



LatticeMico32 System Linux Installation Guide

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

April 2008

Copyright

Copyright © 2008 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, FlashBAK, flexiFlash, flexiMAC, flexiPCS, FreedomChip, 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, LatticeECP, LatticeECP-DSP, LatticeECP2, LatticeECP2M, LatticeMico8, LatticeMico32, LatticeSC, LatticeSCM, LatticeXP, LatticeXP2, MACH, MachXO, MACO, ORCA, PAC, PAC-Designer, PAL, Performance Analyst, PURESPEED, Reveal, 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, Bringing the Best Together, and More of the Best are service marks of Lattice Semiconductor Corporation.

HyperTransport is a licensed trademark of the HyperTransport Technology Consortium in the U.S. and other jurisdictions.

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 IN THIS DOCUMENT, 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

LatticeMico32 System Linux Installation Guide	1
System and Software Requirements	1
System Requirements	1
Software Requirements	1
Contacting Lattice Semiconductor	2
Installation and Setup Requirements	2
Hardware	2
Software	3
Synthesis Tool Requirements	3
Additional Notes for Linux	3
Installation Procedure	4
ispLEVER Embedded LatticeMico32 Installation and Usage	4
LatticeMico32 System Stand-Alone Installation and Usage	5



LatticeMico32 System Linux Installation Guide

This document explains how to install LatticeMico32 System on the Linux operating system.

System and Software Requirements

Your system must meet the following minimum system and software requirements to run LatticeMico32 System on the Linux operating system.

System Requirements

- ◆ Intel Pentium or Pentium-compatible PC with USB port
- ◆ 32-bit Red Hat Enterprise Linux version 4.0 operating system
- ◆ 2 GB memory recommended for FPGAs
- ◆ Approximately 436 megabytes of free disk space
- ◆ 1024 x 768 graphics display

Software Requirements

- ◆ ispLEVER 7.1
- ◆ For Verilog users: Synplicity[®] Synplify Pro[®] or Mentor Graphics[®] Precision[®] RTL Synthesis
- ◆ For VHDL users: Synplicity Synplify Pro version 8.9 or later

Note

Synplicity Synplify Pro is required for LatticeMico32 MSB generation.

- ◆ ispVM System version 17.2 or later. You can download ispVM System from the following Lattice Semiconductor Web site:

<http://www.latticesemi.com/products/designsoftware/ispvmsystem/index.cfm>

- ◆ Perl version 5.8.0. The following Perl modules are required:

- ◆ XML::DOM
- ◆ Getop::Std
- ◆ Scalar::Util
- ◆ FindBin::Bin

These modules are normally found in the native version of Perl that comes with Red Hat Linux.

- ◆ PDF browser, such as Adobe Acrobat
- ◆ Mozilla or Netscape browser (optional)

Contacting Lattice Semiconductor

You can contact Lattice Semiconductor by any of the following means:

- ◆ Lattice Semiconductor Corporation
5555 Northeast Moore Court
Hillsboro, Oregon 97124-6421 U.S.A.
- ◆ Internet: www.latticesemi.com
- ◆ Literature Hotline: 1-888-ISP-PLDS (477-7537)
- ◆ Applications Support
Domestic: 1-800-LATTICE (528-8423)
International: (503) 268-8001
Fax: (503) 268-8556

Installation and Setup Requirements

Installing and setting up LatticeMico32 System on the Linux operating system has the following hardware and software requirements.

Hardware

For LatticeMico32 System hardware designers who are using Mico System Builder (MSB):

- ◆ ispVM System must be installed in the ispLEVER directory in the Linux server, for example, `<isptools_dir>/ispvmsystem`.

Refer to the *Installation Guide for ispVM System Linux* at the following Web site:

<http://www.latticesemi.com/products/designsoftware/ispvmsystem/index.cfm>

- ◆ LatticeMico32 System must be installed in the ispLEVER directory in the Linux server, for example, `<isptools_dir>/micosystem`.

Software

For LatticeMico32 software developers who are using the Software Programming Environment (SPE) only:

- ◆ ispVM System can be installed in any location. Refer to the *Installation Guide for ispVM System Linux* at the following Web site:
<http://www.latticesemi.com/products/designsoftware/ispvmsystem/index.cfm>
- ◆ LatticeMico32 can be installed in any location.

For this installation type, the Mico System Builder extension to Eclipse will not function unless ispLEVER is also installed.

Synthesis Tool Requirements

Since the synthesis tools are not included with the Linux version of ispLEVER software tools, you will need the following:

- ◆ Verilog users – You must synthesize the LatticeMico32 MSB-generated Verilog files in Synplicity Synplify Pro or Mentor Graphics Precision RTL Synthesis to create an EDIF file for bitstream generation.
- ◆ VHDL users – You must use Synplicity Synplify Pro version 8.9 or later.

You must set up Synplicity Synplify Pro so that LatticeMico32 System can access it through an environment variable:

```
SETENV IPEXPRESS_SYN_PATH=xxxxxxxxxxx
```

where xxxxxxxx is the name of the Synplicity Synplify Pro executable file. Here is an example, assuming that Synplify Pro was installed in `/public/install/synplify/fpga8804`:

```
SETENV IPEXPRESS_SYN_PATH=/public/install/synplify/fpga8804/bin/synplify_pro
```

You must also synthesize the LatticeMico32 MSB-generated VHDL file in Synplicity Synplify Pro to create an EDIF file for bitstream generation.

Additional Notes for Linux

You may need the following information to run LatticeMico32 on the Linux operating system:

- ◆ For Linux in the ispLEVER Project Navigator, the design entry type can only be EDIF in the New Project Wizard.
- ◆ If you want to configure your system to use an external browser instead of the Eclipse help browser, you can change this in the LatticeMico32 main window menu by selecting Window > Preferences.

- ◆ If the LatticeMico32 software is in a central server location, only one instance of LatticeMico32 software can be run for each user login, whether you are logged on one host machine or on multiple host machines.
- ◆ Eclipse requires that you set the MOZILLA_FIVE_HOME environment variable to the location of the Mozilla or Firefox library directory before you run the LatticeMico32 software.
- ◆ If you cannot access the PDF files referenced in the LatticeMico32 online Help, you can configure LatticeMico32 to use an external browser by following these steps:
 - a. In the LatticeMico32 software, click on **Window > Preferences**.
 - b. Click on **Help**.
 - c. Select **Use External Browser**.
 - d. Click on **Web Browser**.
 - e. Select **Use External Web Browser**.
 - f. Select **Mozilla** or **Netscape**.
 - g. Click **OK**.

Installation Procedure

You can install LatticeMico32 embedded in ispLEVER, or you can install it as a stand-alone tool.

ispLEVER Embedded LatticeMico32 Installation and Usage

The LatticeMico32 System installation embedded in ispLEVER allows both the Mico System Builder (MSB) and the Software Programming Environment (SPE) to function.

To install LatticeMico32 System embedded in ispLEVER on a Linux machine for release 7.1:

1. Install ispLEVER version 7.1 first, for example, in /usr/local/isp lever.
2. Go to the isptools directory:

```
cd /usr/local/isp lever/isptools
```
3. Install LatticeMico32, assuming that the build source is in /media/cdrom.
4. In the isptools directory, type the following on the command line:

```
/media/cdrom/mico32_install.csh /media/cdrom
```
5. Answer the installation questions.
The installer now untars the files.
6. After the installation is finished, verify the following:

- ◆ The /micosystem directory is created in isptools, for example, /usr/local/ispLever/isptools/micosystem.
 - ◆ In the LatticeMico32Launcher file in the /isptools/micosystem directory, verify that the micoPath variable has the correct path to /micosystem.
7. From ispLEVER, create a project using an FPGA that supports LatticeMico32, such as LatticeECP, LatticeSC, or LatticeXP2.
The LatticeMico32 System icon appears on the toolbar and in the Tools menu.
 8. If you want to run LatticeMico32 System outside the ispLEVER Project Navigator, do the following:
 - a. In the /micosystem directory, type the following on the command line:

```
source ../ispcpld/bin/setup_lv.csh
```
 - b. Type:

```
launchmicosystem
```

LatticeMico32 System Stand-Alone Installation and Usage

The LatticeMico32 System stand-alone installation only provides functionality for the Software Programming Environment (SPE) portion of LatticeMico32 System. Mico System Builder (MSB) will not function with this type of installation.

The stand-alone installation requires the installation of ispVM System version 17.2 or later.

To set up the environment in the Korn shell, use the latticemico32.sh file.

To install stand-alone LatticeMico32 System on a Linux machine:

1. Create a directory, for example, lm32_build91.

```
cd /usr/local
mkdir lm32_build91
```
2. Go to the new directory, as in this example:

```
cd lm32_build91
```
3. Install LatticeMico32 System, assuming that the build source is in /media/cdrom.
4. In the lm32_build91 directory, type the following on the command line:

```
/media/cdrom/mico32_install.csh /media/cdrom
```
5. Answer the installation questions.
The installer now untars the files.
6. After the installation is finished, verify the following:
 - ◆ The /micosystem directory is created in isptools.

- ◆ In the LatticeMico32Launcher file in the /isptools/micosystem directory, verify that the "micoPath" variable has the correct path to /micosystem.
7. Start LatticeMico32 System:
- ◆ On the command line in the micosystem directory, type the following:

```
cd micosystem
./LatticeMico32Launcher
```
 - ◆ From outside /micosystem, such as in lm32_build91, type the following:

```
/usr/local/lm32_build91/micosystem/LatticeMico32Launcher
```