the processor using C/C++ SPE, you can download it to flash memory on the board so that the processor defined in your circuit can use the code to perform process functions. It is also possible to deploy to on-chip memory if enough memory is available.

When deploying your software off-chip to a flash memory, you can either deploy your application code to parallel flash memory, which is volatile memory, or to SPI flash memory, a non-volatile memory. You must have peripheral components defined in your platform for this task, and templates

are included in the LatticeMico32 System software to allow you to do this with relative ease.

Because of the more complex nature of this part of the flow, a quick reference procedure is not provided here. See the *LatticeMico32 System Tutorial* for instructions on downloading your code to a flash memory. It is highly recommended that you run through the explicit steps in the tutorial and also refer to the “Software Deployment” section of the *LatticeMico32 Software Developer’s Guide*, which contains detailed information on general software deployment strategies.

Recommended References

As mentioned previously, this chapter provides a brief tour of the interface and procedures to perform major design flow tasks to get you familiar with the software tools. You should use the *LatticeMico32 System Tutorial* that accompanies this development kit to learn the step-by-step instructions for creating a sample LatticeMico32 processor.

The *LatticeMico32 Software Developer’s Guide*, available within the LatticeMico32 System installation, describes the flow, tools, and other advanced programming topics. This document is a good place to begin for a programmer to learn more about writing software for the LatticeMico32 processor. In addition, to create custom components for the processor, see the *Creating Components in LatticeMico32 System* document.

Index

A

AC adaptor cord **3**
AC connector **9**
AC power supply **9**
Add LatticeMico32 dialog box **18**
Add UART dialog box **18**
adding components to platform **17**
assigning interrupt priorities **19**

B

.bit file **21**
bitstream
 downloading configuration **12**
 downloading to FPGA device **21**
 generating microprocessor **20, 21**
building database **20**
building projects **23**

C

C/C++ perspective **13, 17, 22**
C/C++ Software Project Environment see C/C++
 SPE
C/C++ SPE
 creating software application code **22**
 downloading sample application code **13**
 purpose **5**
Choose Top Module dialog box **20**
Confirm Perspective Switch box **23**
connecting microprocessor to FPGA pins **20**
constraints file **20**
Creating Components in LatticeMico32 System
 document **19, 24**
creating new ispLEVER projects **15**
creating platform in MSB **16**
creating software application code **22**

creating software project **13**

D

Debug dialog box **23**
Debug perspective **17, 23**
Debugger
 debugging software application code **23**
 error messages issued by **9**
 purpose **5**
debugging software application code **23**
design rule checks (DRC) **19**
Device Information dialog box **12, 22**
Device Manager dialog box **10**
download cables **9**
downloading bitstream to FPGA device **1, 21**
downloading sample hardware bitstream **12**
downloading software application code to parallel
 flash controller **1, 23**

E

electrostatic discharge (ESD) **8**
Environment Options dialog box **15**
ezusb.sys file **10**

F

factory configuration **14**
flash memory see parallel flash controller
FPGA pins **20**
functional simulation **20**

G

Generate Address toolbar button **18**
Generate IRQ toolbar button **19**
generating component addresses **18**
generating microprocessor bitstream **21**

generating platform **19**

H

hardware platform **12**

I

Import File dialog box **20**

Import Preference File dialog box **21**

Import Source Type **20**

importing .lpf file **20, 21**

importing Verilog file **20**

Install dialog box **9**

installing LatticeMico32 System software **5**

instruction set simulator (ISS) **23**

interrupt request priority (IRQ) **19**

ispDOWNLOAD USB drivers **9**

ispLEVER

 creating project **15**

 importing .lpf file **20**

 importing Verilog file **20**

 software set **4**

 version compatibility with LatticeMico32
 System software **5, 6**

ispLEVER 6.1 Installation Notice document **4, 5**

ispVM System

 downloading hardware bitstream to FPGA **21**

 downloading sample hardware bitstream **12**

 included in ispLEVER software set **4**

 installing **5**

 required for downloads **6**

 starting **9**

ispvmsystem folder **10**

K

keyboards **7**

L

LatticeECP33 FPGA device **7**

*LatticeMico32 Development Tools Installation
Guide* document **5, 6**

LatticeMico32 microprocessor **5, 18, 20**

LatticeMico32 Software Developer User's Guide
document **22, 23, 24**

LatticeMico32 System software

 creating ispLEVER project **15**

 default installation directory **5**

 installing **4, 5**

 perspectives in **5**

 re-installing **6**

 set contents **4**

 system requirements **6**

LatticeMico32 System Tutorial document **15, 18,
21, 22, 23, 24**

LatticeMico32/DSP development board

 communicating with PC **11**

 connecting to PC **1**

 creating hardware platform **12**

 devices equipped with **7**

 downloading executable from software
 project **13**

 downloading sample hardware bitstream **12**

 layout **8**

 purpose **7**

 reference design **3, 7, 8**

 resetting factory board configuration **12**

 restoring factory configuration **14**

 setting up **1, 8**

 stand-alone **4**

 verifying correct operation of **11**

*LatticeMico32/DSP Development Board User's
Guide* document **3, 4, 8, 11**

LED lights **7, 11, 12, 13**

.lpf file **20, 21**

M

MachXO CPLD device **7**

mapping the design **20, 21**

Mico System Builder see MSB

mico32.bit file **14**

MSB

 adding components to platforms **17**

 assigning interrupt priorities **19**

 creating hardware platform **12, 16**

 generating .msb file **17**

 generating component addresses **18**

 generating platform **19**

 performing design rule checking (DRC) **19**

 purpose **5**

 saving platform **19**

.msb file **16, 17, 19**

MSB perspective **13, 17**

N

New Platform Wizard dialog box **13, 17**

New Project dialog box **22**

New Project Wizard dialog box **13**

non-volatile memory **23**

O

on-chip memory controller **23**

P

parallel flash controller **23**

PDF file **18**

perspectives

 C/C++ **17**

 Debug **17**

 MSB **17**

placing the design **20, 21**

power supply **11**

Project Navigator **15, 20**

Project Wizard **16**

Project Wizard - Select Device dialog box **16**

R

reference design **3, 7, 8**

resetting factory board configuration **12**
restoring factory configuration **14**
routing the design **20, 21**
Run DRC toolbar button **19**
Run Generator toolbar button **20**

S

saving platform **19**
Scan Configuration Setup dialog box **12**
Scan toolbar button **12**
Select Device dialog box **22**
software application code
 creating in C/C++ SPE **22**
 debugging **23**
 downloading to parallel flash controller **23**
SPI flash memory **23**
synthesizing the design **21**
System Properties dialog box **10**
system requirements **6**

T

test pads **7**
timing simulation **20, 21**

U

USB cable **3, 9**
USB drivers **9, 10**
USB port **6**

V

Verilog file **15, 20**
volatile memory **23**