Lattice PCI Express Demo User's Guide

April 2007 Technical Note TN1123

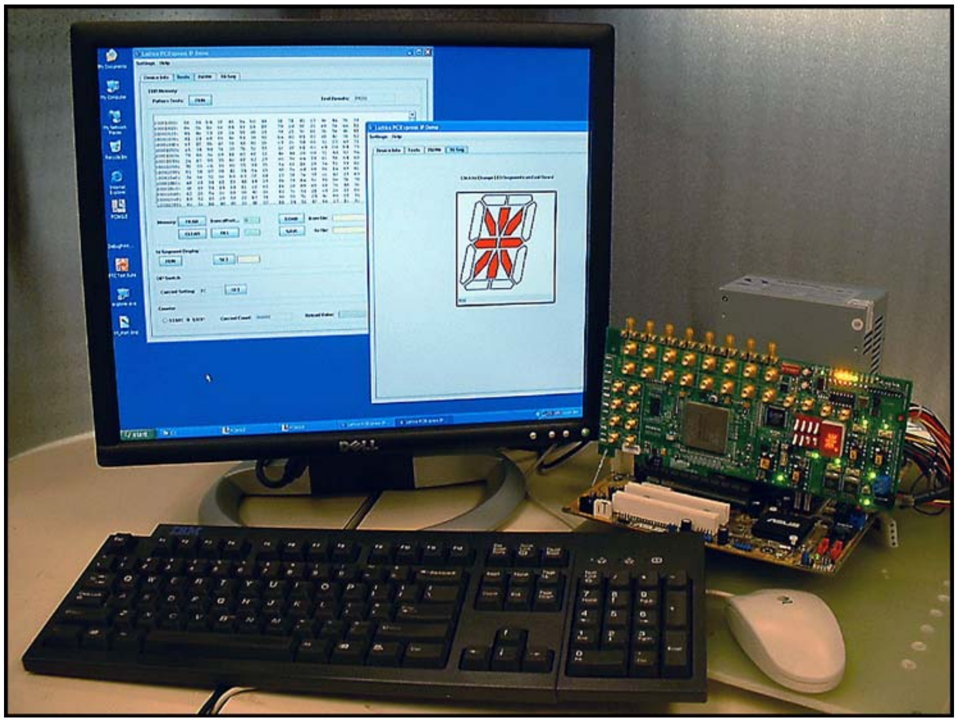
Lattice PCI Express Demo Overview

Introduction

This user's guide describes how to install and run the Lattice PCI Express (PCIe) Endpoint IP demo. The demo consists of hardware, IP and software. The demo shows the capabilities of the Lattice FPGA and the PCI Express IP core functioning in a PCI Express slot in a Windows PC. The demo is designed to be simple and easy to use; and requires no test equipment. The demo uses a Windows application to allow the user to access memory and registers on the board. This provides real-time interaction with the evaluation board hardware to demonstrate a functional PCI Express communications path between the application and driver software (running on the PC CPU) and the FPGA IP. Device driver and application source code are available so a user can modify and extend the behavior of the tests or use them as a starting point for their own experiments or PCIe design.

The demo package consists of a Lattice PCI Express evaluation board, Lattice FPGA bitstream, Windows driver and Windows application all running in a Windows PC. The evaluation board and the Lattice FPGA bitstream provide the PCI Express hardware and the driver and application provide the software to allow the user to exercise the PCI Express link. Figure 1 shows a typical lab setup using a PC motherboard without a case.

Figure 1. Typical Lab Setup



The PCI Express Demo application software operates with any of the Lattice PCI Express evaluation boards, operating in any lane size, with any PCI Express demo bitstream. The software is not affected by the FPGA type (LatticeSC™ or LatticeECP2M™) or lane size (x1, x4, x8, x16). These are physical and data link layer details. The application software operates at the Application layer (layer 7) of the OSI stack model, far removed from the physical details (layers 1 and 2) of how the data is transferred from one location to another. Therefore, PCI Express lane size is not directly discussed with respect to the operation of the demo software. It is irrelevant from the view point

© 2007 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

of the application. As long as a link is established and the hardware can be seen by the device driver, the PCI Express demo will function properly.

In addition, the PCI Express demo software does not control the PCI Express lane size or operating mode. The PC BIOS and/or Windows sets these factors during boot-up. The PCI Express demo software detects the hardware (registers and EBR) in the evaluation board FPGA through a bus device driver model provided by Windows. Register reads and writes are abstracted by the Windows bus driver and their operation is handled internally by Windows, the HAL, BIOS and possibly root complex register settings. The driver and application software are Windows WDM compliant. This means that they only use the APIs exposed by Windows to perform operations on the hardware. No attempt is made to directly control the root complex's PCI Express behavior.

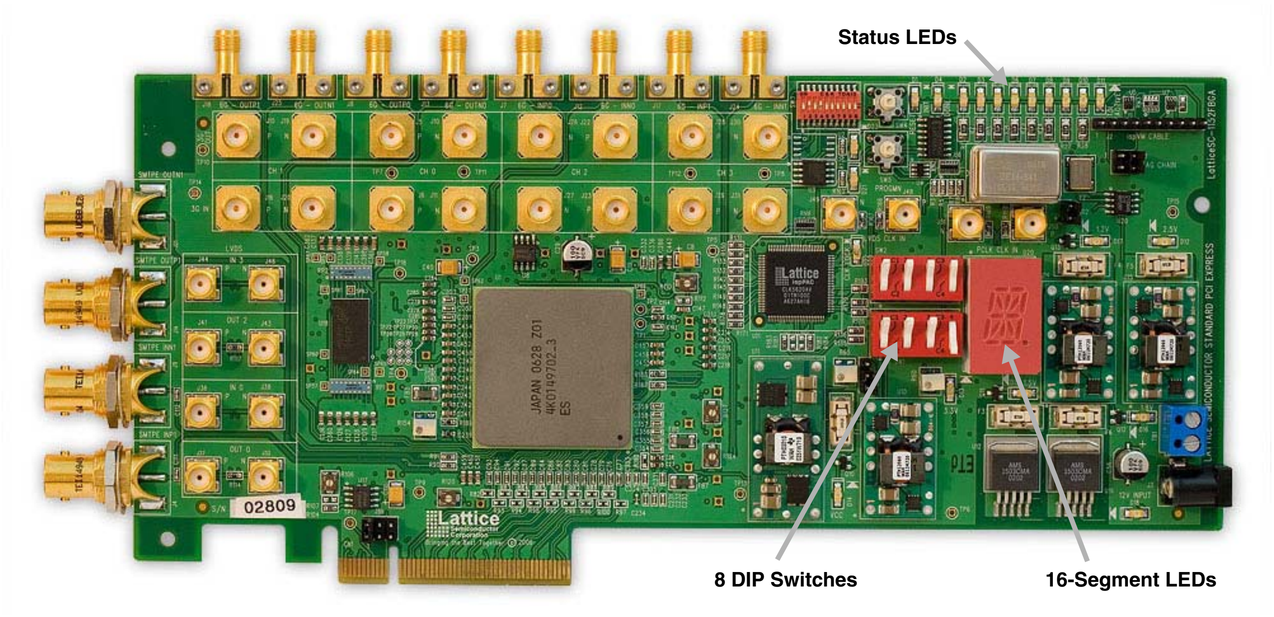
In order to demonstrate the low-level PCI Express connection protocols, a test equipment exerciser and analyzer are needed (or at least an analyzer between the RC and the endpoint). The demo described in this manual does not have the ability to control, or deliver insight into any packets transmitted on the PCI Express link. An external piece of test equipment is needed to generate, capture and examine training sequences, flow control, ACKs, NAK, message packets, configuration packet, memory packets, etc.

Table 1. Demo Setups

Evaluation Board	PCI Express Slot Used	Hardware Adapter Required	PCI Express IP Source
LatticeSC-80 PCI Express x8	x4 PCI Express	x4 Adapter	LatticeSCM MACO flexiMAC™ and LTSSM plus Soft IP
LatticeSC-80 PCI Express x8	x1 PCI Express	x1 Adapter	LatticeSCM MACO flexiMAC and LTSSM plus Soft IP
LatticeSC-25 PCI Express x1	x1 PCI Express	None	LatticeSCM MACO flexiMAC plus Soft IP
LatticeECP2M-35 PCI Express x4	x4 PCI Express	None	LatticeECP2M Soft IP
LatticeECP2M-35 PCI Express x4	x1 PCI Express	x1 Adapter	LatticeECP2M Soft IP

The demo software runs on a Windows PC (32-bit Windows 2000 and XP have been tested; Server2003 may also work; 64 bit OS versions have not been tested). The application program provides access to the LatticeSC™ PCI Express x8, LatticeSC PCI Express x1 or LatticeECP2M PCI Express x4 evaluation boards. The following figures show the target PCI Express evaluation boards.

Figure 2. LatticeSC80 PCI Express x8 Evaluation Board



The LatticeSC80 board has a 16-segment LED display that can be written to by the demo application. The board also has eight DIP switches. The states of these DIP switches are read by the PCI Express demo software and displayed to the user. The Status LEDs are set by the application IP to indicate the states of the PCI Express IP and link.

The LatticeSC80 board has a physical PCI Express x8 connector. Electrically it is wired to eight FPGA SERDES. It has the potential operate in x1, x4 or x8 PCI Express mode. Currently the LatticeSCM MACO PCI Express IP core only supports x1 and x4 modes. A Presence jumper must be set to x1 or x4 to match the intended the operating link width. Since all eight lanes are physically connected to SERDES, the root complex detects all eight lanes during the detect stage. There is potential for the root complex to become confused by detecting all eight lanes electrically, but the board is loaded with a bitstream that supports only x4 PCI Express operation. To avoid this situation, a x4 adapter is used with the LatticeSC80 board to make it appear electrically as a x4 board. The PC root complex then only detects 4 lanes and operation continues as a x4 link.

Consult the Lattice web site (www.latticesemi.com) for the latest news on support of PCI Express x8 with the LatticeSCM.

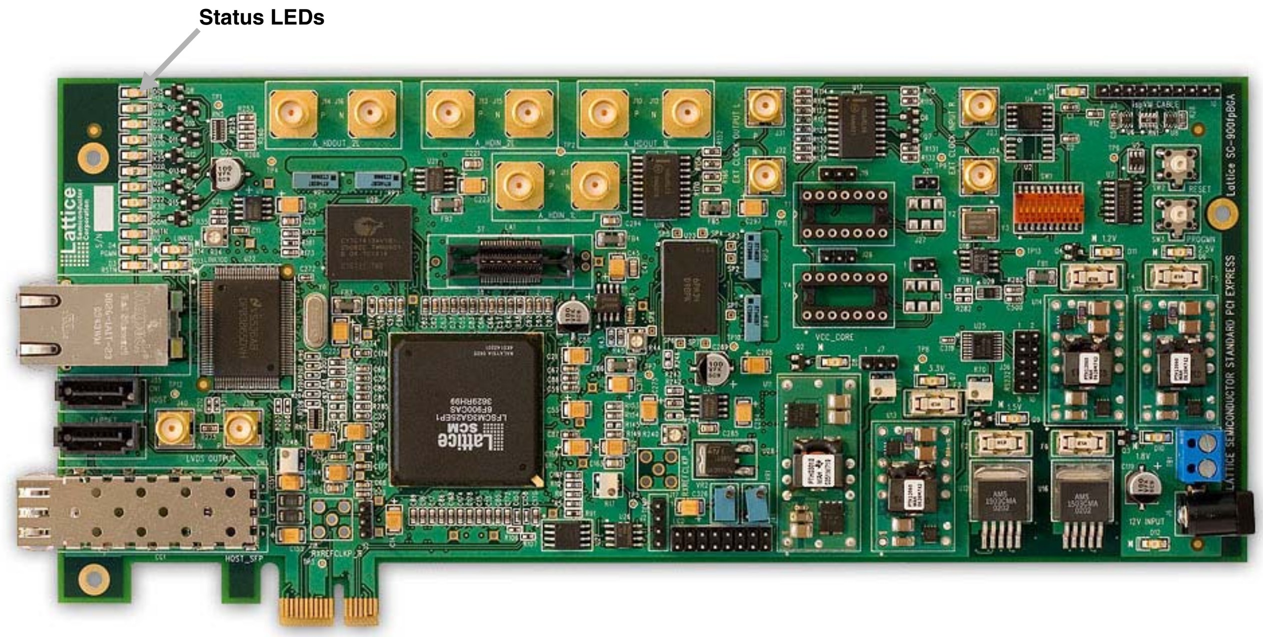
The same situation is true for operation in x1 mode. If a bitstream is loaded that only supports x1 PCI Express mode, then the x1 adapter needs to be installed so the board electrically appears as if it only has one PCI Express link, and the other lanes are not detected.

When the LatticeSC80 board is used with the x1 adapter, it can be installed in any size slot in a PC. All slots must support the ability to fall back to x1 mode. When the LatticeSC80 board is used with the x4 adapter it physically will only fit into x4 or larger slots. There is a potential for a root complex to not support x4 mode in a x8 or x16 slot. The root complex could decide that since the board is not the native size (board links not equal to slot links) it will fall back all the way to x1 and not an intermediate stage (i.e., x4). This potential issue most likely would arise in x16 graphics slots where the PC BIOS expects a x16 (or x8), only sees a x4 and falls back to a x1. In such a case, the demo software will work normally. This shows one of the strong points of PCI Express: the ability to dynamically adapt to lane size without affecting the transport layer.

In summary, for currently available x1 and x4 demonstrations, an adapter is always required when using the LatticeSCM80 PCI Express x8 board. A x4 adapter is required for use with bitstreams that support x4 mode, or a x1 adapter can be used with any bitstream.

For more information on the operation of LatticeSC80 board, please refer to the *LatticeSC PCI Express x8 Evaluation Board User's Guide* for the board, available at: www.latticesemi.com/boards.

Figure 3. LatticeSC25 PCI Express x1 Evaluation Board



The LatticeSC25 PCI Express x1 evaluation board does not have the 16-segment display nor the DIP switches. This board can still be used with the PCI Express demo, but the operations that deal with the 16-segment display or DIP switches will have no affect on the board, since they are not physically present. The EBR memory tests can be used with this board, but the direct, interactive demos will not be of any use.

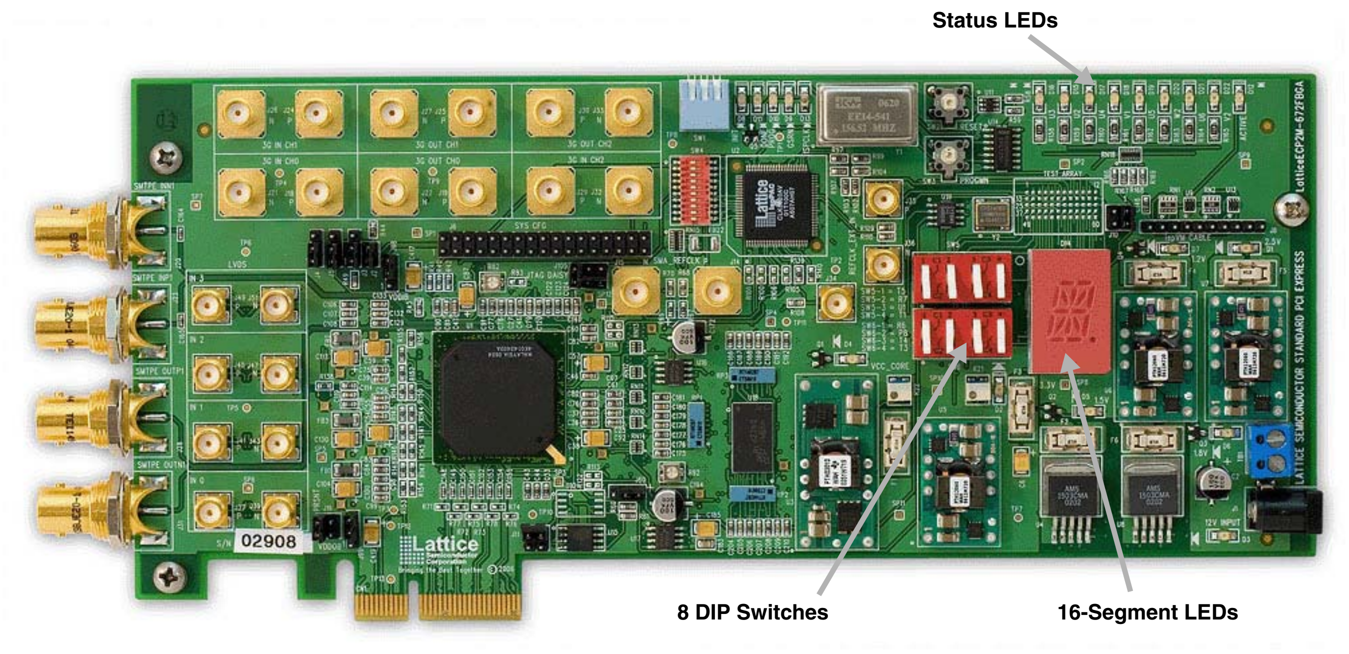
The LatticeSC25 board has a physical PCI Express x1 connector. Electrically it is wired to one FPGA SERDES. It can only operate in x1 PCI Express mode. The bitstream that is loaded must be set the PCI Express IP core as x1 in the LatticeSCM.

The LatticeSC25 board is used without any adapters. It can be installed in any size slot in a PC. All slots must support the ability to fall back to x1 mode. The demo software will work with this board installed in any PCI Express slot

In summary, an adapter is not required when using the LatticeSC25 PCI Express x1 board. It only supports x1 operation.

For more information on the operation of this board, please refer to the *LatticeSC PCI Express x1 Evaluation Board User's Guide*, available at: www.latticesemi.com/boards.

Figure 4. LatticeECP2M PCI Express x4 Evaluation Board



The LatticeECP2M PCI Express x4 evaluation board is very similar to the LatticeSC80 PCI Express x8 evaluation board. Both boards include the 16-segment LED display and the DIP switches.

The LatticeECP2M board has a physical PCI Express x4 connector. Electrically it is wired to four FPGA SERDES. It has the potential to operate in x1 or x4 PCI Express mode. The Lattice ECP2M PCI Express IP can support both x1 or x4 mode. A Presence jumper must be set to x1 or x4 to match the intended the operating link width. Since all four lanes are physically connected to SERDES, the root complex detects all four lanes during the detect stage. There is potential for the root complex to become confused when it sees four electrical lanes, but the board is loaded with a bitstream that supports only x1 PCI Express operation. To avoid this situation, a x1 adapter is used with the LatticeECP2M to make it appear electrically as a x1 board. The PC root complex then only detects 1 lane and operation continues as a x1.

If a bitstream is loaded that support x4 operation, a x1 adapter can also be used to force the board to operate as a x1.

When the LatticeECP2M board is used with the x1 adapter, it can be installed in any size slot in a PC. All slots must support the ability to fall back to x1 mode. When the LatticeECP2M board is used without the x1 adapter it physically will only fit into x4 or larger slots. There is a potential for a root complex to not support x4 mode in a x8 or x16 slot. The root complex could decide that since the board is not the native size (board links not equal to slot links) it will fall back all the way to x1 and not an intermediate stage (i.e., x4). This potential issue most likely would arise in x16 graphics slots where the PC BIOS expects a x16 (or x8), only sees a x4 and falls back to a x1. In such a case, the demo software will work normally. This shows one of the strong points of PCI Express: the ability to dynamically adapt to lane size without affecting the transport layer.

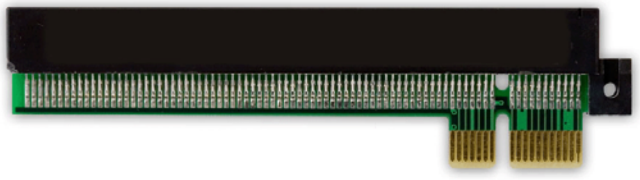
In summary, for currently available x1 and x4 demonstrations, a x1 adapter is always required when using the LatticeECP2M PCI Express board for x1 operation.

For more information on the operation of this board, please refer to the *LatticeECP2M PCI Express x4 Evaluation Board User's Guide,* available at: www.latticesemi.com/boards.

Adapter Boards

Each evaluation board needs to be installed in an available PCI Express slot in a PC. Since some of these boards are physically x4 and x8 connectors, an adapter board can be used to install an evaluation board in a PCI Express x1 slot. Figure 5 shows an adapter that allows any of the evaluation boards to be used in a x1 slot.

Figure 5. Adapter Board



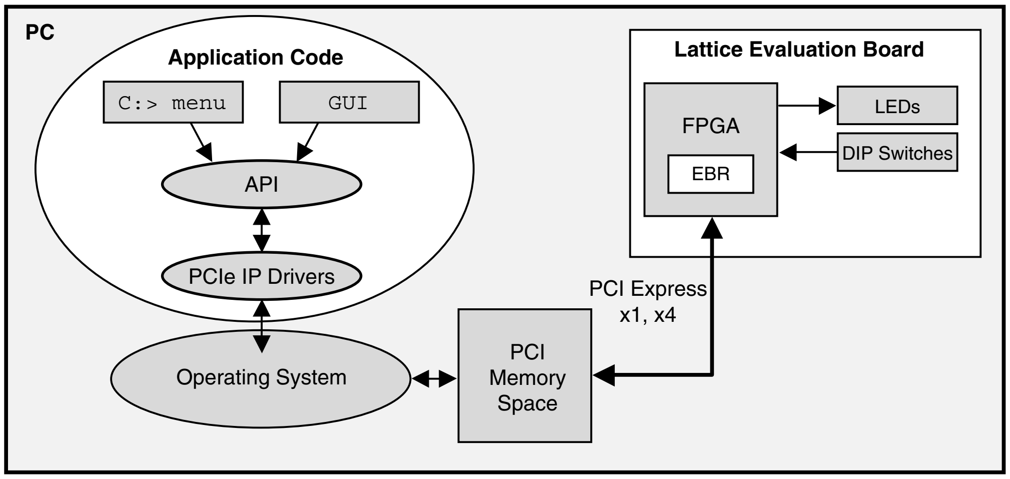
When using an adapter card only the lanes of the adapter card that plug into the motherboard are terminated. SERDES inputs that are connected to SERDES channels that are powered up should not be unterminated. For this reason the user should create a special bitstream for use with the adapter card.

Note: In the LFSCM80 PCI Express demo package there is a file named scm80_x1.txt which describes the procedure to create a x1 bitstream.

Demo Operations Overview

As mentioned, the demo runs on a standard Windows PC, and accesses the Lattice PCI Express evaluation board that is installed in a PCI Express slot. Figure 6 shows the relationship of the hardware and software components of the demo.

Figure 6. PCI Express Block Diagram



The PCI Express IP core in the Lattice FPGA acts as a PCI Express endpoint. A PCI Express endpoint device looks like a regular PCI device to application software executing on a PC. It is a memory-mapped device occupying a certain range(s) of the PCI memory space. When the PC boots, the BIOS and OS probe the PCI Express and PCI buses and detect the devices present on the buses and assigns them ranges in the PCI memory space. The PCI memory space is mapped into the application software's memory space by the supplied driver and OS system calls. Once the mapping is done, application IP registers, sitting on top of the PCI Express IP stack, can be read/written as memory locations by the application software. The demo software writes to the LED register, reads and displays the DIP switch setting and reads/writes the EBR memory.

The demo software shows that the Lattice PCI Express IP core correctly handles the PCI Express protocol in a PC through the interaction with the devices on the evaluation board. The demo exercises the following functions:

1. **Displays Operating System information** on the detected Lattice evaluation board.
2. **Displays information about the PCI Express IP core**, such as reading and displaying all the pertinent information in the configuration registers, extended capability registers and control registers.
3. **Performs GPIO Register Access**: blink LED's; read DIP switches and display value.
4. **Performs Memory Access**: writes a pattern of values into internal FPGA EBR memory, reads back and verifies that all accesses are error-free.

Current Limitations

The demo hardware and software have been validated on an Intel 975XBX motherboard running Windows XP. The demo may operate in other systems, but this is the recommended setup. The current demo software and demo IP design do not currently support some advanced features/applications. The following features are not demonstrated:

- No DMA
- No hardware interrupts or MSI
- No multiple VC's or traffic classes
- No multiple evaluation board operation. The demo cannot select a specific board.

Background Knowledge

This demo assumes the user is familiar with basic PCI Express technology and is comfortable installing new hardware and software packages in a Windows PC. Some experience in these areas is helpful to install the evaluation board and the software.

Installation Guide

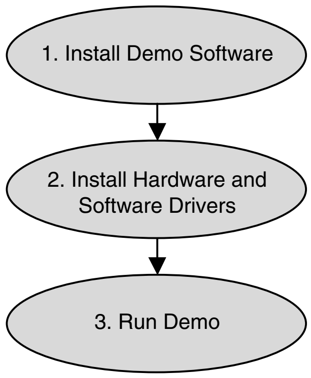
The Lattice PCI Express demo package is released as a Windows install executable. The package includes the Windows driver, FPGA bitstreams, Java GUI, Java runtime, and all demo source code. The executable needs to be run to unpack all the files. After the files are installed on the PC, the drivers are available to be installed when the evaluation board hardware is installed and detected.

This section discusses installing the demo package. The installation of the software and evaluation board hardware are discussed. Please read this section completely before attempting to install the package so that you understand the steps involved and how they apply to your specific situation. An optional simulated evaluation board driver can be installed if you do not have access to an evaluation board. This installation is covered in Appendix A.

Install Overview

The following three steps are taken to install and run the demo:

Figure 7. Installation Procedure



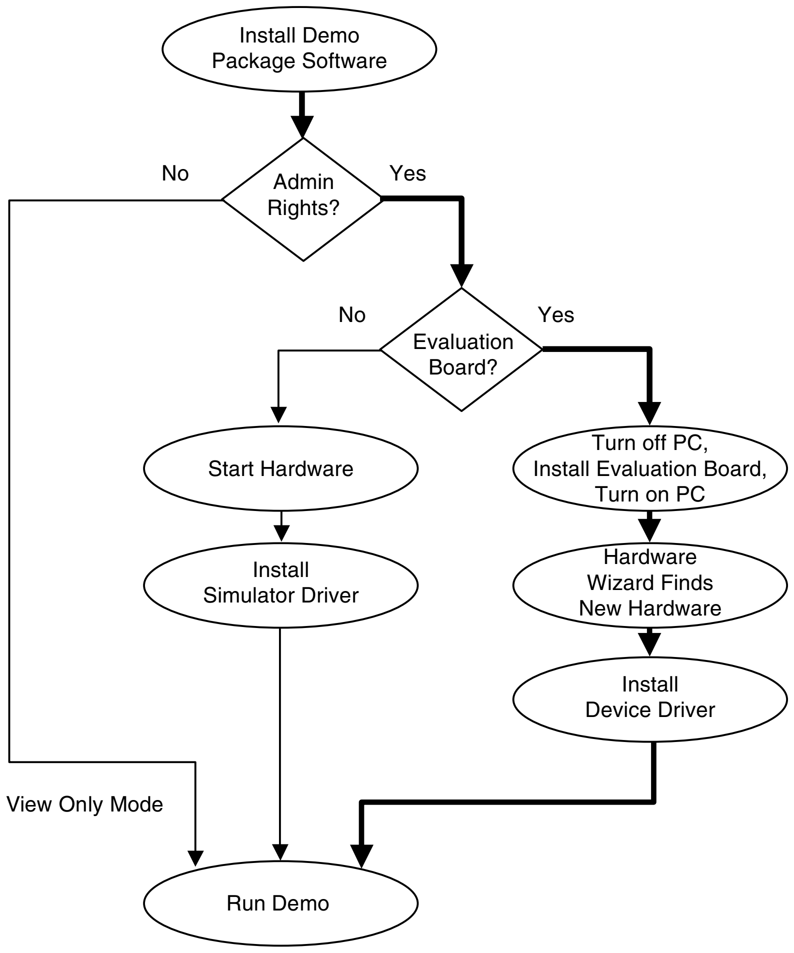
The demo can be run in three different operating modes. Depending on your system privileges and the availability of the evaluation board hardware, you may not be able to install the full hardware demo. You may then select to run a simulated version, to become familiar with the capabilities and see how the drivers work. If you do not have the correct install privileges you may only be able to run the GUI in "view-only" mode and not be able to install or access any drivers or hardware.

Table 2. Operating Mode Summary

Mode	Requirements	Permission	Description
Hardware Demo	Need evaluation board in a PCI Express slot	Admin rights	Full access to Lattice FPGA hardware through PCI Express bus
Simulated Hardware	Software only	Admin rights	Simulated reads/writes
View Only	Software only	Anyone	View the GUI only

Figure 8 shows how to install the software to support any of the three modes (hardware demo, simulator or view only). The sections that follow detail the steps necessary to install the demo software and hardware.

Figure 8. Installation Flow Chart

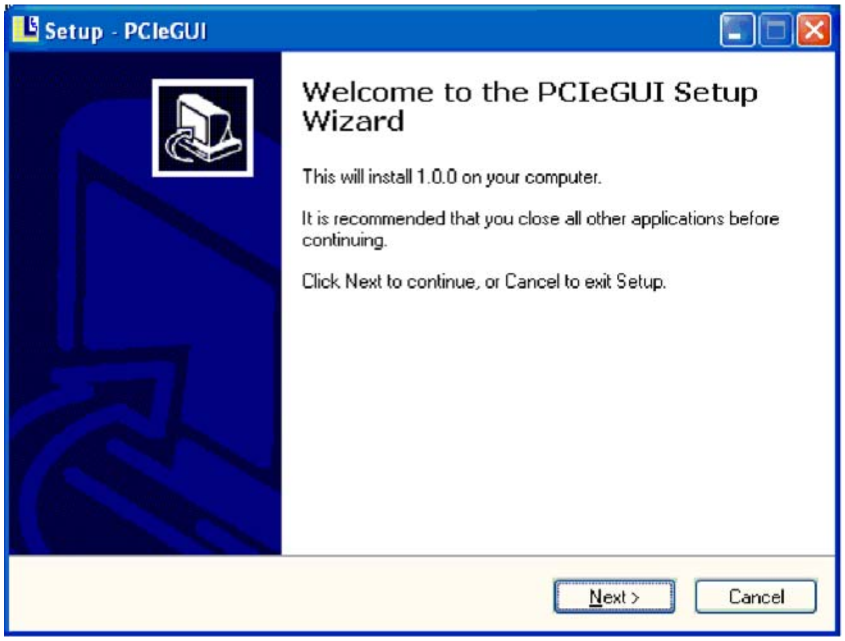


Demo Package Software Installation

Double click on the **PCIeGUI_setup.ex**e program to start the installation process. You may need admin rights to install in the default location of **C:\Program Files**. If you don't have rights, then install in your local directory. This will allow you to run the demo GUI in view-only mode and you will be unable to install hardware device drivers.

Double-click on the **PCIeGUI_setup.exe** install program to start the installation process.

Figure 9.



Setup - PCIeGUI

Welcome to the PCIeGUI Setup Wizard

This will install 1.0.0 on your computer.

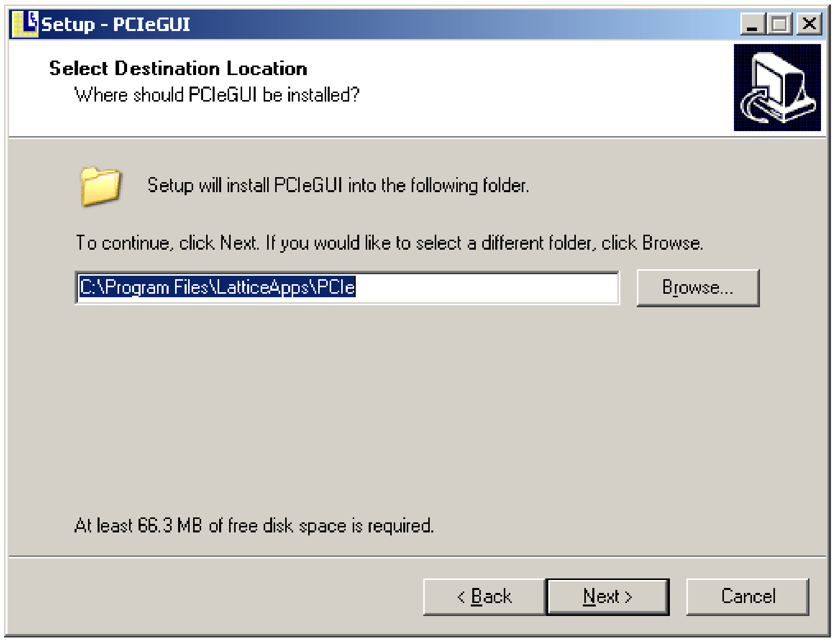
It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Click **Accept** to accept the license and then **Next** to view the installation notes.

The next page lets you choose the installation location. The default location of **c:\Program Files** is acceptable for normal operation (you may need admin rights to install programs there on your machine). If you intend to develop with the device driver DDK, the spaces in the path will create problems during building, so you may want to consider installing elsewhere (no spaces in directory names).

Figure 10.



Setup - PCIeGUI

Select Destination Location
Where should PCIeGUI be installed?

Setup will install PCIeGUI into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

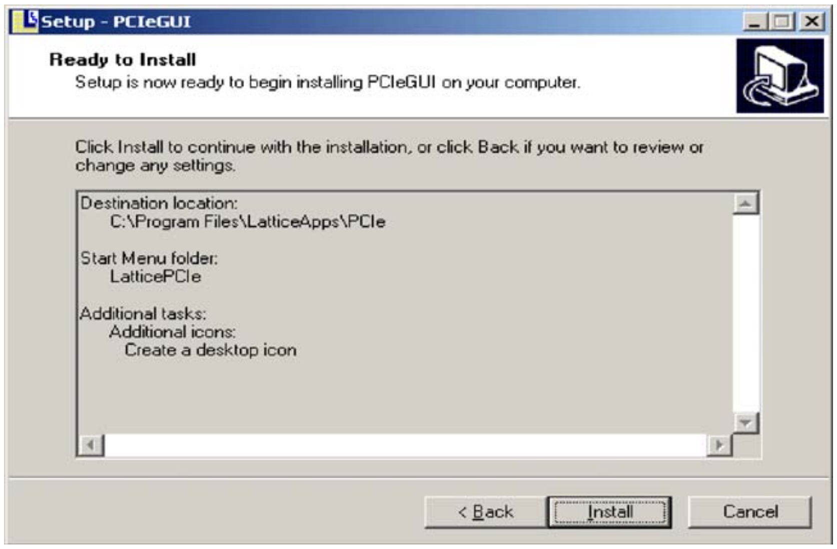
C:\Program Files\LatticeApps\PCIe

At least 66.3 MB of free disk space is required.

Click **Next** and then click Next again to accept the start menu program group name. Select **Create Desktop Icon**.

The Default Install configuration will be:

Figure 11.



Ready to Install
Setup is now ready to begin installing PCIeGUI on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

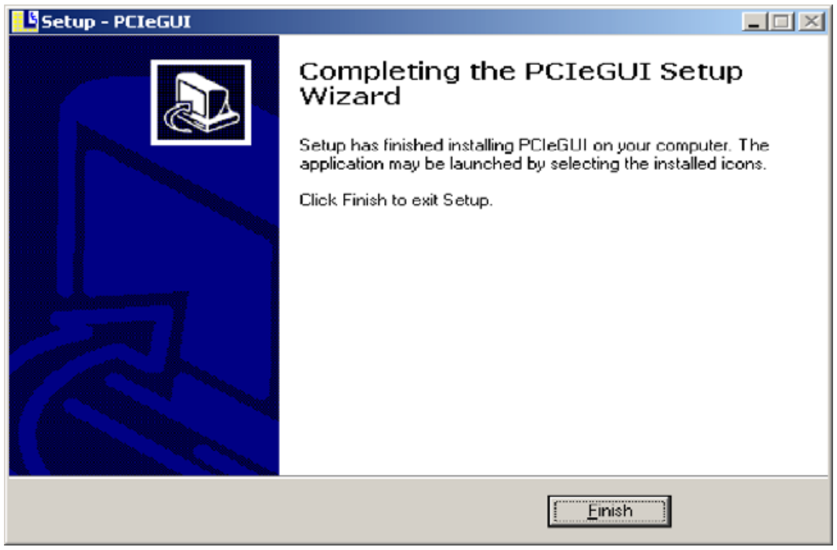
Destination location:
    C:\Program Files\LatticeApps\PCIe

Start Menu folder:
    LatticePCIe

Additional tasks:
    Additional icons:
        Create a desktop icon

Click **Install** to begin the process.

Figure 12.



Completing the PCIeGUI Setup Wizard

Setup has finished installing PCIeGUI on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

After a successful installation, the above screen will appear and an icon (PCI Express GUI) will be on the desktop, as well as a Lattice PCI Express program group from the **Start->Programs** menu bar.

Evaluation Board Hardware Installation

At this point, if you have an evaluation board and the necessary permissions (admin rights) you can install the hardware.

IMPORTANT: Shut down the PC and then unplug the power cord. PC power supplies have standby voltages that are present even when the PC power light (and fan) is turned off. Unplugging the PC is the safest way to ensure the board will not be "hot-swapped".

Note: The FPGA bitstream must already be loaded into the SPI flash on the evaluation board. The FPGA must be programmed with the PCI Express IP core when the PC powers up so that the endpoint is seen by the North Bridge root complex and the BIOS and OS allocate resources for it. If the FPGA is loaded with the bitstream (via ispVM® System software) after the PC has booted, it will need to be restarted (soft reset, not a power-off) so that the board is seen when the PC boots.

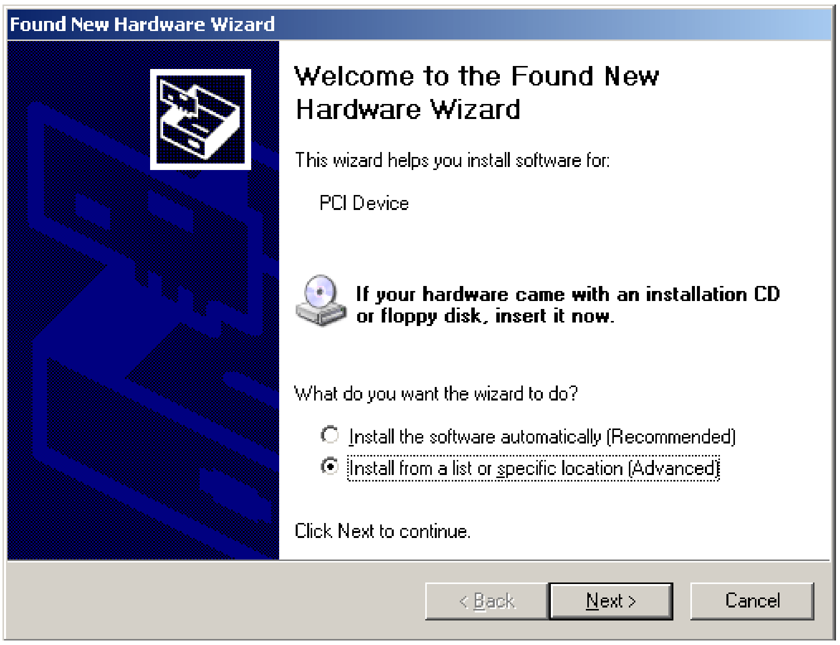
To install the Lattice PCI Express evaluation board:

1. Shut down Windows and the PC.
2. Unplug the power cord.
3. Locate an available PCI Express slot. Using ESD precautions, install the Lattice PCI Express Evaluation Board in the PCI Express slot. You may need an adapter board to convert the board to a x1 slot.
4. Power-on the PC and observe that it boots normally to the Windows login screen. Consult the Trouble Shooting section of this document if anything abnormal occurs.
5. Log in as a user with admin rights. During the login process Windows will detect the new hardware and ask if you want to install it.

Evaluation Board Driver Installation (Windows XP)

This section describes installation of the evaluation board device driver software on a Windows PC. The screen shots included are from installation on an XP system. The Windows 2000 procedure differs slightly. The two operating systems have slightly different dialog box options, but the process is very similar. The "Found New Hardware" screen will appear when the PC is booted for the first time with the evaluation board installed. If this screen does not appear, the evaluation board was not properly detected by the PC BIOS or Windows. Refer to the Trouble Shooting section of this document for further information.

Figure 13.


Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

PCI Device

If your hardware came with an installation CD or floppy disk, insert it now.

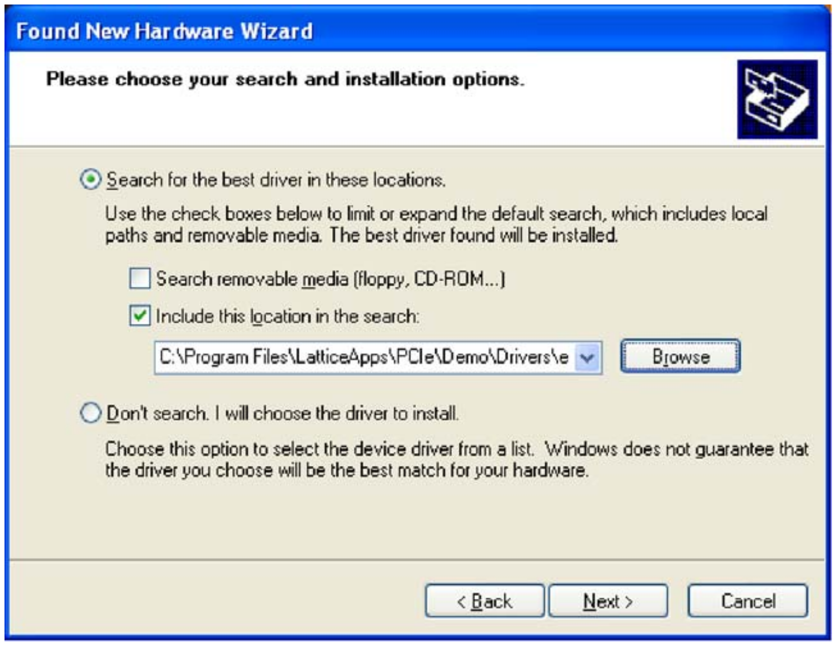
What do you want the wizard to do?

( ) Install the software automatically (Recommended)
(●) Install from a list or specific location (Advanced)

Click Next to continue.

Select the **Install from specific location** option and click **Next**. Browse to where you installed the demo package, locate the **Demo\Drivers** directory and select the **EvalBoard** folder. The default location of the folder is: **C:\Program Files\LatticeApps\PCIe\Demo\Drivers\evalBoard**.

Figure 14.



Found New Hardware Wizard

Please choose your search and installation options.

(●) Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

[ ] Search removable media (floppy, CD-ROM...)

[✓] Include this location in the search:

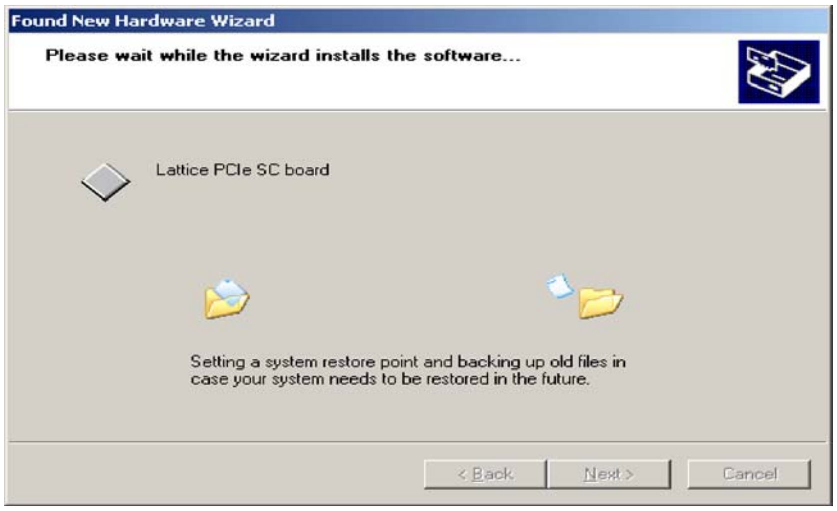
C:\Program Files\LatticeApps\PCIe\Demo\Drivers\e

( ) Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Click **Next** to continue.

Figure 15.



Found New Hardware Wizard

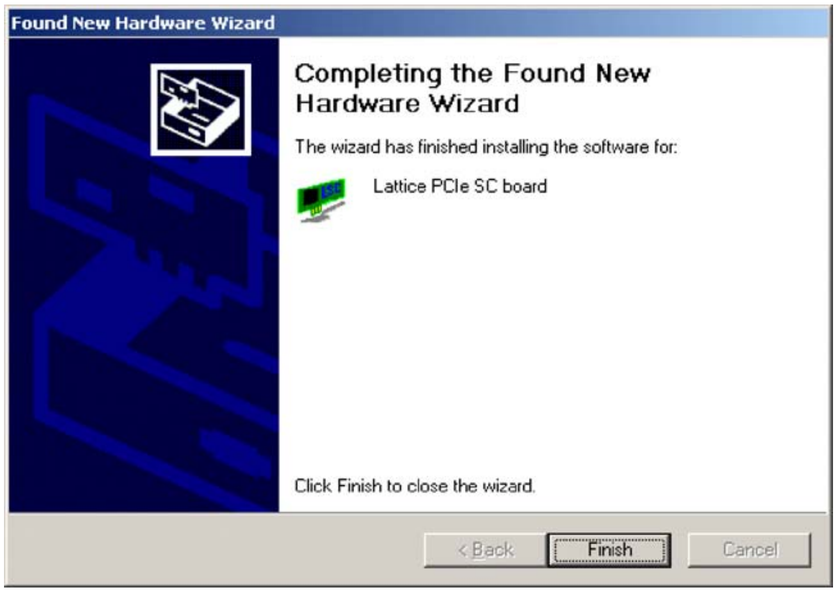
Please wait while the wizard installs the software...

Lattice PCIe SC board

Setting a system restore point and backing up old files in case your system needs to be restored in the future.

Windows now copies the driver files and will display the following screen when finished.

Figure 16.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

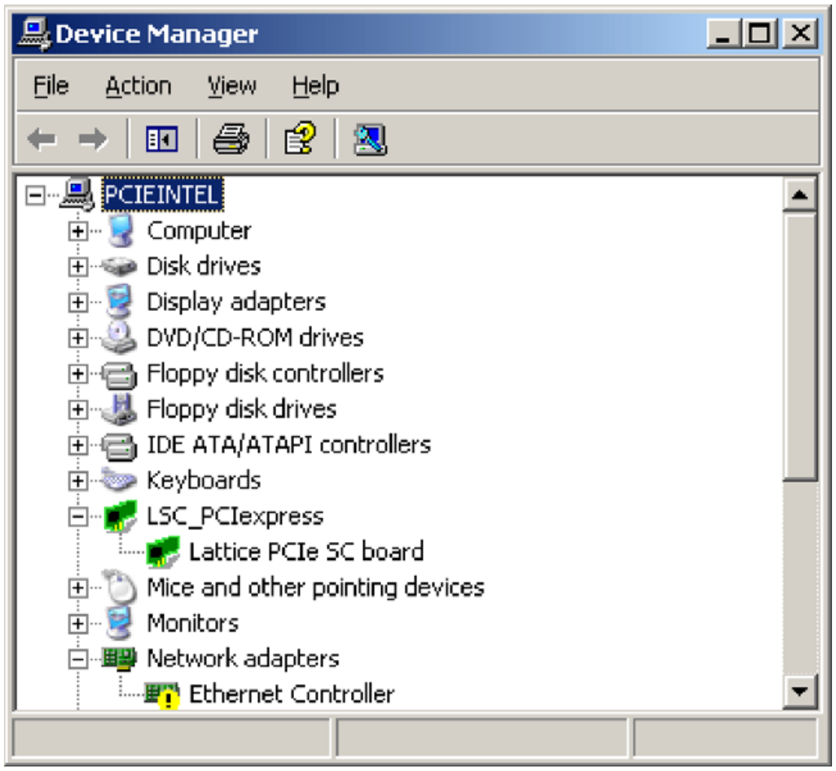
The wizard has finished installing the software for:

Lattice PCIe SC board

Click Finish to close the wizard.

To confirm the installation and verify that Windows properly sees the evaluation board hardware, right-click on **My Computer** (on the Desktop) and choose **Properties**. Select the **Hardware** tab and press the **Device Manager** button. The Lattice Evaluation Board (LSC_PCI Express) will appear in the list of hardware devices in your system.

Figure 17.



```
PCIEINTEL
  Computer
  Disk drives
  Display adapters
  DVD/CD-ROM drives
  Floppy disk controllers
  Floppy disk drives
  IDE ATA/ATAPI controllers
  Keyboards
  LSC_PCIexpress
    Lattice PCIe SC board
  Mice and other pointing devices
  Monitors
  Network adapters
    Ethernet Controller
```

Right click on the **Lattice Eval Board** icon and select **Properties** to show the resources assigned to the device and the driver information. Memory ranges corresponding to the configured BAR registers will be assigned to the board. If this is all present, then the demo program is able to run and access the hardware on the evaluation board.

Installation in a Different Slot

Windows identifies PCI/PCI Express hardware devices using the bus, slot, vendor ID and device ID fields. If you install the board into a different slot, the slot number will change. This will cause Windows to display the "Found New Hardware" message box when the system powers up. The entire set of preceding steps to install the driver do not need to be taken, since the driver has already been installed. If the board is installed in a new slot, simply choose to let Windows search for the driver ("Install the software automatically (Recommended)" and install automatically. Windows will then associate the newly created device registry tag (bus, slot, vendor and device ID) with the lscpcie.sys driver and the demo GUI will work with the board in the new slot.

This same scenario will occur when installing a LatticeECP2M evaluation board after installing a LatticeSCM evaluation board. The LatticeECP2M has a different PCI device ID, so Windows considers it new hardware and needs to associate it with the lscpcie.sys driver that has already been installed.

Un-Installing the Software

If you want to remove the software from your system (e.g. to perform a clean installation of a new version), the Lattice PCI Express program group includes a remove program option which will remove the application code. To remove the driver code, use the "Uninstall/Unplug a device" option from the Hardware Wizard. Choose to un-install and select the Lattice PCI Express device.

Un-installing may require manual operations to remove all traces of driver files stored in the Windows system directories and the registry. The following steps assume you have admin rights and that no files are hidden by Explorer.

1. Go to C:\Windows\inf\ and remove any oemxx.inf files that have LSC in them
2. Remove C:\Windows\system32\lscinst.dll.
3. Remove any C:\Windows\system32\drivers\lsc.sys files.
4. Run regedit and search for and remove VirtualPCI or LSC_VirtualPCI or LSC_PCI Express tags or search for LSC (most keys are prefixed with this identifier).

Once all traces of the driver files have been removed, a clean installation is possible.

Environment Setup

This section notes additional setup that may be necessary before running the demo. At this time, everything is self-contained in the demo installation directory and no additional programs or settings need to be present.

Java

The GUI is written in Java and uses the Java Swing libraries for display. The Java 1.5.0 run-time libraries are included in the release and installed in the demo directory. No additional setup is required and the Java files included with the demo will not interfere with any other Java installations.

Environment Variables

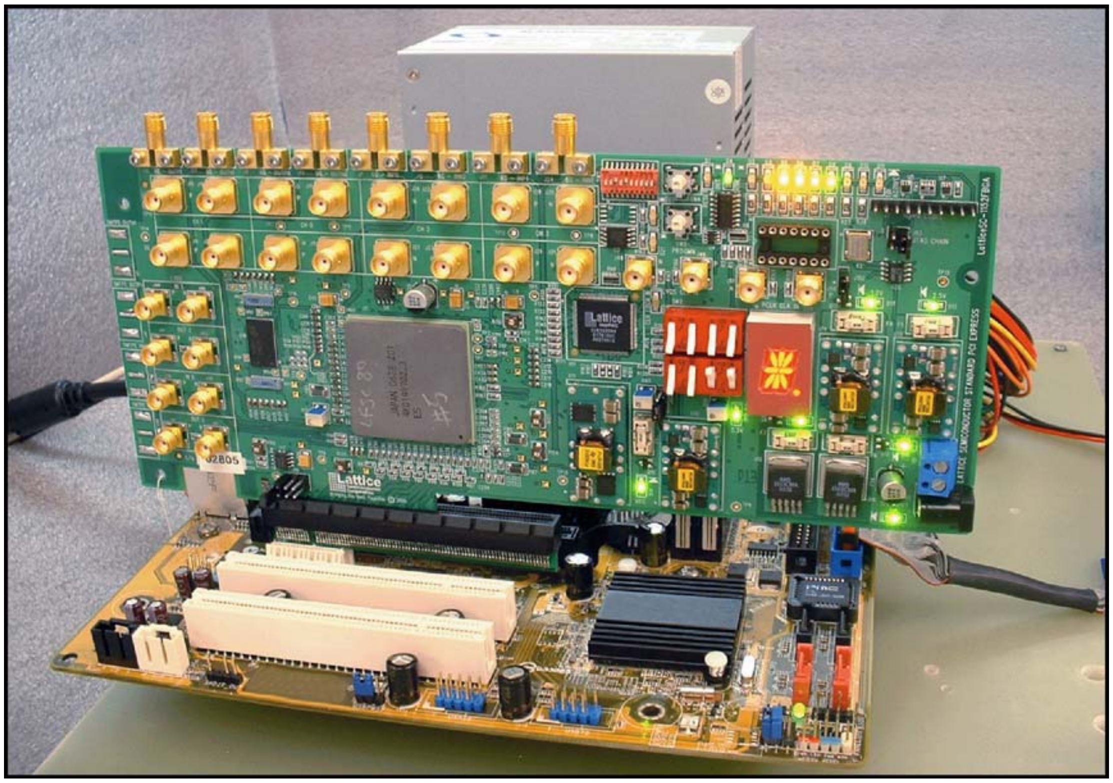
The following environment variables are used to identify the type of board that the demo will talk to. The options from **Start -> Programs -> LatticePCIe** set the appropriate values when running. Therefore, these should not need to be set, but are included here for completeness.

- PCIE_BOARD = SC – if the evaluation board is a LatticeSC board
- PCIE_BOARD = EC – if the evaluation board is a LatticeECP2M board
- PCIE_BOARD = SIM – if no evaluation board hardware or simulator is installed

Running the Demo

Figure 18 is an example of an operational demo.

Figure 18. Operational Demo



This is a LatticeSCM80 evaluation board installed in a standard PC motherboard. Note the 16-segment display has been set by the demo to display an "*". The four status LEDs at the top right are lit, indicating a functional PCI Express link. The decimal point on the 16-segment display is lit showing PCI Express memory reads and writes are taking place over the bus.

Demo Setup

No setup is required if the demo programs are run from the **Start->Programs->LatticePCIe** program group. To run from the command line, the environment variable PCIE_BOARD needs to be set in the shell you are running from.

When the PC if powered on, the status LEDs will light to indicate a good communication link over the PCI Express bus. See Appendices B-D for details on LED status, jumpers and switch settings for all Evaluation boards.

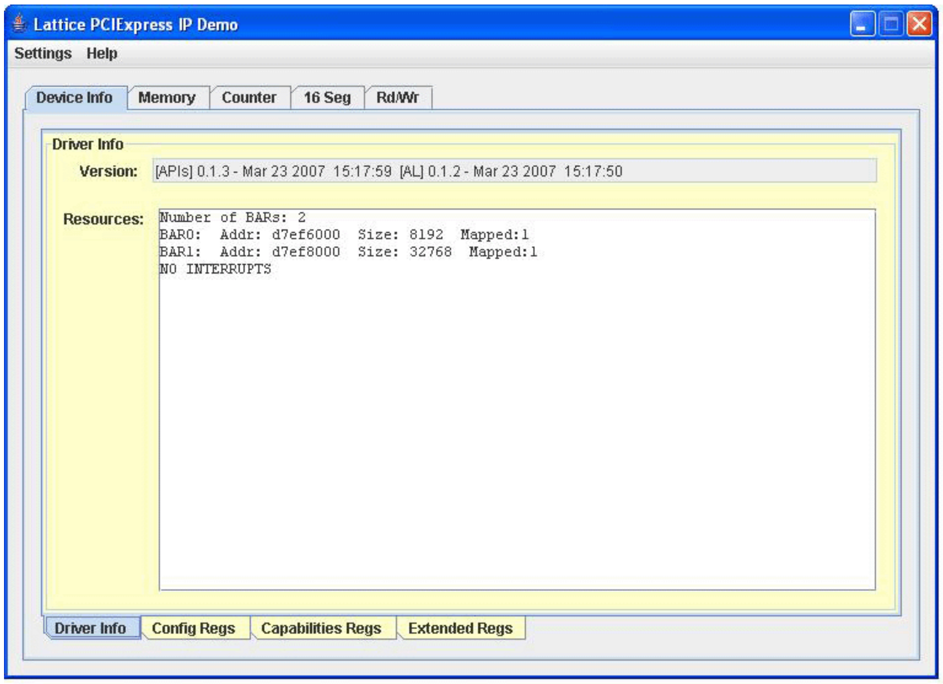
Running the Demo (Graphics Mode)

The PCI Express GUI program can be started from the Windows Start button: **Start->Programs->LatticePCIe->PCIe_GUI**. If you have a LatticeECP2M board, choose the **EC_GUI**.

Device Info Page

The opening page displays information about the device driver and the device's PCI configuration registers.

Figure 19.



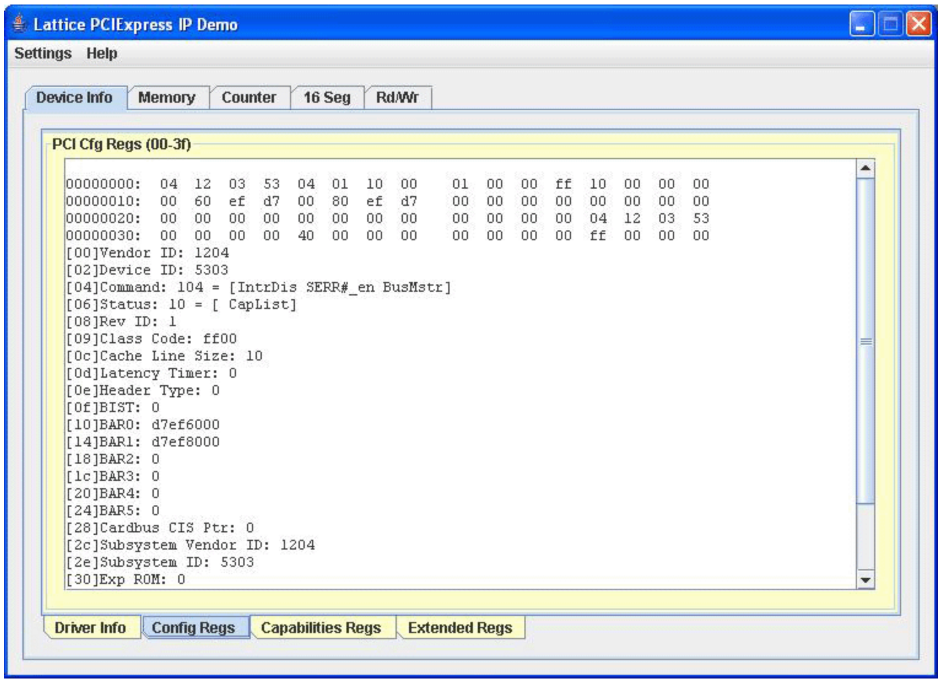
Version: [APIs] 0.1.3 - Mar 23 2007 15:17:59 [AL] 0.1.2 - Mar 23 2007 15:17:50

Resources:

```
Number of BARs: 2
BAR0:  Addr: d7ef6000  Size: 8192  Mapped:1
BAR1:  Addr: d7ef8000  Size: 32768  Mapped:1
NO INTERRUPTS
```

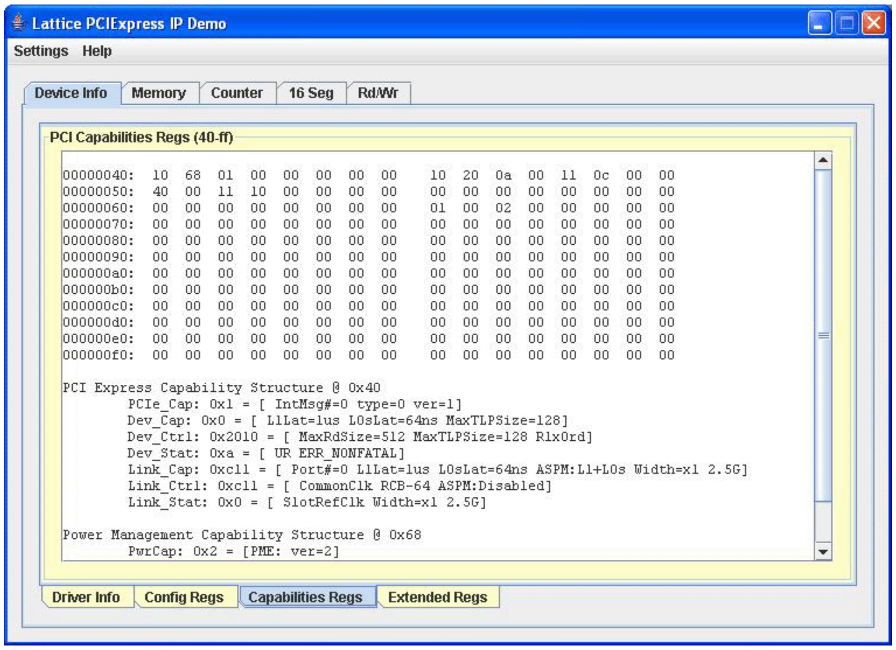
The data displayed is for informational purposes only and cannot be edited. This information is obtained from the Lattice evaluation board PCI Config space registers by the Lattice PCI Express driver when the demo is started. Displaying this page causes the driver to issue PCI Config Type 0 read requests and re-displays the register values.

The Config Reg sub-tab displays the standard PCI Config type 0 registers with each field annotated.



```
00000000:  04  12  03  53  04  01  10  00    01  00  00  ff  10  00  00  00
00000010:  00  60  ef  d7  00  80  ef  d7    00  00  00  00  00  00  00  00
00000020:  00  00  00  00  00  00  00  00    00  00  00  00  04  12  03  53
00000030:  00  00  00  00  40  00  00  00    00  00  00  00  ff  00  00  00
[00]Vendor ID: 1204
[02]Device ID: 5303
[04]Command: 104 = [IntrDis SERR#_en BusMstr]
[06]Status: 10 = [ CapList]
[08]Rev ID: 1
[09]Class Code: ff00
[0c]Cache Line Size: 10
[0d]Latency Timer: 0
[0e]Header Type: 0
[0f]BIST: 0
[10]BAR0: d7ef6000
[14]BAR1: d7ef8000
[18]BAR2: 0
[1c]BAR3: 0
[20]BAR4: 0
[24]BAR5: 0
[28]Cardbus CIS Ptr: 0
[2c]Subsystem Vendor ID: 1204
[2e]Subsystem ID: 5303
[30]Exp ROM: 0
```

The Capabilities Regs sub-tab displays the PCI Express capabilities structures that are found in register range 0x40 to 0xff. The applicable bit-fields of registers are parsed and displayed in readable format.



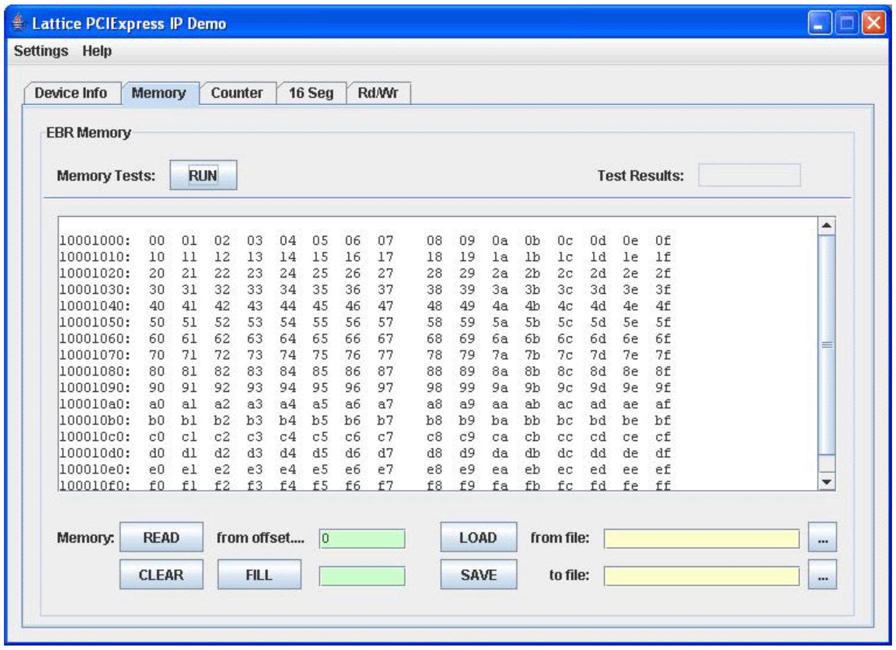
```
00000040:  10  68  01  00  00  00  00  00    10  20  0a  00  11  0c  00  00
00000050:  40  00  11  10  00  00  00  00    00  00  00  00  00  00  00  00
00000060:  00  00  00  00  00  00  00  00    01  00  02  00  00  00  00  00
00000070:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00
00000080:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00
00000090:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00
000000a0:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00
000000b0:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00
000000c0:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00
000000d0:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00
000000e0:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00
000000f0:  00  00  00  00  00  00  00  00    00  00  00  00  00  00  00  00

PCI Express Capability Structure @ 0x40
        PCIe_Cap: 0x1 = [ IntMsg#=0 type=0 ver=1]
        Dev_Cap: 0x0 = [ L1Lat=1us L0sLat=64ns MaxTLPSize=128]
        Dev_Ctrl: 0x2010 = [ MaxRdSize=512 MaxTLPSize=128 RlxOrd]
        Dev_Stat: 0xa = [ UR ERR_NONFATAL]
        Link_Cap: 0xc11 = [ Port#=0 L1Lat=1us L0sLat=64ns ASPM:L1+L0s Width=x1 2.5G]
        Link_Ctrl: 0xc11 = [ CommonClk RCB-64 ASPM:Disabled]
        Link_Stat: 0x0 = [ SlotRefClk Width=x1 2.5G]

Power Management Capability Structure @ 0x68
        PwrCap: 0x2 = [PME: ver=2]
```

Memory Page

The Memory tab has various memory access tests that can be run to show that the IP is accessible from host software via the PCI Express bus. The GUI has fields for entering data to be sent to device registers in the FPGA design. These fields are color-coded to indicate the data format they accept.

- GREEN = hex values, do not include any prefixes (0x) or suffixes (H), enter digits
- YELLOW = character string, i.e. file names, paths, a letter.
- BLUE = decimal (base 10) values.

Figure 20.



```
10001000:  00  01  02  03  04  05  06  07    08  09  0a  0b  0c  0d  0e  0f
10001010:  10  11  12  13  14  15  16  17    18  19  1a  1b  1c  1d  1e  1f
10001020:  20  21  22  23  24  25  26  27    28  29  2a  2b  2c  2d  2e  2f
10001030:  30  31  32  33  34  35  36  37    38  39  3a  3b  3c  3d  3e  3f
10001040:  40  41  42  43  44  45  46  47    48  49  4a  4b  4c  4d  4e  4f
10001050:  50  51  52  53  54  55  56  57    58  59  5a  5b  5c  5d  5e  5f
10001060:  60  61  62  63  64  65  66  67    68  69  6a  6b  6c  6d  6e  6f
10001070:  70  71  72  73  74  75  76  77    78  79  7a  7b  7c  7d  7e  7f
10001080:  80  81  82  83  84  85  86  87    88  89  8a  8b  8c  8d  8e  8f
10001090:  90  91  92  93  94  95  96  97    98  99  9a  9b  9c  9d  9e  9f
100010a0:  a0  a1  a2  a3  a4  a5  a6  a7    a8  a9  aa  ab  ac  ad  ae  af
100010b0:  b0  b1  b2  b3  b4  b5  b6  b7    b8  b9  ba  bb  bc  bd  be  bf
100010c0:  c0  c1  c2  c3  c4  c5  c6  c7    c8  c9  ca  cb  cc  cd  ce  cf
100010d0:  d0  d1  d2  d3  d4  d5  d6  d7    d8  d9  da  db  dc  dd  de  df
100010e0:  e0  e1  e2  e3  e4  e5  e6  e7    e8  e9  ea  eb  ec  ed  ee  ef
100010f0:  f0  f1  f2  f3  f4  f5  f6  f7    f8  f9  fa  fb  fc  fd  fe  ff
```

from offset.... 0

EBR Memory: The EBR Memory section tests the access to the 16kB of EBR internal to the FPGA. Accesses are done on a byte basis. All 16kB memory locations are accessed successively, testing the throughput of PCI Express and the memory interface. The following actions can be performed:

- **Pattern Tests** - Pressing **Run** starts a test to check that all locations of the EBR can be read and written and that the contents are correct. First, all 16kB are cleared to 0 and verified. Then various patterns (AA, 55, 01, FF) are written to all locations and verified. If everything passes, PASS is displayed. If a memory location has an incorrect value the test aborts and displays ERRORS! The memory contents are left with an incrementing pattern 00 01 02... that is displayed when the test successfully finishes.
- **READ** - The contents of the EBR memory are read from the value entered in the offset field. 256 bytes are read and displayed in the window above.
- **CLEAR** - Sets all 16kB to 0.
- **FILL** - Writes the byte value entered in the field to all 16kB locations.
- **LOAD** - Loads 16kB of binary data from the file specified (or as much data as is in the file) into EBR memory, starting at location 0. This can be used to load a known pattern into the EBR memory by using a file created by another tool.
- **SAVE** - Writes all 16kB of EBR memory to the file specified. This can be used to save the contents of EBR memory for off-line processing (i.e., to verify that the pattern loaded in with LOAD is correctly saved in the EBR).

Counter Page

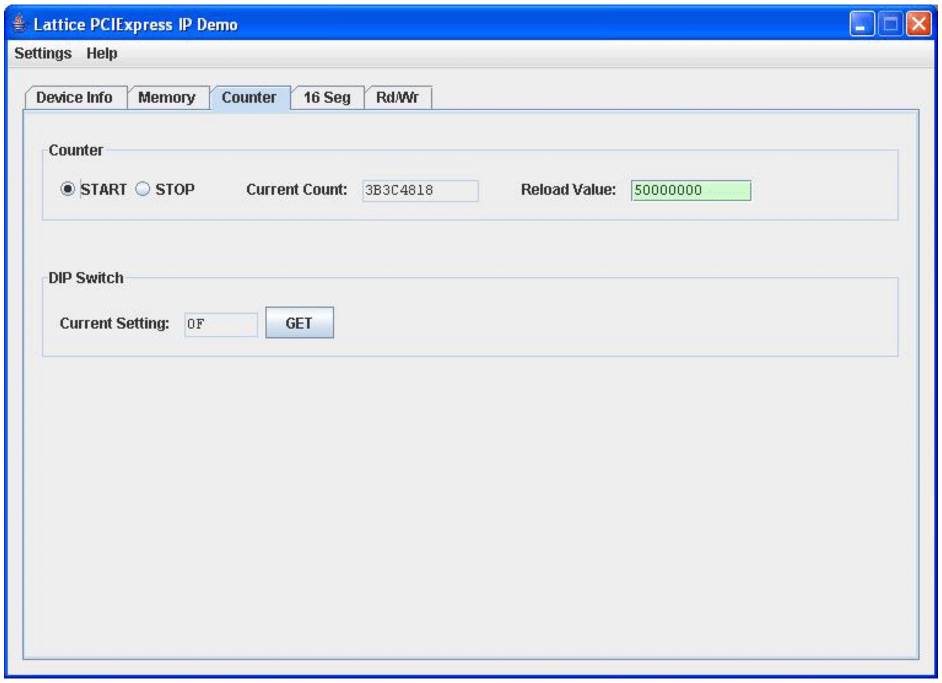
DIP Switch: The DIP switch section shows that user changes to the switches on the evaluation board are seen by the application software on the PC. The GUI polls the DIP switch register 10 times per second and displays the value read from the 8-bit DIP switch register.

The **Get** button can be used to immediately update the value. This is used if **No Polling** was selected from the **Settings** drop-down menu. Normally this is not used.

Counter: The Counter section allows you to control a 32-bit down counter in the FPGA hardware. The counter is driven by the 125 MHz clock that feeds the IP. The counter is started by selecting the **Start** radio button. Counting begins from the value entered into the **Reload Value** field. The current count value is displayed in the **Current Count** field.

The Current Count value is updated 10 times per second by the GUI polling (reading) the count register in the FPGA application IP.

Figure 21.



Counter: [✓ START] STOP — Current Count: 3B3C4818 — Reload Value: 50000000

DIP Switch: Current Setting: 0F

16-Segment Control Page
The 16-Segment Control page provides a way to interactively light segments on the display. Clicking on a segment in the image will immediately cause the corresponding segment on the LED to light. The states of the LED segments are converted to a 16-bit word value (each segment is controlled by a bit) and written to the LED control register in the GPIO portion of the IP in the FPGA. This demonstrates a memory write across the PCI Express bus.

Demo operations:

- Clicking on a segment will turn it on or off (toggles).
- Right-clicking on the background area behind the segments will clear the entire display.

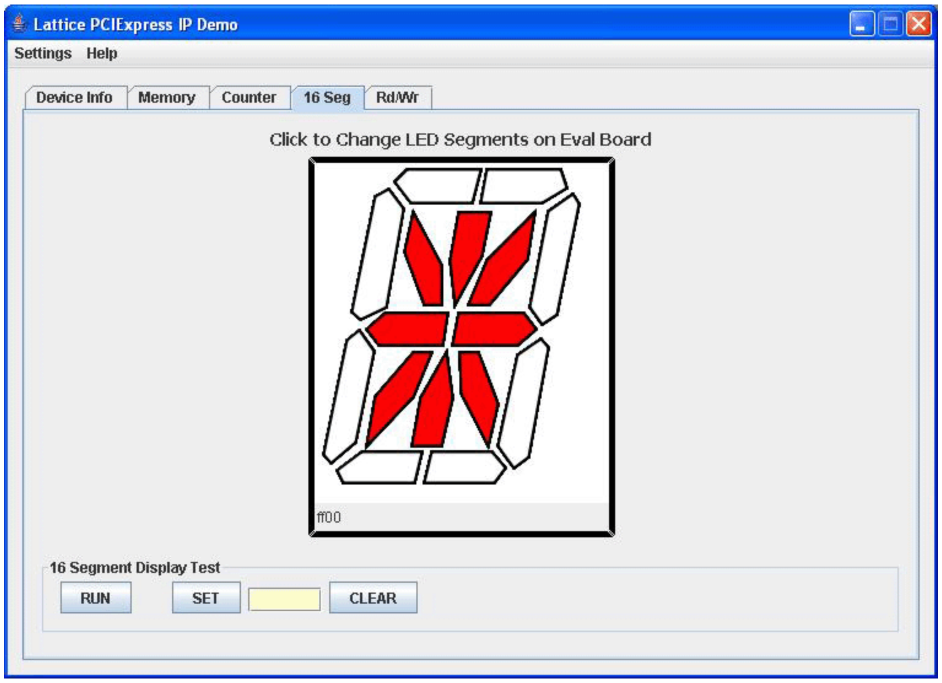
The 16-bit value written to the LED register in the FPGA is shown in the bottom left.

16-Segment Display Section: The 16-segment display has two test modes. In the first mode, a pre-set sequence of segments are lit and characters are written to the display. Pressing the **Run** button starts this operation. It will take approximately 30 seconds to complete. You must observe the 16-segment LEDs to see if it is operating correctly. The correct sequence is:

1. Light all segments, one at a time, around the perimeter.
2. Light all inner segments in a clock-wise order.
3. Turn off all inner segments in reverse order.
4. Turn off all outer segments in reverse order.
5. Write the characters "LATTICE*" one at a time to the display.
6. The "*" will be displayed when the test ends.

The second mode of operation allows a single character to be sent to the display by typing the character into the box to the right of the Set button and then pressing **Set**. Any printable ASCII character can be displayed (lower case are displayed as upper case). The CLEAR button turns off all segments in the display (you can't write a blank character using SET).

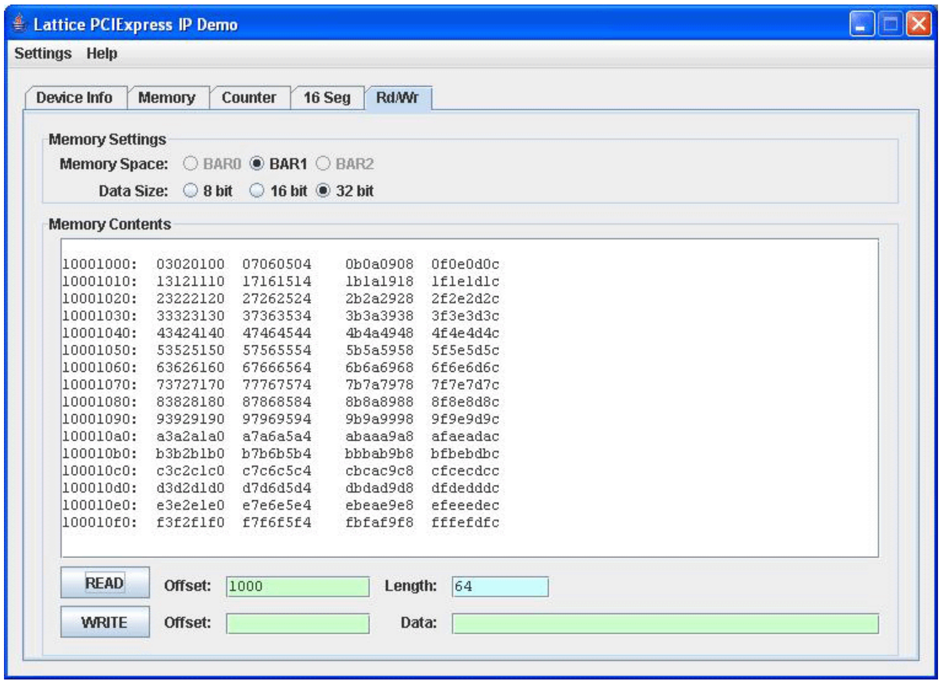
Figure 22.



Read/Write Page

The Read/Write page is used for "peeking" and "poking" registers and EBR memory values in the application IP. It is primarily used for debugging and diagnosing the application IP registers.

Figure 23.



Memory Settings

Memory Space: BAR0 [✓ BAR1] BAR2

Data Size: 8 bit 16 bit [✓ 32 bit]

Memory Contents

```
10001000:  03020100  07060504   0b0a0908  0f0e0d0c
10001010:  13121110  17161514   1b1a1918  1f1e1d1c
10001020:  23222120  27262524   2b2a2928  2f2e2d2c
10001030:  33323130  37363534   3b3a3938  3f3e3d3c
10001040:  43424140  47464544   4b4a4948  4f4e4d4c
10001050:  53525150  57565554   5b5a5958  5f5e5d5c
10001060:  63626160  67666564   6b6a6968  6f6e6d6c
10001070:  73727170  77767574   7b7a7978  7f7e7d7c
10001080:  83828180  87868584   8b8a8988  8f8e8d8c
10001090:  93929190  97969594   9b9a9998  9f9e9d9c
100010a0:  a3a2a1a0  a7a6a5a4   abaaa9a8  afaeadac
100010b0:  b3b2b1b0  b7b6b5b4   bbbab9b8  bfbebdbc
100010c0:  c3c2c1c0  c7c6c5c4   cbcac9c8  cfcecdcc
100010d0:  d3d2d1d0  d7d6d5d4   dbdad9d8  dfdedddc
100010e0:  e3e2e1e0  e7e6e5e4   ebeae9e8  efeeedec
100010f0:  f3f2f1f0  f7f6f5f4   fbfaf9f8  fffefdfc
```

Offset: 1000 Length: 64

Offset: Data:

Data accesses can be specified as byte, short or word operations by selecting the Data Size. Access is done to the selected BAR. The memory contents are displayed in the window. In the address, the upper nibble (31:28) specifies the BAR being accessed. The above example shows reading the GPIO registers (BAR 1, starting at offset 0) in the application IP and displaying them in word format.

Data can be written to registers using the WRITE button. Specify the BAR Offset to start writing at and the hex data in the Data field. Separate each value with a space. Data size should match the Data Size selected at the top of the page in Memory Settings.

Only BAR1 is accessible in the current demo.

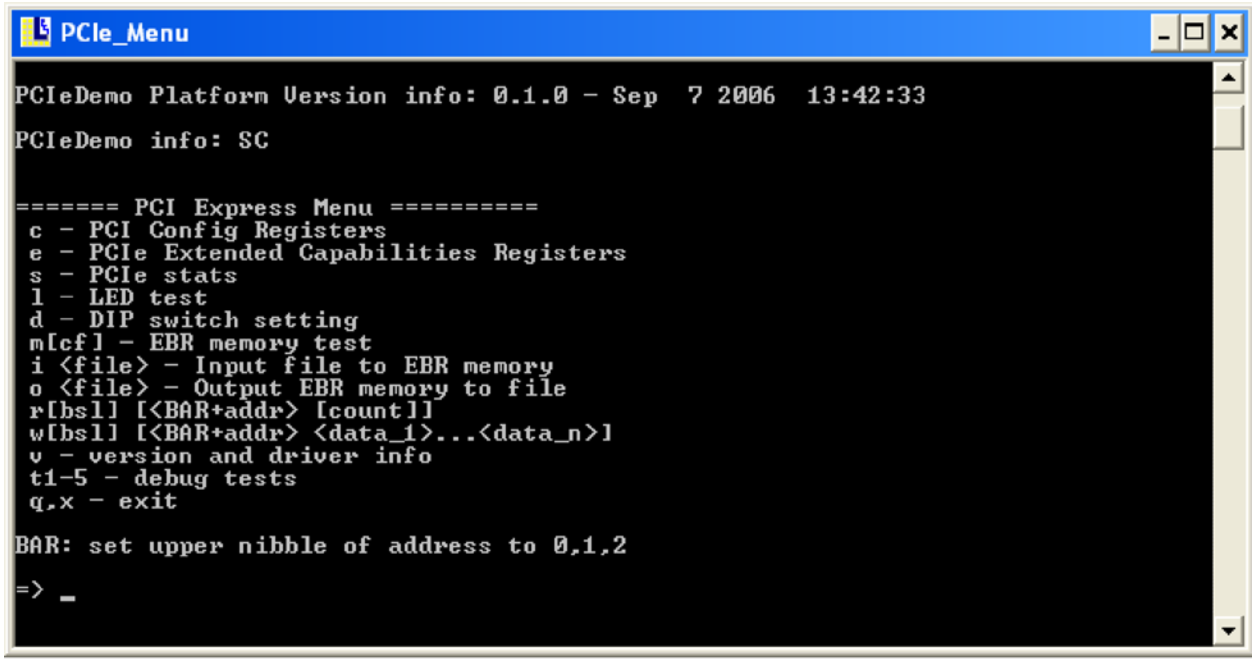
See the IP Register Memory Map section of this document for a list of valid device addresses.

Running the Demo (Menu Mode)

The text menu program can be started from the Windows Start bar. **Start->Programs->LatticePCIe->PCIe_Menu**. If you have a LatticeECP2M board, choose the **EC_Menu**. The program executable and batch file to start it are located in the directory where the demo package is installed (**C:\Program Files\LatticeApps\PCIe\Demo**). Double-click on the **menu.bat** or **EC_menu.bat** (depending on the type of evaluation board installed).

Note: The text menu screen shots that follow may not reflect the most recent version of the software, but the functionality remains the same.

Figure 24.



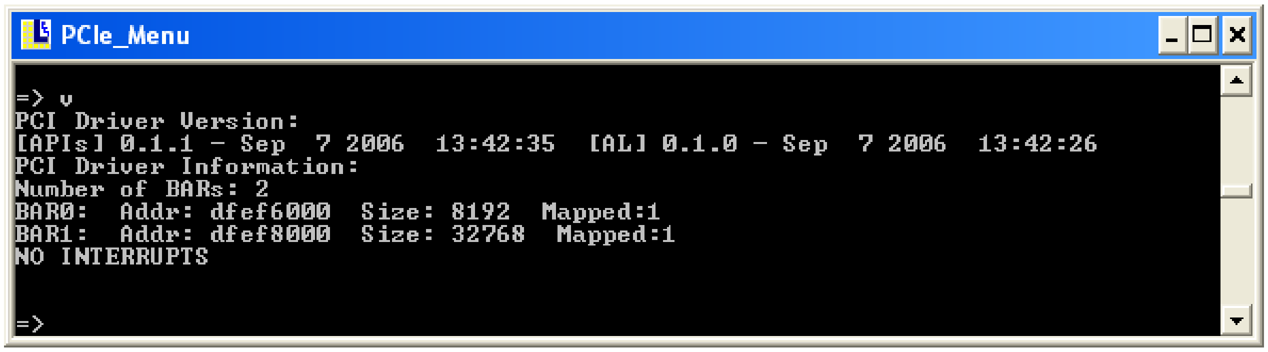
```
PCIeDemo Platform Version info: 0.1.0 - Sep  7 2006  13:42:33

PCIeDemo info: SC


======= PCI Express Menu ==========
 c - PCI Config Registers
 e - PCIe Extended Capabilities Registers
 s - PCIe stats
 l - LED test
 d - DIP switch setting
 m[cf] - EBR memory test
 i <file> - Input file to EBR memory
 o <file> - Output EBR memory to file
 r[bsl] [<BAR+addr> [count]]
 w[bsl] [<BAR+addr> <data_1>...<data_n>]
 v - version and driver info
 t1-5 - debug tests
 q,x - exit

BAR: set upper nibble of address to 0,1,2

=> _
```

The top lines display information about the board and the driver installation. Most important is the "PCIeDemo info:" line that says "SC" or "EC" to indicate the demo and drivers are communicating with the evaluation board hardware.

The lower half of the screen shows the various menu options available. Type the letter or command string and press **<Enter>** to execute. "Q" or "X" exits back to the command prompt.

Type **V** to display details about the driver resources.

Figure 25.

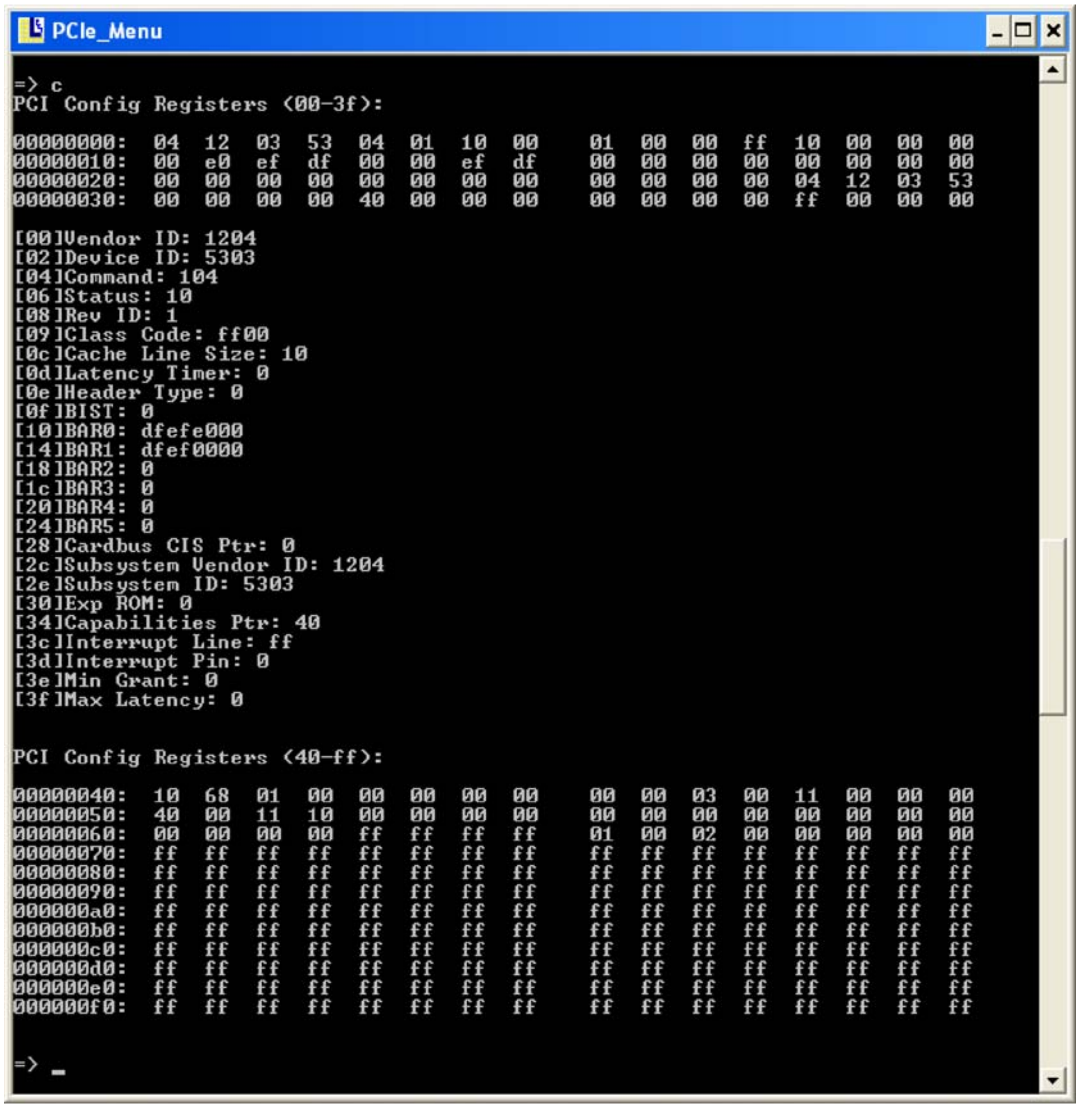


```
=> v
PCI Driver Version:
[APIs] 0.1.1 - Sep  7 2006  13:42:35  [AL] 0.1.0 - Sep  7 2006  13:42:26
PCI Driver Information:
Number of BARs: 2
BAR0:  Addr: dfef6000  Size: 8192  Mapped:1
BAR1:  Addr: dfef8000  Size: 32768  Mapped:1
NO INTERRUPTS


=>
```

This indicates that the evaluation board hardware is recognized as a PCI device by the PC and Windows and is assigned address spaces corresponding to the BARs programmed into the FPGA IP.

Type **C** to display the PCI Config Type 0 registers, as read by the driver during start-up.

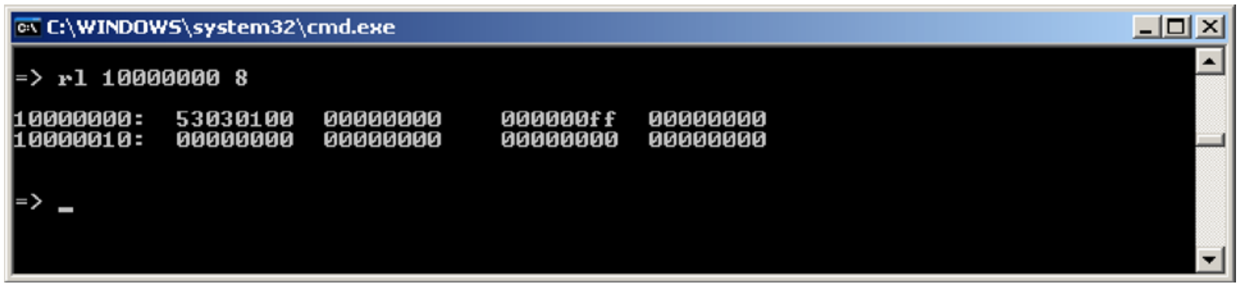
Figure 26.



```
=> c
PCI Config Registers (00-3f):

00000000:  04  12  03  53  04  01  10  00    01  00  00  ff  10  00  00  00
00000010:  00  e0  ef  df  00  00  ef  df    00  00  00  00  00  00  00  00
00000020:  00  00  00  00  00  00  00  00    00  00  00  00  04  12  03  53
00000030:  00  00  00  00  40  00  00  00    00  00  00  00  ff  00  00  00

[00]Vendor ID: 1204
[02]Device ID: 5303
[04]Command: 104
[06]Status: 10
[08]Rev ID: 1
[09]Class Code: ff00
[0c]Cache Line Size: 10
[0d]Latency Timer: 0
[0e]Header Type: 0
[0f]BIST: 0
[10]BAR0: dfefe000
[14]BAR1: dfef0000
[18]BAR2: 0
[1c]BAR3: 0
[20]BAR4: 0
[24]BAR5: 0
[28]Cardbus CIS Ptr: 0
[2c]Subsystem Vendor ID: 1204
[2e]Subsystem ID: 5303
[30]Exp ROM: 0
[34]Capabilities Ptr: 40
[3c]Interrupt Line: ff
[3d]Interrupt Pin: 0
[3e]Min Grant: 0
[3f]Max Latency: 0


PCI Config Registers (40-ff):

00000040:  10  68  01  00  00  00  00  00    00  00  03  00  11  00  00  00
00000050:  40  00  11  10  00  00  00  00    00  00  00  00  00  00  00  00
00000060:  00  00  00  00  ff  ff  ff  ff    01  00  02  00  00  00  00  00
00000070:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff
00000080:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff
00000090:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff
000000a0:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff
000000b0:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff
000000c0:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff
000000d0:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff
000000e0:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff
000000f0:  ff  ff  ff  ff  ff  ff  ff  ff    ff  ff  ff  ff  ff  ff  ff  ff


=> _
```

Use the "r" and "w" commands to read and write registers in the user space of the design Consult the memory map of the demo IP for register addresses. The following command reads back all GPIO registers in BAR 1:

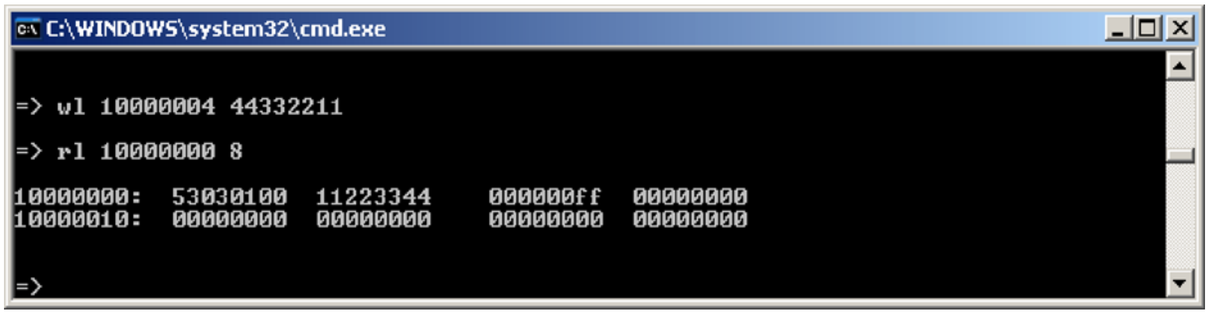
Figure 27.



```
C:\WINDOWS\system32\cmd.exe

=> rl 10000000 8

10000000:  53030100  00000000    000000ff  00000000
10000010:  00000000  00000000    00000000  00000000

=> _
```

The "r" and "w" commands can specify 8, 16, or 32 bit accesses using "rb", "rs" and "rl" respectively. The BAR to read from is indicated in the most significant nibble. In the above command, the "1" in 10000000 indicates access offset 0 (the remaining 0000000) in BAR 1. This command format requires typing all eight digits of the 32-bit address. The command "rl 0" reads from BAR0 offset 0. This command sequence shows writing something into the Scratchpad register and then reading it back.

Figure 28.

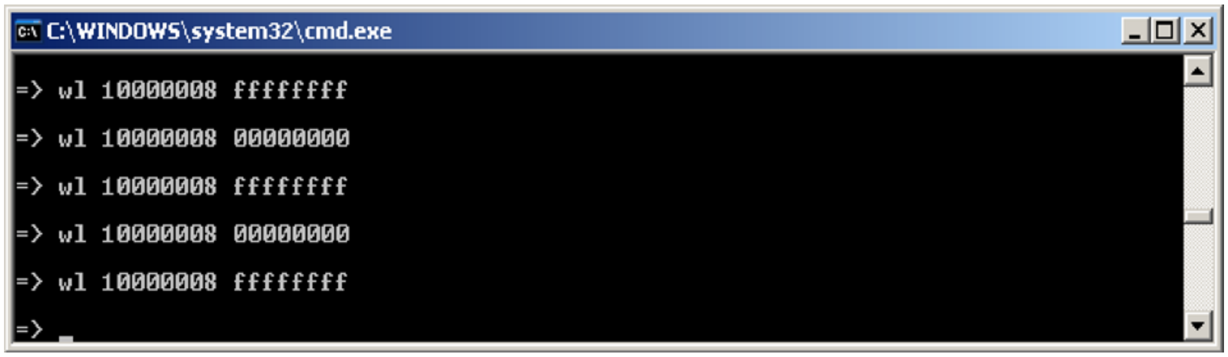


```
C:\WINDOWS\system32\cmd.exe

=> wl 10000004 44332211

=> rl 10000000 8

10000000:  53030100  11223344    000000ff  00000000
10000010:  00000000  00000000    00000000  00000000

=>
```

Note: Current issues with byte lane decoding require a word (32-bit) access to the LED register.

The following command sequence will toggle the 16-segment LEDs on and off:

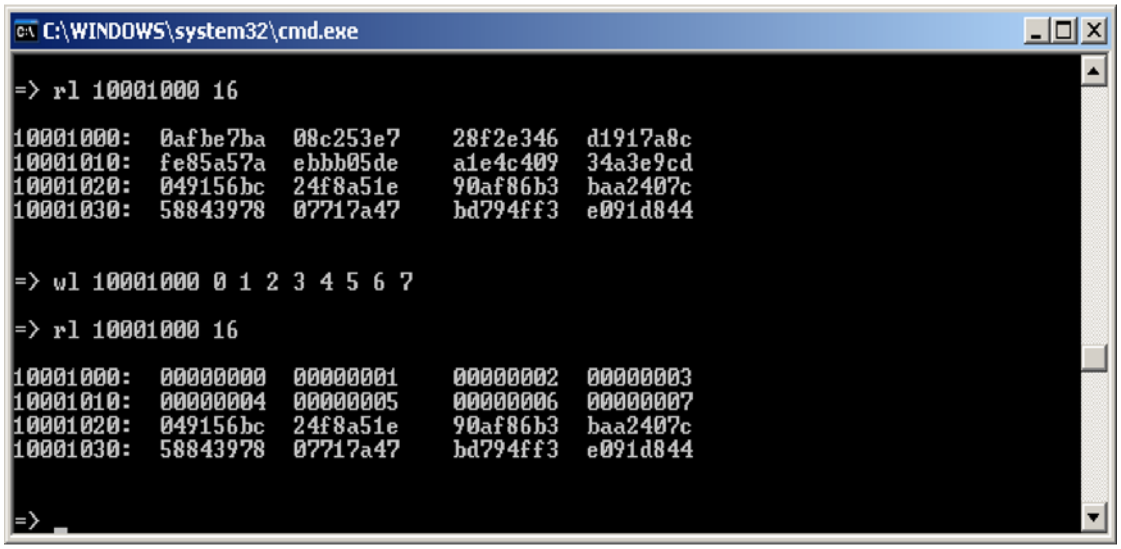
Figure 29.



```
C:\WINDOWS\system32\cmd.exe

=> wl 10000008 ffffffff

=> wl 10000008 00000000

=> wl 10000008 ffffffff

=> wl 10000008 00000000

=> wl 10000008 ffffffff

=> _
```

The following commands show access to the EBR memory, which is located at offset 0x1000 in BAR1. The first command displays the current, uninitialized contents. The second command writes 32-bit words into memory and the third command displays the modified memory locations.

Figure 30.



```
=> rl 10001000 16

10001000:  0afbe7ba  08c253e7    28f2e346  d1917a8c
10001010:  fe85a57a  ebbb05de    a1e4c409  34a3e9cd
10001020:  049156bc  24f8a51e    90af86b3  baa2407c
10001030:  58843978  07717a47    bd794ff3  e091d844


=> wl 10001000 0 1 2 3 4 5 6 7

=> rl 10001000 16

10001000:  00000000  00000001    00000002  00000003
10001010:  00000004  00000005    00000006  00000007
10001020:  049156bc  24f8a51e    90af86b3  baa2407c
10001030:  58843978  07717a47    bd794ff3  e091d844


=> _
```

Demo Design Details

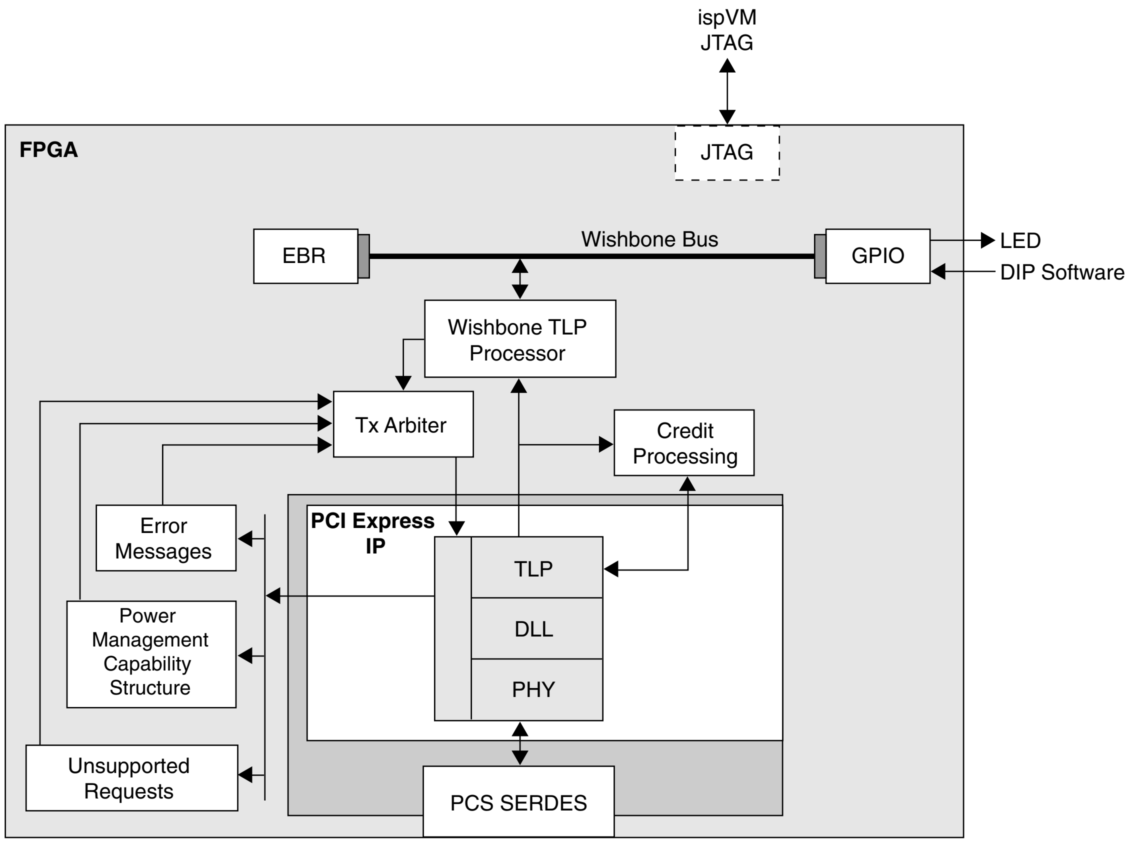
This section provides technical details of the demo system design to give users a better understanding of how the PCI Express IP core is implemented and demonstrated. The supporting IP around the PCI Express core is described in detail. The software application and driver design are also discussed.

IP Components

The following block diagram shows a high level view of the IP blocks within the Lattice FPGA

The demo design is comprised of application IP and PCI Express Core IP, as shown in Figure 31. The purpose of each block is explained below.

Figure 31. Demo Design Block Diagram



IP Interconnection

The IP modules are interconnected using the Wishbone bus. The Wishbone bus is an open standard that defines a microprocessor bus suitable for use in FPGAs. The Wishbone bus specification can be found at www.opencores.org. The Wishbone is used to transfer data (reads/writes) between the top of the PCI Express stack and the register and memory devices in the user IP. All device registers appear as memory locations that can be read and written by the PC CPU in PCI address space. The Wishbone bus is a 32-bit address and data bus with byte enables to select the proper byte(s) when doing 8, 16 or 32-bit accesses.

PCI Express IP Details

Details of the PCI Express IP core can be found in the *LatticeSCM PCI Express x1, x4 IP Core User's Guide* and the *LatticeECP2M PCI Express x1 Endpoint IP Core User's Guide* located on the Lattice web site at www.latticesemi.com.

The PCI Express IP core is the "black-box" IP that a user will get after generating a PCI Express module in IPexpress™.

PCI Express Core: The PCI Express core contains the PHY, Data Link Layer and Transaction Layer. The SERDES and PCS hardware is also included in this block. All initial register settings are programmed via the bitstream so the stack is operational at power-up.

PCI Configuration Registers: The PCI configuration registers are tied to the core since they are required for bus operation, PCI enumeration and configuration cycles. These registers are configured in the user's HDL design and loaded by the bitstream so that their values are present at power-up.

EBR Memory IP

The demo uses 16kB of EBR memory for storage and to demonstrate large data transfers. The memories start at address 0x1000 on the Wishbone system bus, which corresponds to the BAR 1 offset 0x1000 as seen by the PC CPU. They support 8, 16 and 32-bit accesses.

GPIO Register IP

The GPIO IP implementation provides access to the LEDs and DIP switches on the evaluation board. It is a simple module with registers that are written and read from the Wishbone bus. Writes to the LED register bits will cause LEDs to turn on (1) or turn off (0). Reads from these registers will return their current settings. Reads of the DIP switch register will return its current switch settings. The GPIO block also contains a version register, scratch pad register and 32-bit counter circuit. These are used for various types of access validation.

BARs / Address Spaces

The target address space is divided into different address ranges for the purpose of demonstrating the core's ability to support multiple BARs and to demonstrate how application IP will handle the accesses to different width devices.

Table 3. BAR Summary

BAR	Size	Access Width	Devices
BAR0	8kB	8,16,32 bit	FUTURE: PCI Express Core Configuration and Status Registers, PCS/SERDES
BAR1	20kB	8,16,32 bit	GPIO Registers, Interrupt Control, EBR Memory, DMA Control
BAR2	???	32 bit	FUTURE: FAST Path to External RAM

Byte Ordering

The PCI Express board is intended to be installed in a PC motherboard. Based on Intel chips, PCs are little endian. The PCI Express configuration registers are also in little endian format. Therefore, all multi-byte registers in the design are mapped to the Wishbone bus in little endian format so they are in the native byte order of the application software running on the PC.

Bit ordering is also defined to eliminate any confusion. The standard practice of having LSB = bit 0 and MSB = bit 7/15/31 is followed. Byte ordering for multi-byte values is little endian, as shown in Table 4.

Table 4. Byte Ordering for Multi-Byte Values

Address (Hex)	Byte Location			
	Byte 3 [31:24]	Byte 2 [23:16]	Byte 1 [15:8]	Byte 0 [7:0]
0000				▨
0001			▨	
0002		▨		
0003	▨			
0004			MSB -------------------------- LSB	
0006	MSB -------------------------- LSB			
0008	MSB -- LSB			

Memory Maps

PCI Express Configuration Register Map

The PCI Express Configuration Register Memory Map includes the standard PCI TYPE 0 configuration registers that must be present, as well as the Capabilities registers and the PCI Express Extended Capabilities registers.

Table 5. PCI Express Configuration Register Map

CFG	Location			
Offset (Hex)	**Byte 3 [31:24]**	**Byte 2 [23:16]**	**Byte 1[15:8]**	**Byte 0 [7:0]**
00	Device ID		Vendor ID	
04	Status		Command	
08	Class Code			Revision ID
0C	BIST	Header Type	Latency Timer	Cache Line Size
10	BAR 0			
14	BAR 1			
18	BAR 2			
1C	BAR 3			
20	BAR 4			
24	BAR 5			
28	Cardbus CIS Pointer			
2C	Subsystem ID		Subsystem Vendor ID	
30	Expansion ROM Base Address			
34	Reserved			Capabilities Pointer
38	Reserved			
3C	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line
40 ... FF	New Capabilities Registers (application specific)			
100 ... FFF	Extended Capabilities Registers (application specific)			

- Vendor ID = 0x1204 (Lattice Semiconductor)
- Device ID = 0x5303 = SC board, 0xEC02 = ECP2M board
- Revision ID = 0x01 (currently, increments with every design change)
- Class Code = 0xff0000 (other device code)
- BAR0 = 0xffffe008
- BAR1 = 0xffff8008
- BAR2 = TBD - not used at this time
- Subsystem Vendor ID = 0x1204 (Lattice Semiconductor)
- Subsystem ID = 0x5303 or 0xEC02
- Capabilities Pointer = 0x40, points to PCIe Capabilities Structure

PCI Express Capabilities Structure Register Map

The PCI Express Capabilities Structure must contain valid values for the board to be identified as a PCI Express compliant device. The values in Table 6 have been chosen for the purposes of the Demo IP:

Table 6. Capabilities Structure Register Map

CFG	Location			
Offset (Hex)	Byte 3 [31:24]	Byte 2 [23:16]	Byte 1[15:8]	Byte 0 [7:0]
40	00	01	68	10
44	00	00	00	00
48	00	00	00	00
4C	00	00	00	11
50	10	11	00	40
54	00	00	00	00
58	00	00	00	00
5C	00	00	00	00
60	00	00	00	00

The PCI Express Capabilities structure is filled in with the following parameters:

Table 7. PCI Express Capabilities Parameters

Offset	Name	Value	Description
0	ID	0x10	PCI Express Capabilities ID
1	Next	0x68	Points to power management structure
2	Dev Cap	0x00000000	Version 1, endpoint
4	Dev Ctrl	0x0000	Default values of all 0's
8	Dev Status	0x0000	(valid at run-time)
0A	Link Cap.	0x000000011	2.5 GB/s, x1 link width
0C	Link Ctrl	0x0040	Common reference clock
10	Link Status	0x0000	(valid at run-time)

PCI Express Power Management Structure Register Map

The PCI Express Power Management Structure must contain valid values in order for the board to be identified as a PCI Express compliant device. The values in Table 8 have been chosen for the purposes of the Demo IP.

Table 8. Power Management Structure Register Map

CFG	Location			
Offset (Hex)	Byte 3 [31:24]	Byte 2 [23:16]	Byte 1[15:8]	Byte 0 [7:0]
68	00	02	00	01
6C	00	00	00	00

The PCI Express Power Management structure is filled in with the parameters listed in Table 9.

Table 9. Power Management Parameters

Offset	Name	Value	Description
0	ID	0x01	PCI Express Power Management ID
1	Next	0x00	Last structure in list
2	Pwr Mng Cap	0x0002	Version 1.1, no power management supported
4	Control/Status	0x0000	Default values of all 0's, no power management
6	Brdg Support	0x00	Not supported
7	Data Reg	0x00	Not supported

GPIO Register Map

The General Purpose I/O (GPIO) block contains registers that are used by the demo to validate accesses from the host CPU. They provide visual indicators and manual inputs to verify PCI Express accesses.

Table 10. GPIO Register Map

BAR1	Location			
Offset (Hex)	Byte 3 [31:24]	Byte 2 [23:16]	Byte 1[15:8]	Byte 0 [7:0]
0000	ID Register			
0004	Scratch Pad Register			
0008	DIP Switch		16-Segment LED	
			Upper 8 Bits	Lower 8 Bits
000C	Reserved			Counter Control
0010	Current Counter Value			
0014	Counter Reload Value			

- Demo ID Register (0x00) - 32-bit pre-programmed, read-only value: 0x53030100 [DeviceID][HdwRev][AppID]

Table 11. Demo ID Register

MSB [31:24]	[23:16]	[15:8]	LSB [7:0]
53	03	01	00

- Scratch Pad Register (0x04) - 32-bit register that can be read or written with any user value to verify reads/writes. Byte, short or long access is supported.
- 16-Segment LED Register (0x08)
  - 16-bit register, writing a 1 to a bit lights the LED segment, writing a 0 turns it off
  - LED segment A wired to LSB (d0) and segment U wired to MSB (d15)

Table 12. 16-Segment LED Register

Register Bit	Segment	Pin
D0 (LSB)	A	AC33
D1	B	AA30
D2	C	AD34
D3	D	AA28
D4	E	AA33
D5	F	AB34
D6	G	AA29
D7	H	Y31
D8	K	Y32
D9	M	W24
D10	N	W33
D11	P	Y34
D12	R	W26
D13	S	V34
D14	T	W25
D15 (MSB)	U	U33

Note: The decimal point is used by the IP to indicate PCI Express activity and is not available for software control.

- DIP Switch Register (0x0b)
  - Read Only. Each bit indicates the position of a corresponding switch. 1= up, 0 = down.
  - Switch2 is the upper nibble, Switch 1 is lower nibble.
  - Ordering is done so MSB is on the left, LSB is on the right

Table 13. DIP Switch Register Map

Register Bit	Switch	NetName	Pin
D0 (LSB)	SW1:4	Switch5	F20
D1	SW1:3	Switch6	J22
D2	SW1:2	Switch7	H22
D3	SW1:1	Switch8	B19
D4	SW2:4	Switch1	A20
D5	SW2:3	Switch2	K22
D6	SW2:2	Switch3	K21
D7 (MSB)	SW2:1	Switch4	G20

- Counter Control (0x0c) - Bits to control operating mode of the 32-bit counter register.

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reload	Run

  - Bit 0 = Run: 0 = Stop, 1 = Counter runs
  - Bit 1 = Reload: 0 = Halt at 0, 1 = Reload with reload register and continue counting

- Current Counter Value (0x10) - 32-bit current down-count value 31

31		0
MSB	32-bit current count	LSB

- Counter Reload Value (0x14) - 32-bit value to reload into counter when count reaches 0

31		0
MSB	32-bit reload value	LSB

EBR Register Map

The EBR is mapped as a 32-bit wide device with 4096 addresses. The total size is 16kB (4096 x 32 bit). It can be accessed with byte, short or word accesses.

Table 14. EBR Register Map

BAR1	Location			
Offset (Hex)	**Byte 3 [31:24]**	**Byte 2 [23:16]**	**Byte 1[15:8]**	**Byte 0 [7:0]**
1000				
...				
...				
4ffc				

Error Messages

The Error Messages module is provided to generate error messages that are required for PCI Express compliance. This module receives error indications for fatal, non-fatal, and correctable errors from the PCI Express core and in turn generates the proper Message TLP.

Unsupported Requests

The Unsupported Requests module is provided to generate unsupported request completions that are required for PCI Express compliance. This module will monitor TLPs from the PCI Express core and produce unsupported request and completer abort completion TLPs.

Tx Arbiter

The Tx Arbiter module is used to arbitrate between several modules that require ownership of the PCI Express core transmit interface. This module employs a simple round robin scheduler between the Error Messages, Power Management, Unsupported Requests, and the Wishbone TLP Processor.

Credit Processing

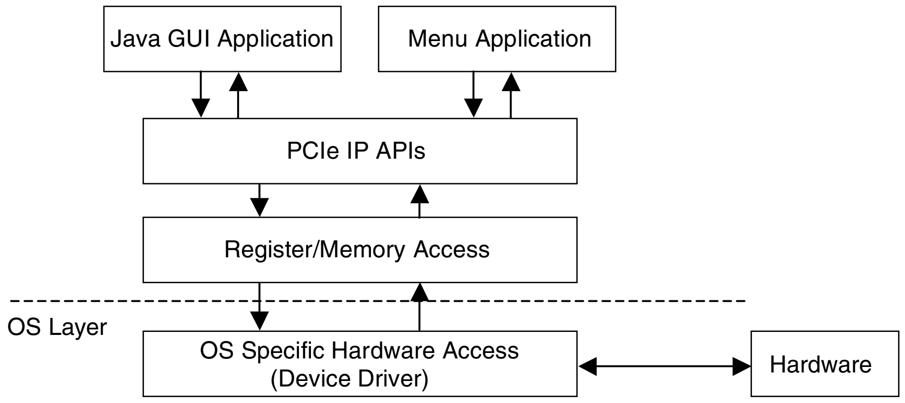
The Credit Processing module is a simple module to free credits used by the root complex. As TLPs are received by the demo design the available credits from the root complex point of view decreases. This module provides the proper credit signals to the PCI Express core to free up these credits. It is a simple implementation since all TLPs are simply terminated and it does not depend on how the TLP is actually used.

The demo does not support transmit credit processing. As an endpoint demo that is not an initiator on the bus, only Completion TLPs will be sent. Completion TLPs are a response to a memory request from the root complex. The root complex will not ask for more data if its buffers are full so available credit of the root complex is simply not checked before sending the completion TLP.

Demo Software Components

The demo software consists of the GUI (or menu) executable and a Windows WDM device driver. The components are divided into the following hierarchical layers.

Figure 32. Demo Software Components



OS Specific Hardware Access (Windows WDM Driver)

This portion of the software provides the OS driver to allow application software to "open" a device and gain access to the evaluation board. This code is Windows and Lattice specific in that it looks for the PCI Device ID, Vendor ID, etc. to know that it is talking to the PCI Express evaluation board. The driver is a Windows WDM kernel driver. It supports all the plug-n-play Windows messages, but most do nothing. Power management is not implemented, nor are interrupts. The driver's main purpose is to be associated with a hardware device (evaluation board) when Windows Plug-n-Play Manager scans the PCI/PCI Express buses. Once Windows loads the driver, the driver asks Windows for the device's resources (how many BARs, sizes, interrupts, etc.) and maps the BARs into memory space the driver code can access. The driver's main purpose is to provide an interface to allow user applications to read or write data to a hardware device. The user application opens the device driver and uses standard Windows OS system calls (read, write.ioctl) to read or write data (bytes, ints, longs) to an address on the evaluation board via a BAR space. The driver does not process any information to/from the evaluation board asynchronously. It is invoked by the user space code to perform a read/write and the results are returned back to the user code.

Register/Memory Access
This portion of the software provides an OS-independent set of functions to read and write to a device. The upper layers of the application code can then be designed to be OS independent and can be built to run on Windows or Linux. This layer is similar to Windows' HAL.

PCI Express IP APIs
These API functions provide simplified access (helper functions) to the standard PCI configuration registers and the Lattice PCI Express demo IP specific registers. The API's access:

1. PCI configuration registers - BARs, device ID
2. Extended capabilities registers
3. Demo GPIO registers - scratchpad, LEDs, switches, EBR
4. Driver version information

Application Software
The application software is the demo or test code that is used to demonstrate PCI Express operations with a Lattice FPGA. The demo software is either a Java GUI or a simple console-based text menu. Both applications use the APIs provided to access the PCI Express driver to do the following:

1. Perform memory read/writes to access the LEDs and switches to show real-time control
2. Stress test the PCI Express interface via high-throughput accesses to the EBR memory and memory tests to verify contents are correct.

The Java GUI application uses JNI methods to invoke functions in a DLL that in turn call the APIs. The JNI methods allow the C code to be invoked by Java. The advantage is that most of the elementary demo operations are handled by the API library which is shared by the text menu and the GUI. Functional changes need to be made in only one place, and the user can be assured that the operations in the GUI and menu are equivalent.

Troubleshooting

This section outlines some debug procedures to follow when experiencing trouble installing or running the demo.

Trouble Installing the Demo Software Package

The most likely issue that may arise is the issue of permissions when installing. Depending on the system security policies, the user my need to have administrator permission to install into the Program Files directory (the default location).

Trouble with the Board

Ensure the board is installed in a PCI Express slot. It can physically fit into a PCI slot. This can damage the board or PC if power is applied when its the wrong type of slot.

Ensure the board has a valid PCI Express bitstream loaded in the SPI flash and for LatticeECP2M that the Mode DIP switches are set to program from SPI flash (does not apply to LatticeSC evaluation boards).

Ensure the four Status LEDs are on, indicating the board is seen as a PCI Express endpoint. If the two yellow LEDs and two green LEDs are not on, the board will not be recognized by the PC BIOS or Windows. You can try installing in a different PCI Express slot to see if that fixes the link-up problem. You can also try pressing the evaluation board's reset button immediately after the PC boots.

Ensure the board is seen by Windows. Check **My Computer->Properties->Hardware->Device Manager** and verify the LSC_PCIe driver and evaluation board are shown in the list. If not, shut down the system and try another slot. If the board is present, check its Properties and the Resource tab to verify memory was assigned to it. Also verify the Vendor ID and Device ID are valid, as seen be Windows Plug-n-Play. If the values are invalid, the bitstream may be corrupt and may need to be reloaded into SPI flash.

Trouble with the Driver

The evaluation board must be installed in the PC, and seen by Windows, for the driver to be installed. If you do not see the "Found New Hardware" message when logging in after installing the board, check the board LEDs. Try a different PCI Express slot.

Make sure you specify the search location for the driver during installation. Tell Windows to install from the **Demo\Drivers\EvalBoard** directory.

You will need administrator permission to install device driver files.

Trouble Running the Demo

The evaluation board must be installed in the PC, and seen by Windows, for the driver to be installed/loaded. The driver must be loaded by Windows in order to run the demo. Again, verify that Windows sees the board and has loaded a driver for it.

If the GUI displays an error (ERROR LOADING LIBRARY:Cpp_Jni - running in View Only mode) when launched, then the driver may not be found or loaded. There are two causes:

1. The driver was never loaded (or evaluation board is not installed)
2. The board failed to be detected by Windows.

Either way, the board needs to be installed and seen by Windows and the LSC_PCIe driver needs to be associated with the hardware.

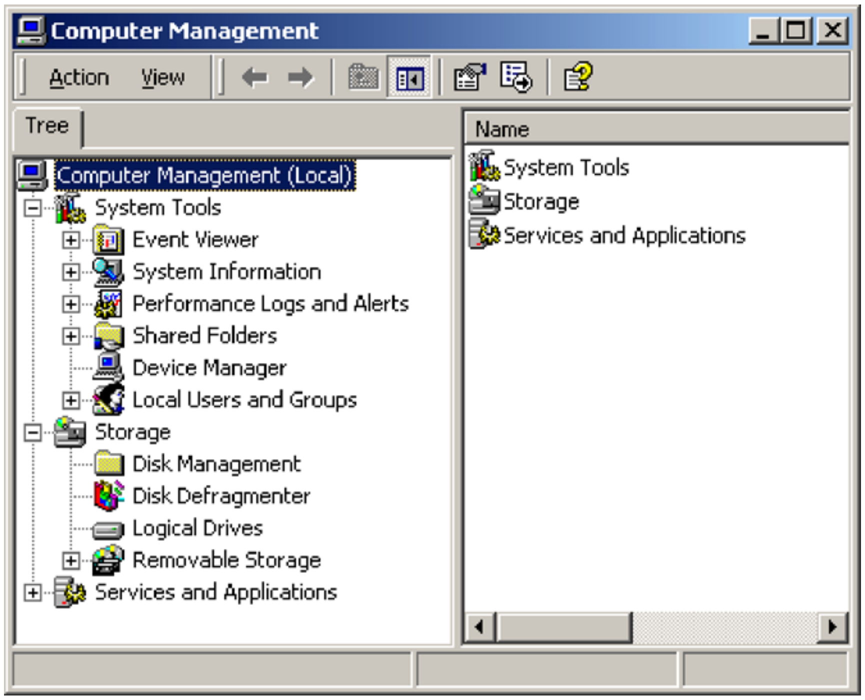
Windows Debug Tools

Windows offers some utilities to interrogate the operation of hardware devices. Use these tools to verify the evaluation board device driver is loaded and running.

Computer Management

Start->Settings->Control Panel->Administrative Tools->Computer Management

Figure 33.



Computer Management (Local)
- System Tools
  - Event Viewer
  - System Information
  - Performance Logs and Alerts
  - Shared Folders
  - Device Manager
  - Local Users and Groups
- Storage
  - Disk Management
  - Disk Defragmenter
  - Logical Drives
  - Removable Storage
- Services and Applications

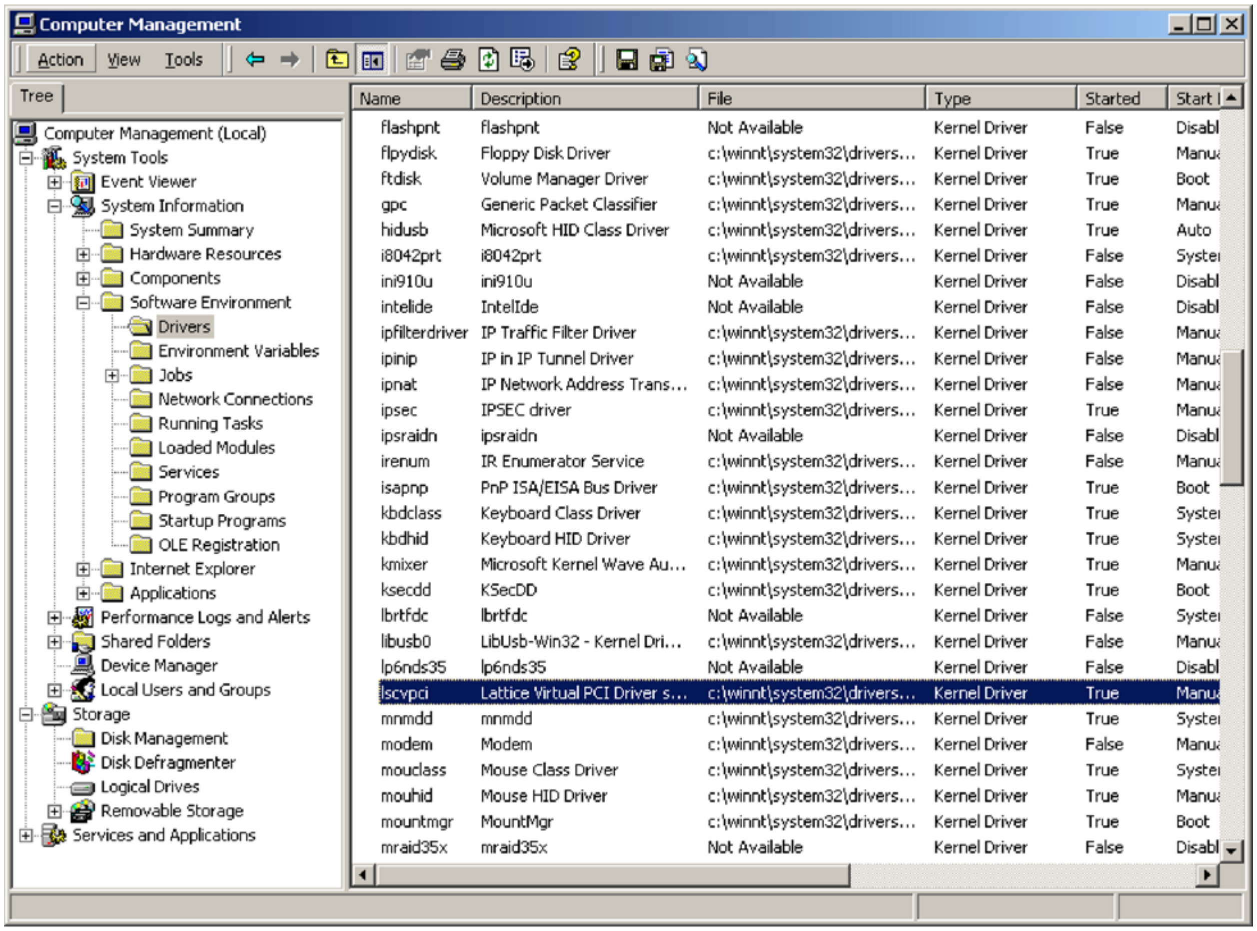
System Tools
Storage
Services and Applications

Event Viewer displays messages from system device drivers, including the PCI Express Driver. Status messages about booting, run-time configuration and error messages are sent to the System Log.

The Device Manager shows the hardware that is identified and controlled by drivers. Of particular interest are the "Other Devices" category which show the Lattice PCI Express evaluation board and the VirtualPCIe board.

The **System Information -> Software Environment -> Drivers** shows the software drivers that are loaded and their status (started, etc.). Look for the **lscvpci** and/or **lscpcie** driver names which indicate the board simulator or evaluation board driver has been loaded.

Figure 34.



Name	Description	File	Type	Started	Start I
flashpnt	flashpnt	Not Available	Kernel Driver	False	Disabl
flpydisk	Floppy Disk Driver	c:\winnt\system32\drivers...	Kernel Driver	True	Manu
ftdisk	Volume Manager Driver	c:\winnt\system32\drivers...	Kernel Driver	True	Boot
gpc	Generic Packet Classifier	c:\winnt\system32\drivers...	Kernel Driver	True	Manu
hidusb	Microsoft HID Class Driver	c:\winnt\system32\drivers...	Kernel Driver	True	Auto
i8042prt	i8042prt	c:\winnt\system32\drivers...	Kernel Driver	False	Syste
ini910u	ini910u	Not Available	Kernel Driver	False	Disabl
intelide	IntelIde	Not Available	Kernel Driver	False	Disabl
ipfilterdriver	IP Traffic Filter Driver	c:\winnt\system32\drivers...	Kernel Driver	False	Manu
ipinip	IP in IP Tunnel Driver	c:\winnt\system32\drivers...	Kernel Driver	False	Manu
ipnat	IP Network Address Trans...	c:\winnt\system32\drivers...	Kernel Driver	False	Manu
ipsec	IPSEC driver	c:\winnt\system32\drivers...	Kernel Driver	True	Manu
ipsraidn	ipsraidn	Not Available	Kernel Driver	False	Disabl
irenum	IR Enumerator Service	c:\winnt\system32\drivers...	Kernel Driver	False	Manu
isapnp	PnP ISA/EISA Bus Driver	c:\winnt\system32\drivers...	Kernel Driver	True	Boot
kbdclass	Keyboard Class Driver	c:\winnt\system32\drivers...	Kernel Driver	True	Syste
kbdhid	Keyboard HID Driver	c:\winnt\system32\drivers...	Kernel Driver	True	Syste
kmixer	Microsoft Kernel Wave Au...	c:\winnt\system32\drivers...	Kernel Driver	True	Manu
ksecdd	KSecDD	c:\winnt\system32\drivers...	Kernel Driver	True	Boot
lbrtfdc	lbrtfdc	Not Available	Kernel Driver	False	Syste
libusb0	LibUsb-Win32 - Kernel Dri...	c:\winnt\system32\drivers...	Kernel Driver	False	Manu
lp6nds35	lp6nds35	Not Available	Kernel Driver	False	Disabl
lscvpci	Lattice Virtual PCI Driver s...	c:\winnt\system32\drivers...	Kernel Driver	True	Manu
mnmdd	mnmdd	c:\winnt\system32\drivers...	Kernel Driver	True	Syste
modem	Modem	c:\winnt\system32\drivers...	Kernel Driver	False	Manu
mouclass	Mouse Class Driver	c:\winnt\system32\drivers...	Kernel Driver	True	Syste
mouhid	Mouse HID Driver	c:\winnt\system32\drivers...	Kernel Driver	True	Manu
mountmgr	MountMgr	c:\winnt\system32\drivers...	Kernel Driver	True	Boot
mraid35x	mraid35x	Not Available	Kernel Driver	False	Disabl

My Computer
Right-click on the **My Computer** icon and select **Properties** to bring up the Hardware Wizard and Device Manager. The Device Manager provides the same basic set of software driver information as in the Computer Management window. The Hardware Wizard allows the installation and removal of drivers. You must to have admin rights to run the Hardware Wizard and install/remove drivers.

Again, the most useful thing is to verify that the lscpcie driver and the lscvpci driver (if enabled) have been installed.

References

The following documents provide more information on topics discussed throughout this guide.

- PCI Express IP User's Guide
- LatticeSC80 PCI Express x8 Evaluation Board User's Guide

- LatticeSC25 PCI Express x1 Evaluation Board User's Guide
- LatticeECP2M Advanced Evaluation Board User's Guide

These documents are available on the Lattice web site at www.latticesemi.com.

Technical Support Assistance

Hotline:	1-800-LATTICE (North America)
	+1-503-268-8001 (Outside North America)
e-mail:	techsupport@latticesemi.com
Internet:	www.latticesemi.com

Revision History

Date	Version	Change Summary
September 2006	01.0	Initial release.
October 2006	01.1	Added Appendices B, C and D.
November 2006	01.2	Updated Default Jumper Settings table in Appendix D.
March 2007	01.3	Updated Demo Setups table.
		Updated Adapter Boards section.
		Added Error Messages, Unsupported Requests, Tx Arbiter and Credit Processing subsections to Memory Maps section.
April 2007	01.4	Updated Demo Setups table.
		Updated text regarding the LatticeSC80 PCI Express x8 Evaluation Board in the Introduction section.
		Updated text regarding the LatticeECP2M PCI Express x4 Evaluation Board in the Introduction section.
		Updated LED Order and Functionality table
		Updated Normal Operation table.
April 2007	01.5	Updated Presence descriptions in the Introduction section.
		Updated to reflect x4 support for LatticeECP2M.
		Updated LED definitions in Appendix D.

Appendix A. Installing the Hardware Simulator Driver

This section describes installing the optional hardware simulator device driver. This driver allows the GUI to "access" a simulated evaluation board, when in fact none is installed in the system. The hardware simulator was developed to test the demo software before the evaluation boards were available for actual use. The simulated driver can serve a purpose for those who do not have access to an evaluation board and still want to run the GUI and experience the demo software. The APIs access the driver in the same way that they access the evaluation board device driver. Therefore, this can be of some use to software developers who want to explore the operation of a device driver and interaction with user space code. The GUI is run with the simulator using the **Start -> Programs -> LatticePCIe -> SimGUI** option. For command line use, set the environment variable **PCIE_BOARD = SIM** before running the menu or GUI application. Details can be found in the batch files in the Demo directory. The installation of the driver is discussed.

Installing the Simulated Evaluation Board Device Driver

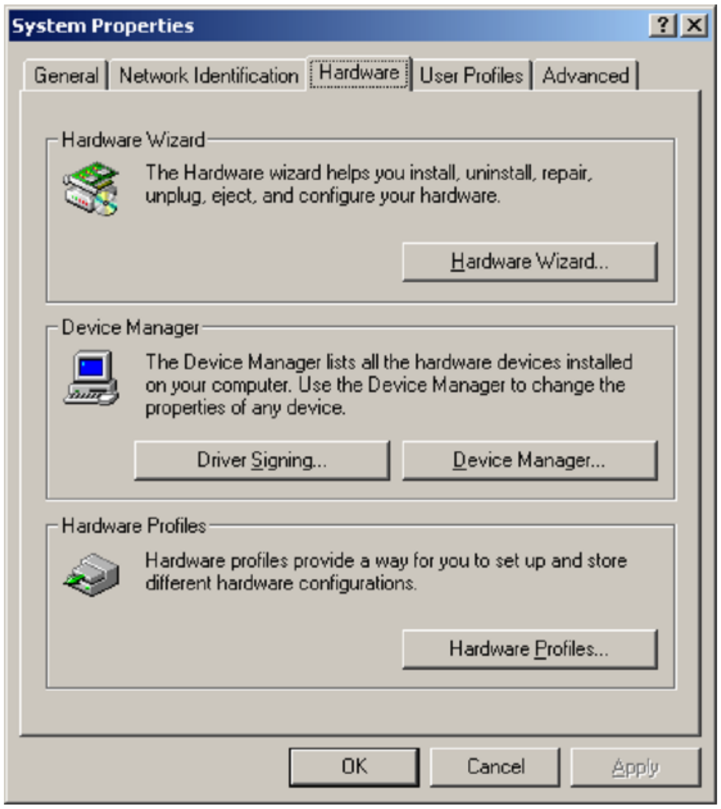
Note: You will probably need admin rights to install device drivers. See your network administrator.

Note: This installs only the hardware simulator. It does not perform any hardware accesses. All hardware resources are simulated in kernel memory structures.

Windows 2000 Install Procedures:
The major difference in installing the simulator driver is that there is no hardware for Windows to detect. Therefore, you must force it to install the driver even though it does not correspond to a physical device.

Select My computer->(right click) Properties->Hardware

Figure 35.



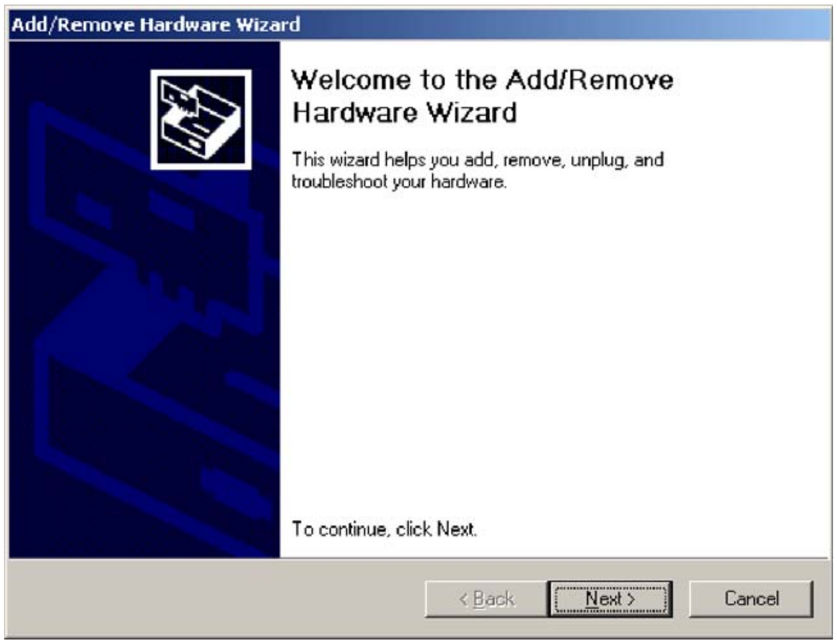
The Hardware wizard helps you install, uninstall, repair, unplug, eject, and configure your hardware.

The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

Hardware profiles provide a way for you to set up and store different hardware configurations.

Then click on the **Hardware Wizard** button.

Figure 36.



Add/Remove Hardware Wizard

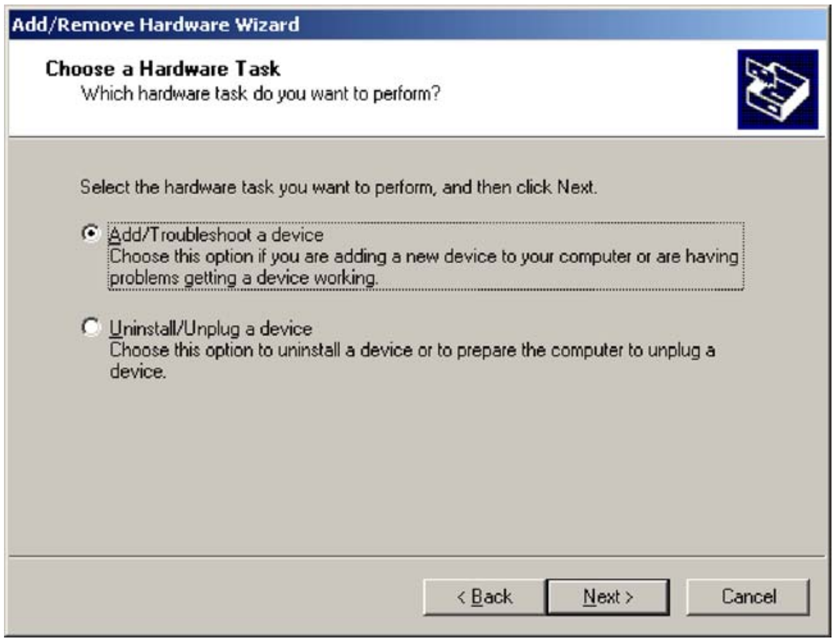
Welcome to the Add/Remove Hardware Wizard

This wizard helps you add, remove, unplug, and troubleshoot your hardware.

To continue, click Next.

Click **Next**.

Figure 37.



Add/Remove Hardware Wizard

Choose a Hardware Task
Which hardware task do you want to perform?

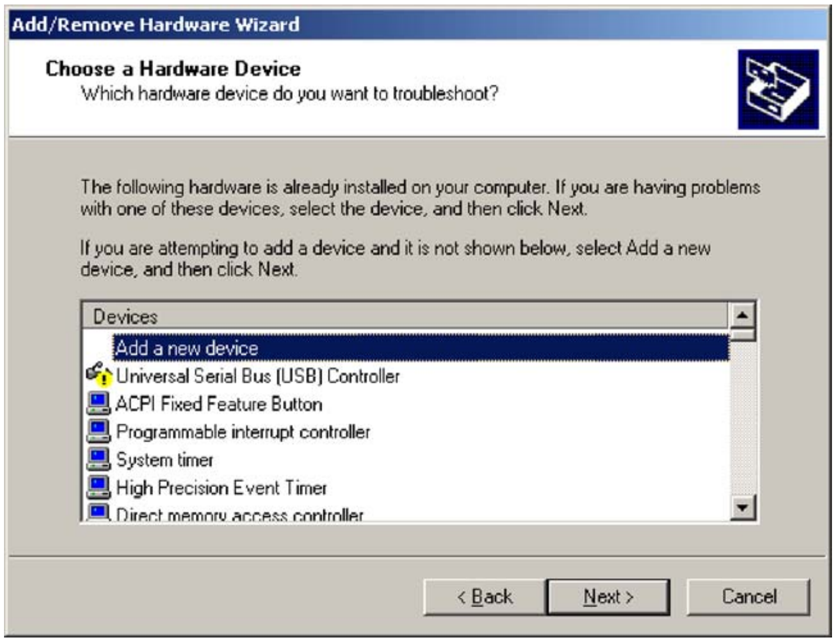
Select the hardware task you want to perform, and then click Next.

[✓ Add/Troubleshoot a device]
Choose this option if you are adding a new device to your computer or are having problems getting a device working.

Uninstall/Unplug a device
Choose this option to uninstall a device or to prepare the computer to unplug a device.

Select **Add/Troubleshoot a Device** then click **Next.** Wait while Windows searches for new hardware (it won't find any since it is virtual).

Figure 38.



Add/Remove Hardware Wizard

Choose a Hardware Device
Which hardware device do you want to troubleshoot?

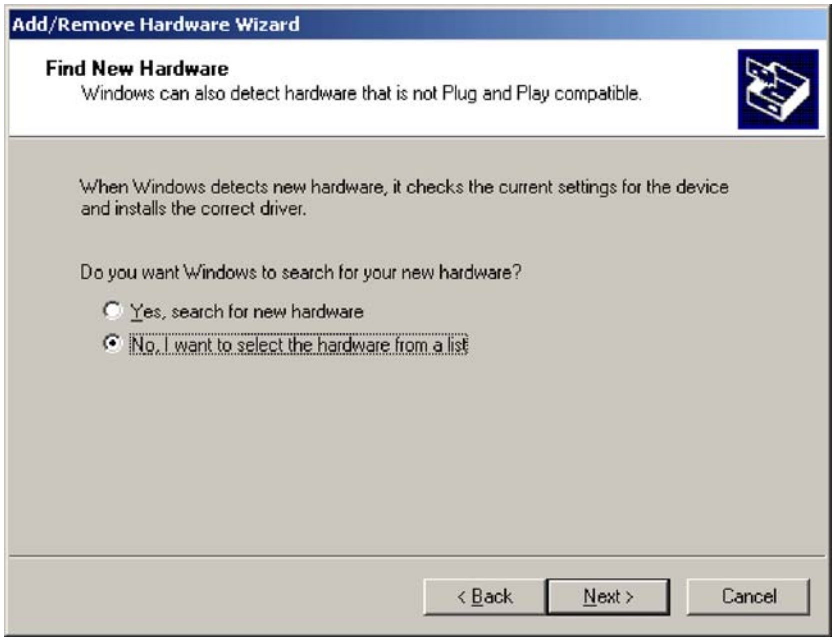
The following hardware is already installed on your computer. If you are having problems with one of these devices, select the device, and then click Next.

If you are attempting to add a device and it is not shown below, select Add a new device, and then click Next.

Devices
- Add a new device
- Universal Serial Bus (USB) Controller
- ACPI Fixed Feature Button
- Programmable interrupt controller
- System timer
- High Precision Event Timer
- Direct memory access controller

Select **Add a New Device** and click **Next**.

Figure 39.



Add/Remove Hardware Wizard

Find New Hardware
Windows can also detect hardware that is not Plug and Play compatible.

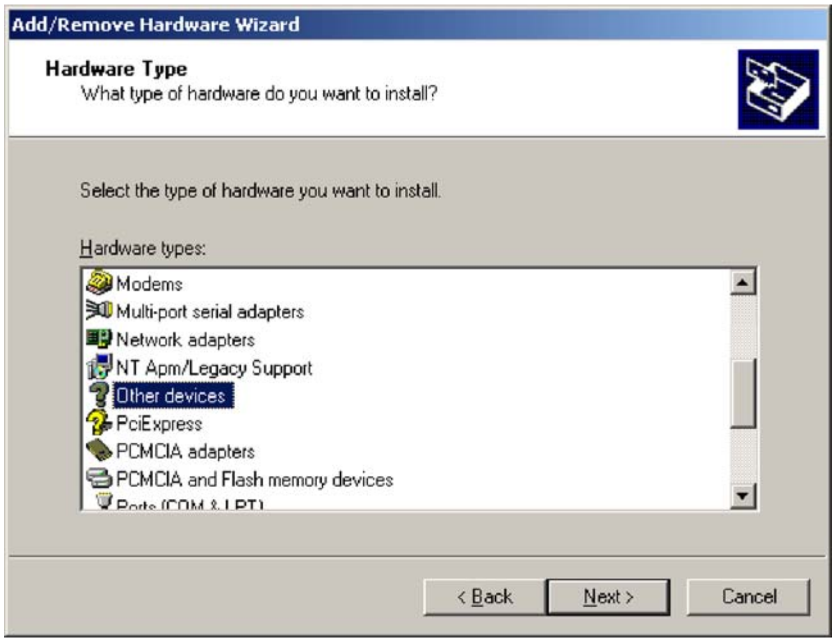
When Windows detects new hardware, it checks the current settings for the device and installs the correct driver.

Do you want Windows to search for your new hardware?

○ Yes, search for new hardware
● No, I want to select the hardware from a list

Don't let Windows search for the hardware or driver. Click **Next**.

Figure 40.


Add/Remove Hardware Wizard

Hardware Type
What type of hardware do you want to install?

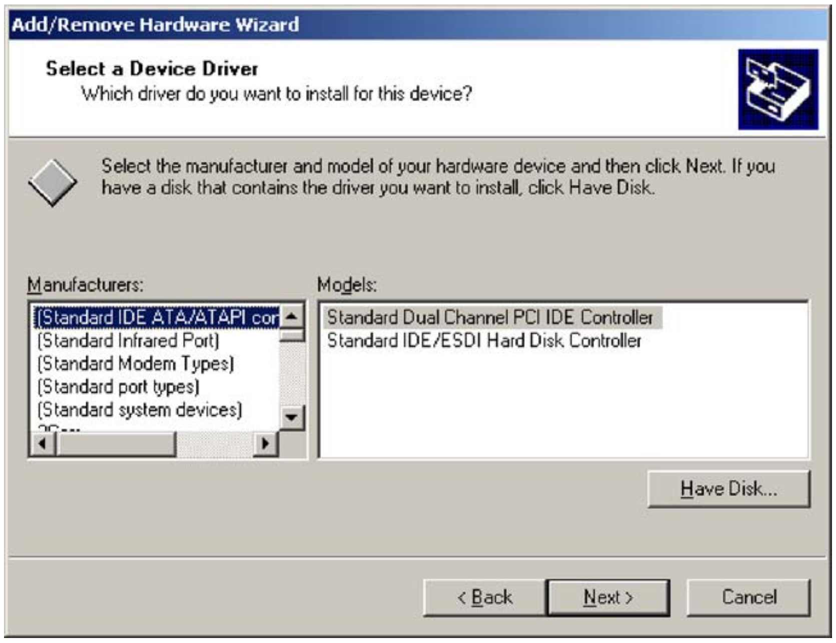
Select the type of hardware you want to install.

Hardware types:

- Modems
- Multi-port serial adapters
- Network adapters
- NT Apm/Legacy Support
- Other devices
- PciExpress
- PCMCIA adapters
- PCMCIA and Flash memory devices
- Ports (COM & LPT)

Select the Hardware Type as **Other** since the evaluation board does not fit a standard device category.

Figure 41.


Add/Remove Hardware Wizard

Select a Device Driver
Which driver do you want to install for this device?

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Manufacturers:

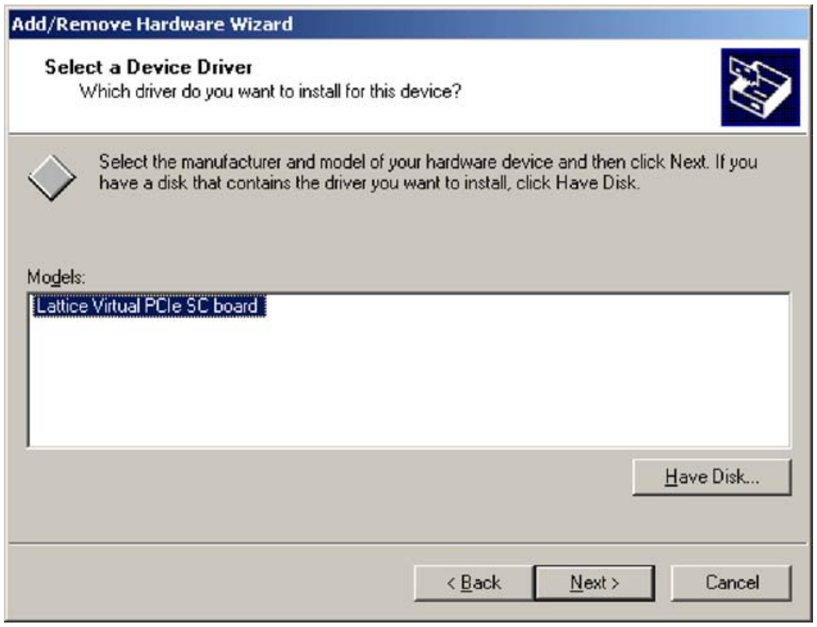
- (Standard IDE ATA/ATAPI con
- (Standard Infrared Port)
- (Standard Modem Types)
- (Standard port types)
- (Standard system devices)

Models:

- Standard Dual Channel PCI IDE Controller
- Standard IDE/ESDI Hard Disk Controller

Ignore the selections in the lists and select **Have Disk**. Browse to specify the location of the driver. Navigate to where you installed the demo package and find the **Demo\Drivers\Simulator** directory. Select **Open** and then click **OK**. This will install the driver found in the Simulator directory.

Figure 42.



Add/Remove Hardware Wizard

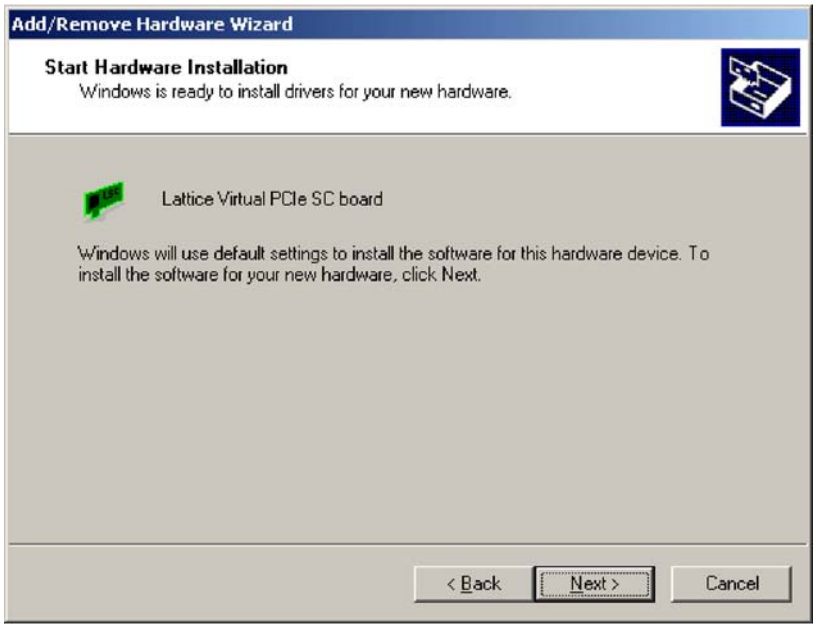
Select a Device Driver
Which driver do you want to install for this device?

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Models:
Lattice Virtual PCIe SC board

The Wizard will find the driver files and display the Lattice Virtual PCI Express LatticeSC board device to be installed. Click **Next** to continue.

Figure 43.



Add/Remove Hardware Wizard

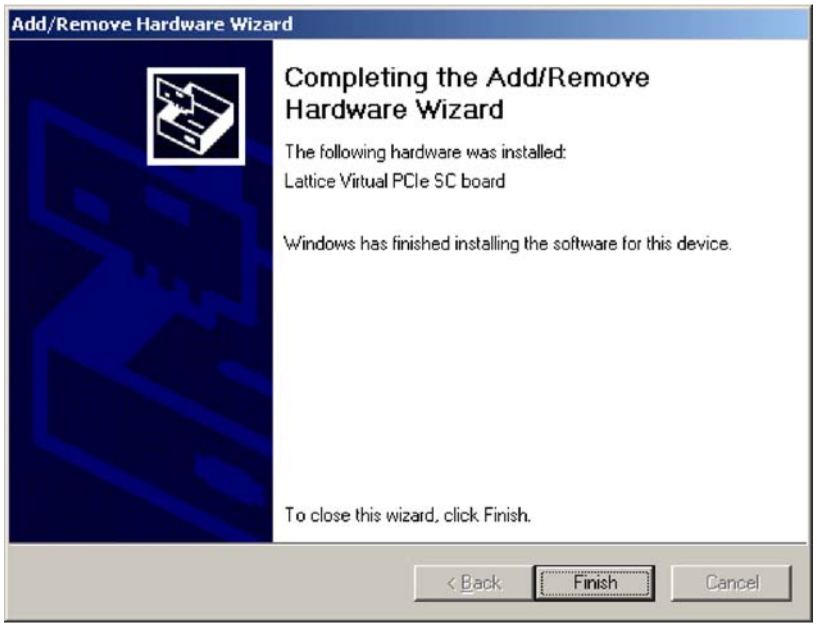
Start Hardware Installation
Windows is ready to install drivers for your new hardware.

Lattice Virtual PCIe SC board

Windows will use default settings to install the software for this hardware device. To install the software for your new hardware, click Next.

Click **Next** to continue.

Figure 44.



Add/Remove Hardware Wizard

Completing the Add/Remove Hardware Wizard

The following hardware was installed:

Lattice Virtual PCIe SC board

Windows has finished installing the software for this device.

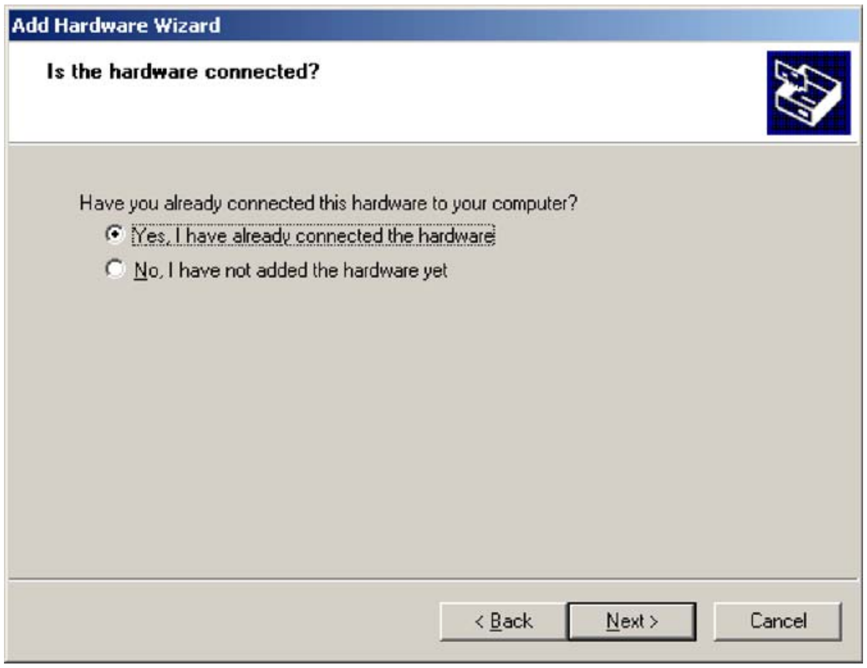
To close this wizard, click Finish.

Installation is now complete.

Windows XP Installation
In Windows XP, Add Hardware was moved to the Control Panel. Some of the screens and the order are slightly different from Windows 2000 procedures.

Start-> Control Panel -> Add Hardware starts the Hardware Wizard. Wait while it searches for hardware but it won't find it since it is a virtual device.

Figure 45.



Add Hardware Wizard

Is the hardware connected?

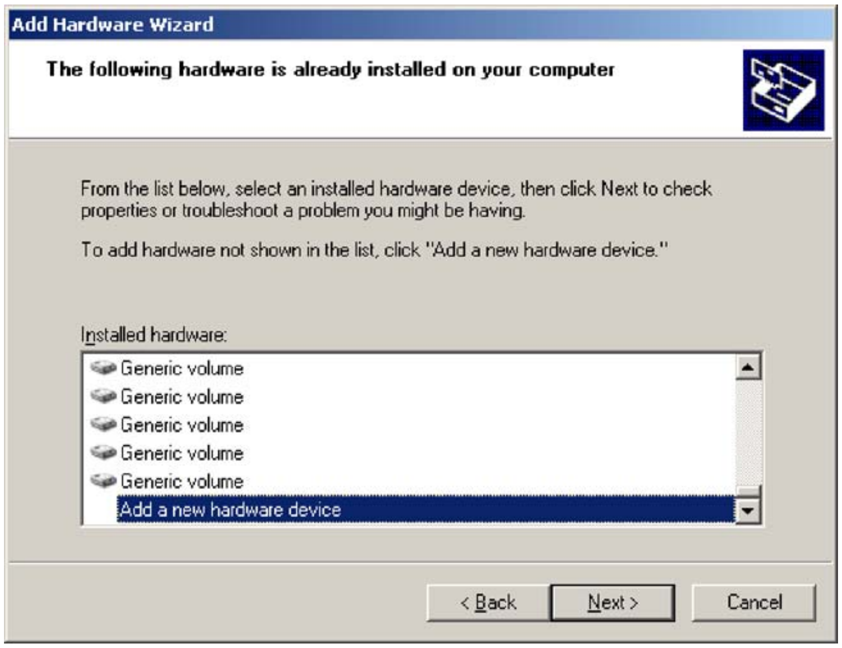
Have you already connected this hardware to your computer?

[✓ Yes, I have already connected the hardware]

No, I have not added the hardware yet

Select that you have installed the hardware and click **Next**. Scroll down and select **Add a new hardware device** (the only applicable option).

Figure 46.



Add Hardware Wizard

The following hardware is already installed on your computer

From the list below, select an installed hardware device, then click Next to check properties or troubleshoot a problem you might be having.

To add hardware not shown in the list, click "Add a new hardware device."

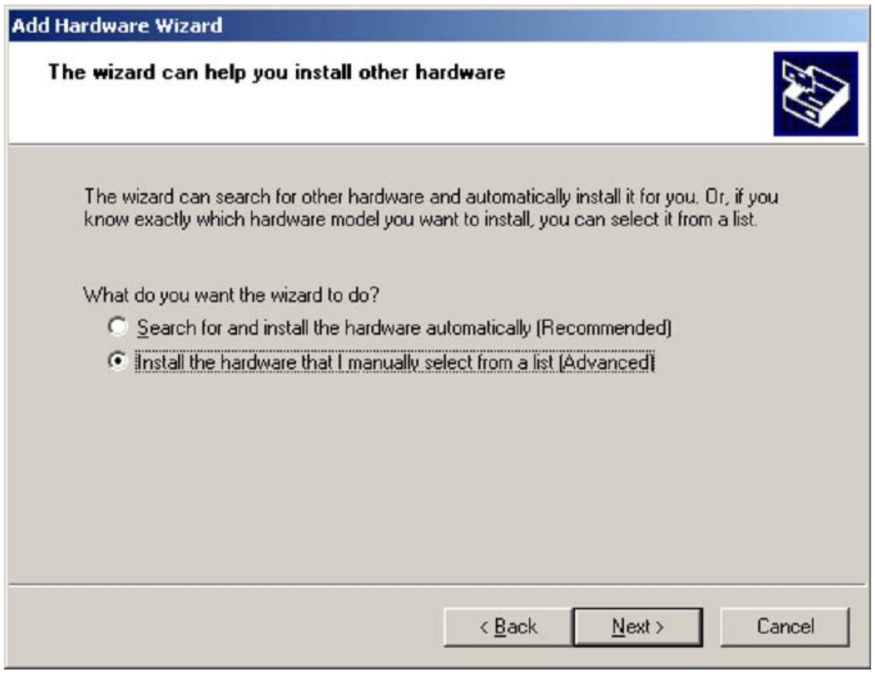
Installed hardware:

- Generic volume
- Generic volume
- Generic volume
- Generic volume
- Generic volume
- Add a new hardware device

< Back | Next > | Cancel

Select **Install the hardware that I manually select** and click **Next**.

Figure 47.



Add Hardware Wizard

The wizard can help you install other hardware

The wizard can search for other hardware and automatically install it for you. Or, if you know exactly which hardware model you want to install, you can select it from a list.

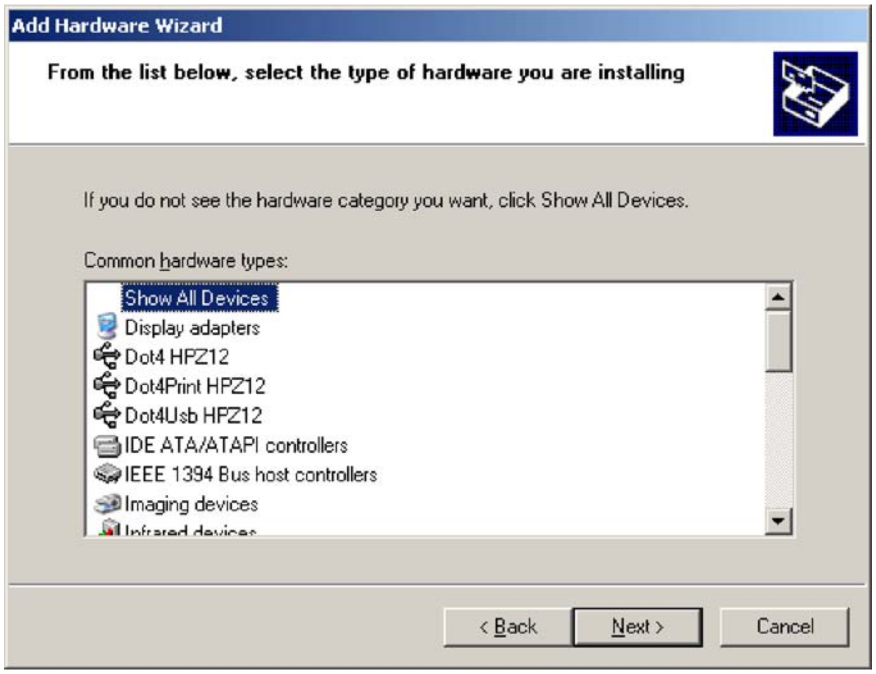
What do you want the wizard to do?

- ○ Search for and install the hardware automatically (Recommended)
- ● Install the hardware that I manually select from a list (Advanced)

< Back | Next > | Cancel

Select **Show All Devices** and click **Next**.

Figure 48.



Add Hardware Wizard

From the list below, select the type of hardware you are installing

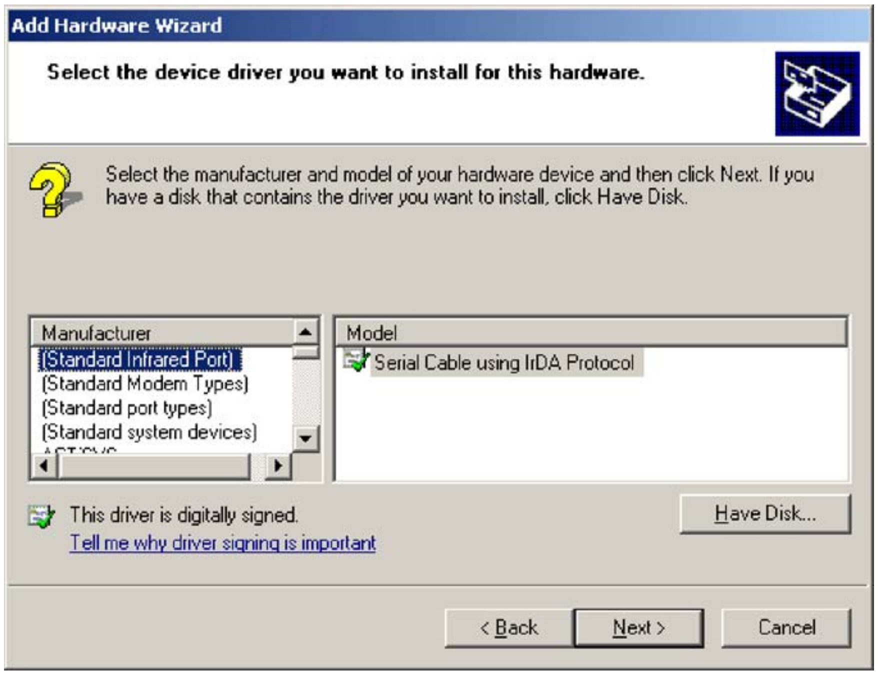
If you do not see the hardware category you want, click Show All Devices.

Common hardware types:

- Show All Devices
- Display adapters
- Dot4 HPZ12
- Dot4Print HPZ12
- Dot4Usb HPZ12
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Imaging devices
- Infrared devices

Select **Have Disk**.

Figure 49.



Add Hardware Wizard

Select the device driver you want to install for this hardware.

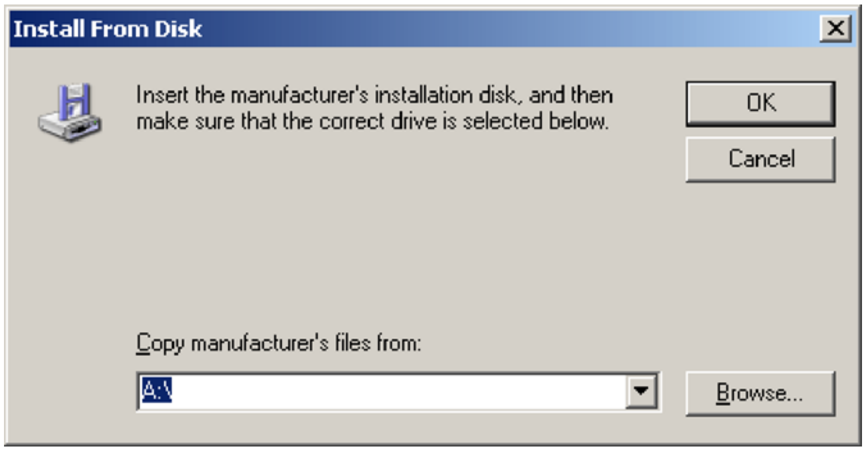
Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Manufacturer	Model
(Standard Infrared Port)	Serial Cable using IrDA Protocol
(Standard Modem Types)	
(Standard port types)	
(Standard system devices)	

This driver is digitally signed.
Tell me why driver signing is important

Then click **Next**.

Figure 50.



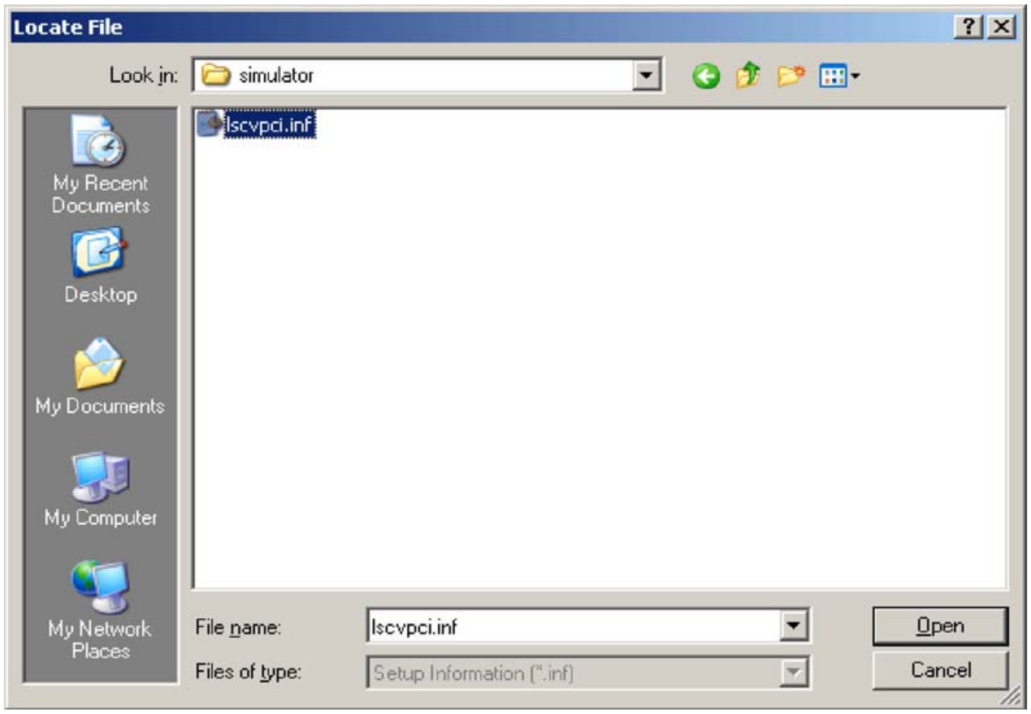
Install From Disk

Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.

Copy manufacturer's files from: A:\

Click on **Browse** and navigate to the location you installed the Demo Package (default location is **C:\Program Files\LatticeApps\PCIe\Demo\Drivers\Simulator**) lscvpcie.inf file.

Figure 51.



Locate File

Look in: simulator

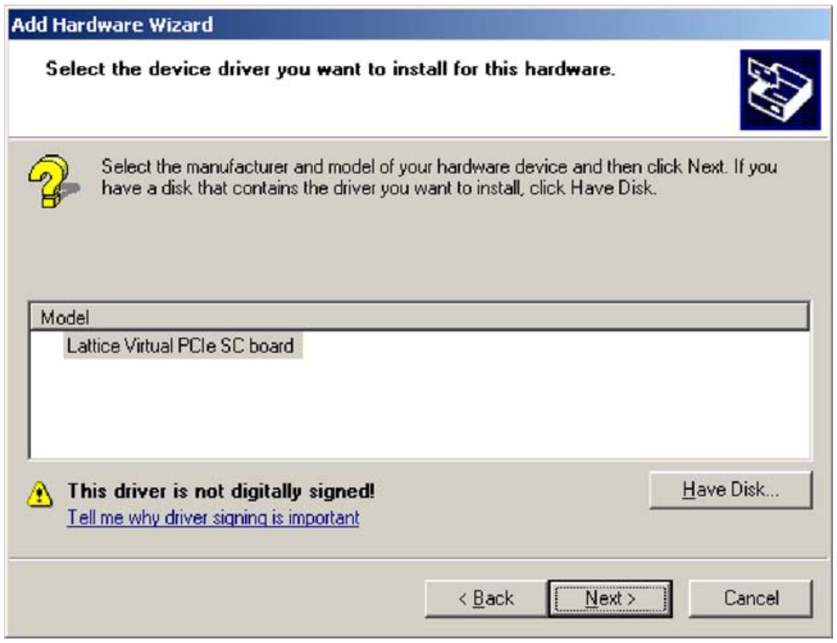
lscvpci.inf

File name: lscvpci.inf

Files of type: Setup Information (*.inf)

Click on **Open** and then **OK**.

Figure 52.



Select the device driver you want to install for this hardware.

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

Lattice Virtual PCIe SC board

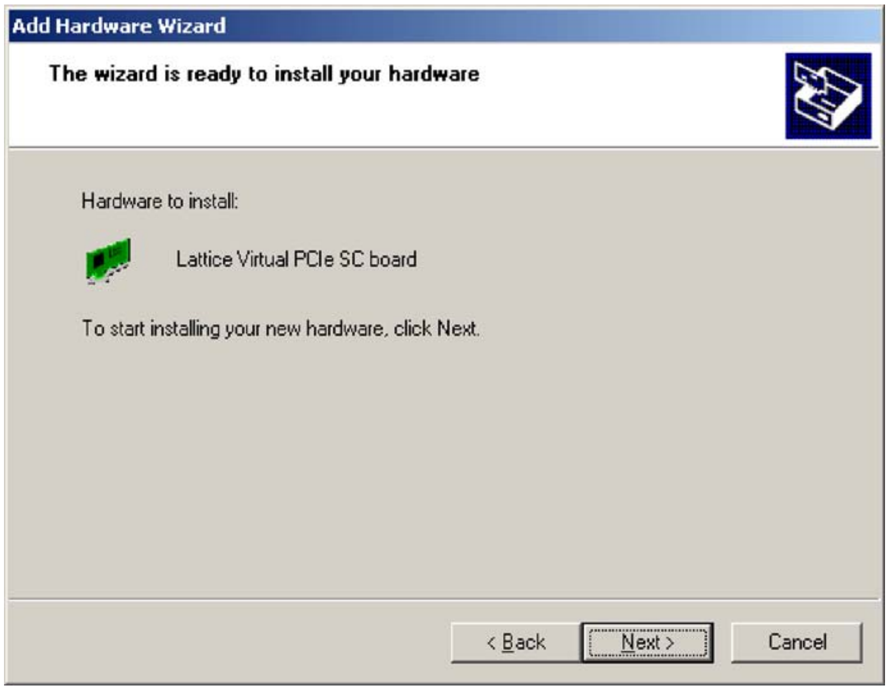
This driver is not digitally signed!

Tell me why driver signing is important

Select **Next** (ignore the message about the driver not being signed).

This screen appears indicating all driver files have been located and its ready to install.

Figure 53.



The wizard is ready to install your hardware

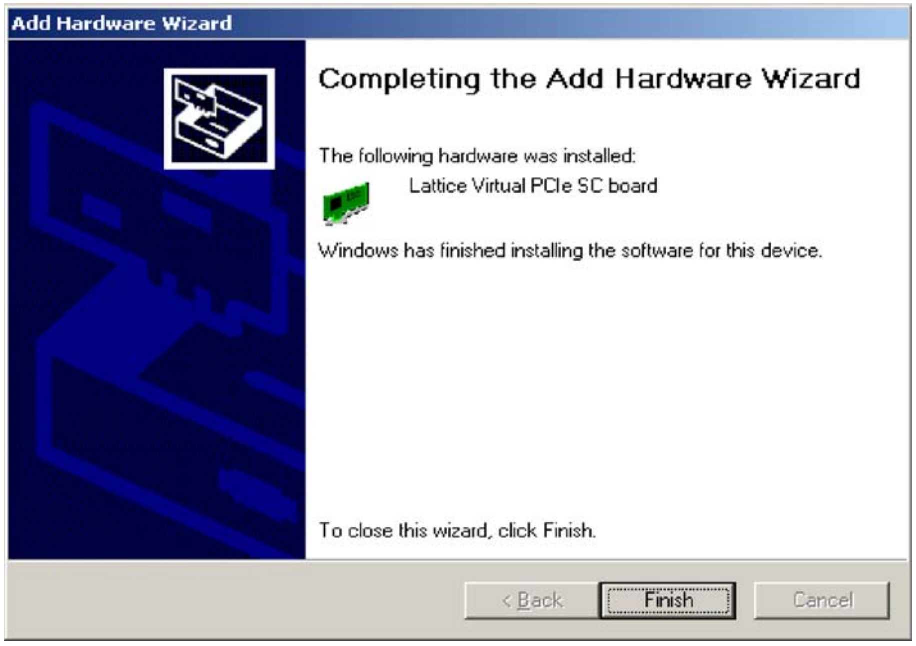
Hardware to install:

Lattice Virtual PCIe SC board

To start installing your new hardware, click Next.

Final screen indicating success.

Figure 54.



Add Hardware Wizard

Completing the Add Hardware Wizard

The following hardware was installed:

Lattice Virtual PCIe SC board

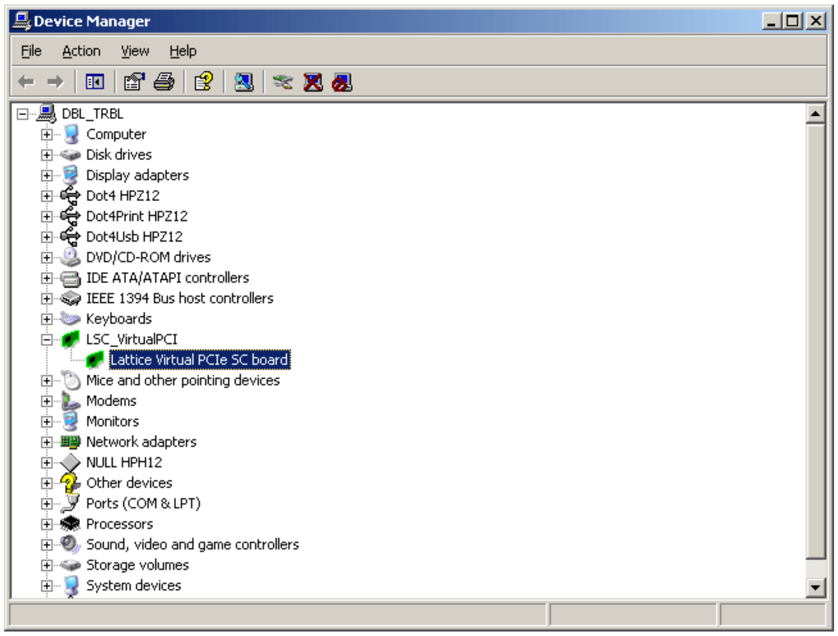
Windows has finished installing the software for this device.

To close this wizard, click Finish.

Verifying Installation

To verify that the simulator driver was installed properly, open **My Computer->Properties->Device Manager** and look for the LSC_VirtualPCI device.

Figure 55.



```
DBL_TRBL
  Computer
  Disk drives
  Display adapters
  Dot4 HPZ12
  Dot4Print HPZ12
  Dot4Usb HPZ12
  DVD/CD-ROM drives
  IDE ATA/ATAPI controllers
  IEEE 1394 Bus host controllers
  Keyboards
  LSC_VirtualPCI
    Lattice Virtual PCIe SC board
  Mice and other pointing devices
  Modems
  Monitors
  Network adapters
  NULL HPH12
  Other devices
  Ports (COM & LPT)
  Processors
  Sound, video and game controllers
  Storage volumes
  System devices
```

Appendix B. LatticeSC PCI Express x1 Evaluation Board

LED Definitions

The Status LEDs on the LatticeSC PCI Express x1 Evaluation Board are located vertically along the left edge of the board (i.e., they are visible through the expansion slot when installed in a PC). The LEDs are in the following order and have the following functions:

Table 15. LED Order and Functionality

LED	Color	Usage
D15 (top)	Red	PCI Express access to IP on wishbone bus (EBR and registers). Negative logic: on=no activity, off=activity.
D16	Orange	Data Link up, ready for packets at Transaction Layer (PCI enumeration of configuration registers).
D17	Green	L0 training sequences completed, PHY layer up and ready for flow control.
D18	Blue	Not used (off).
D19	Red	Segment K of the GUI 16-segment display. Negative logic: on=no activity, off=activity.
D20	Orange	LTSSM has completed electrical Detect state and made it to Polling state (TS1, TS2, etc.).
D21	Green	PLL locked to PCI Express 100MHz clock
D22	Blue	Segment K of the GUI 16-segment display. Negative logic: on: bit=0, off: bit=1.
D5	Green	Done
D2	Red	Init
D4	Red	Program
D3	Red	Reset

Table 16. Normal Operation

D15 (top)	Red/blinking
D16	Orange
D17	Green
D18	off
D19	Red
D20	Orange
D21	Green
D22	Blue
D5	Green
D2	off
D4	off
D3	off

Default Switch Settings

SW1 – Controls SPI flash programming (D=down, U=up).

1	2	3	4	5	6	7	8	9	10
D	D	U	D	U	D	D	D	D	D

SW4 - on back side of board, user-selectable for demo DIP switch reading.

Default Jumper Settings

J5	1-2
J7	2-3
J19	1-2
J21	Jumper Installed
J26	1-2
J27	Jumper Installed

Appendix C. LatticeSC PCI Express x8 Evaluation Board

LED Definitions

The Status LEDs on the LatticeSC PCI Express x8 Evaluation Board are located horizontally along the top right edge of the board. The LEDs are in the following order and have the following functions:

Table 17. LED Order and Functionality

LED	Color	Usage
D1 (far left)	Red	Init
D4	Green	Done
D2	Red	Not used (off).
D3	Red	Not used (off).
D5	Yellow	PLL locked to PCI Express 100MHz clock.
D6	Yellow	LTSSM has completed electrical Detect state and made it to Polling state (TS1, TS2, etc.).
D7	Green	L0 training sequences completed, PHY layer up and ready for flow control.
D8	Green	Data Link up, ready for packets at Transaction Layer (PCI enumeration of configuration registers).
D9	Blue	Not used (off).
D10	Blue	Not used (off).
D11	Green	JTAG activity.
16-Segment Decimal Point	Red	PCI Express access to IP on wishbone bus (EBR and registers). On=activity, off=bus accesses.

Table 18. Normal Operation

D1	D4	D2	D3	D5	D6	D7	D8	D9	D10	D11
off	GRN	off	off	YLW	YLW	GRN	GRN	off	off	off

Default Switch Settings

SW1 – User-selectable for demo

SW2 – User-selectable for demo

SW3 – Controls SPI flash programming (D=down, U=up)

1	2	3	4	5	6	7	8	9	10
D	D	U	D	U	D	D	D	D	D

SW8 - On back side of board, all down (off)

Default Jumper Settings

J60	1-2 (down)
J93	3-4 (right)
J96	1-2 for x1 Mode, 3-4 for x4 Mode
J102	Jumper Not Installed

Appendix D. LatticeECP2M PCI Express x4 Evaluation Board

LED Definitions

The Status LEDs on the LatticeECP2M PCI Express x4 Evaluation Board are located horizontally along the top right edge of the board. The LEDs are in the following order and have the following functions:

Table 19. LED Order and Functionality

LED	Color	Usage
D16 (far left)	Red	CDR lock
D15	Red	RxValid status
D17	Yellow	PLL locked to PCI Express 100MHz clock.
D18	Yellow	Not used (off).
D19	Green	L0 training sequences completed, PHY Layer up and ready for flow control.
D20	Green	Data Link up, ready for packets at Transaction Layer (PCI enumeration of config registers).
D21	Blue	X4 Link
D22	Blue	Not used (off).
D12	Green	JTAG activity (negative logic - off when JTAG access).
16-Segment Decimal Point	Red	PCI Express access to IP on wishbone bus (EBR and registers). On=activity, off=bus accesses.

Table 20. Normal x4 Operation

D16	D15	D17	D18	D19	D20	D21	D22	D12
RED	RED	YLW	off	GRN	GRN	BLUE	off	GRN

Table 21. Normal x1 Operation

D16	D15	D17	D18	D19	D20	D21	D22	D12
RED	RED	YLW	off	GRN	GRN	off	off	GRN

Default Switch Settings

SW1 – All ON, (in towards circuit board).

SW4 – All OFF, (towards left side).

SW5 – All down to boot, once running user selectable for demo.

SW6 – All down to boot, once running user selectable for demo.

Default Jumper Settings

J2	1-2
J3	1-2
J4	1-2
J5	1-2
J10	1-2 (Right Pair)
J11	Top Pair
J16	1-2 (Top, Across); Presence x1 Mode
J60	2-3 (Right)
J98	2-3
J99	2-3
J105	1-2, 4-6