LabVIEW™ FPGA Module Release and Upgrade Notes

Version 8.2

These release notes contain instructions for installing the LabVIEW FPGA Module, introduce new features, and provide upgrade information. Refer to the resources listed at the end of this document for information about developing applications with the FPGA Module.

Contents

NATIONAL INSTRUMENTS™

System Requirements

The *development computer* is a PC or PXI system running Windows and on which you install the LabVIEW development system and the FPGA Module. You create and edit FPGA VIs and host VIs on the development computer. A host VI is the VI you create to run in Windows or on an RT target and communicate with the FPGA VI on the FPGA target.

The development computer must meet the following requirements to run the FPGA Module.

- Windows 2000/XP
- 1.2 GHz Pentium processor or a compatible processor of equal or higher speed
- at least 512 MB memory for 1 million gate FPGA targets; at least 2 GB memory for 3 million gate FPGA targets

Note Although these memory requirements apply to typical designs, the unique characteristics of each design affect the memory requirements. Design complexity and constraints affect whether the FPGA Module can implement the design using more or less memory. You must monitor the memory and adjust the memory, if necessary.

- at least 2.5 GB additional available disk space
- LabVIEW 8.2 Full or Professional Development System

You also must have an *FPGA target* and target-specific software, which is typically packaged with the FPGA target. The FPGA target is the hardware on which the FPGA VI runs, such as an NI Reconfigurable I/O device. You can develop an FPGA VI without an FPGA target, but you cannot run the FPGA VI without an FPGA target.

Installing the LabVIEW FPGA Module

Complete the following steps to set up a development computer with LabVIEW, the LabVIEW Real-Time (RT) Module (optional), the FPGA Module, and the appropriate NI device drivers.

1. Log in to the development computer as an administrator or as a user with administrative privileges.
2. Install LabVIEW 8.2 from the LabVIEW 8.2 installation CD. Refer to the *LabVIEW Release Notes* for information about installing the LabVIEW development system.

Note The LabVIEW installer prompts you to insert the NI Device Drivers CD to install certain device drivers. If you plan to install the LabVIEW Real-Time Module, wait until after step 4 to install the NI Device Drivers CD.

3. (Optional) Install the LabVIEW Real-Time Module from the LabVIEW Real-Time Module 8.2 installation CD. Refer to the *LabVIEW Real-Time Module Release and Upgrade Notes* for information about installing the RT Module and the device drivers you need and for information about configuring RT targets.
4. Install the FPGA Module from the LabVIEW FPGA Module 8.2 installation CD, following the instructions that appear on the screen. Select **Complete** when you reach the **Select Installation Type** step. By default, the FPGA Module installer prompts you to insert the NI-RIO CD, which installs driver software for 78*xx*R Series and CompactRIO devices. Install NI-RIO if it is the appropriate driver software for the FPGA target you use.

   The FPGA Module installs program files and documentation and copies files from Xilinx ISE to the *x*:\NIFPGA82 directory, where *x* is the drive on which you installed LabVIEW 8.2. Xilinx ISE is third-party software that the FPGA Module uses to compile FPGA VIs into code that runs on the FPGA target.
5. (Optional) Install NI device drivers available on the NI Device Drivers CD.

Note The NI Device Drivers CD might not include drivers for the FPGA target you use. If the NI Device Drivers CD does not include the driver for the FPGA target you use, install FPGA target drivers from the CDs available with the targets in step 6.

6. (Optional) Install the appropriate drivers and FPGA Module support files for the FPGA target you will use. Refer to the specific hardware documentation for information about the appropriate drivers and for information about installing and configuring the FPGA target.

LabVIEW FPGA Module 8.2 Features

The FPGA Module 8.2 includes the following new features to help you better manage and implement the components of an FPGA application.

Refer to the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help** in LabVIEW, for more information about the new features.

Enhanced Support for FPGA Memory

In the FPGA Module 8.0 and earlier, you can access up to 16 KB of memory on the FPGA using the Memory Read and Memory Write functions. In the FPGA Module 8.2, you can access all available memory on the FPGA target using memory items with the Memory Read and Memory Write functions. A memory item represents the memory on the FPGA target. A memory item can be of a Boolean data type or an 8-, 16-, 32-, or 64-bit signed or unsigned integer data type. You can use the Memory Read and Memory Write functions in a single-cycle Timed Loop.

To access FPGA memory, you must create target-scoped or VI-scoped memory items to store and retrieve data. Target-scoped memory items are available within the VI hierarchy of any FPGA VI under the same target in the **Project Explorer** window. VI-scoped memory items are available only within the VI in which they reside. If you use a VI-scoped memory item in a reentrant FPGA VI, LabVIEW creates a separate copy of the memory item for each instance of the VI, which allows you to create reusable subVIs while avoiding resource conflicts.

To create a target-scoped memory item, right-click the FPGA target in the **Project Explorer** window and select **New»Memory** from the shortcut menu. In the **Memory Properties** dialog box that appears, configure the memory item. You then can drag a memory item from the **Project Explorer** window to the block diagram to create a Memory Read function that is associated with the memory item. Right-click the Memory Read function and select **Select Method»Write** from the shortcut menu to switch to a Memory Write function.

To create a VI-scoped memory item, place a VI-Scoped Memory Configuration node on the block diagram, right-click the node, and select **Properties** from the shortcut menu to display the **Memory Properties** dialog box. Use the **Memory Properties** dialog box to configure the memory item. After you configure the memory item, place a Memory Read or Memory Write function on the block diagram. Right-click the function and select **Select Memory»*x*** from the shortcut menu, where ***x*** is the name of the memory item on the block diagram. LabVIEW prepends VI:: to the name of VI-scoped memory items.

Refer to the *Memory Read and Memory Write Functions* section of this document for information about upgrading the Memory Read and Memory Write functions in FPGA VIs from 8.0 and earlier.

High Speed Data Transfer from Host to FPGA Target Using DMA FIFOs

The FPGA Module 8.2 includes support for writing to DMA FIFOs from host VIs. When you create a DMA FIFO in the **FPGA FIFO Properties** dialog box, select **DMA—Host to Target** from the **Type** pull-down menu. In the host VI, use the Invoke Method function with the Write method to write to the DMA FIFO.

Note Some targets might not support DMA FIFOs. Refer to the specific hardware documentation for information about supported features.

New FPGA Math & Signal Processing VIs

The FPGA Module 8.2 includes the following new VIs:

- Butterworth Filter—Filters an input signal using a Butterworth filter.
- DC and RMS Measurements—Calculates the DC, RMS, Sum, Mean Square, and/or Square Sum values of an input signal.
- Analog Period Measurement—Calculates the period of an evenly sampled periodic signal using threshold crossing detection.

Improved FPGA Code Reuse with VI-Scoped FIFOs

The FPGA Module 8.2 includes VI-scoped FIFOs, which are available only within the VI in which they reside. Use VI-scoped FIFOs to pass data within a VI. If you use a VI-scoped FIFO in a reentrant FPGA VI, LabVIEW creates a separate copy of the FIFO for each instance of the VI, which allows you to create reusable subVIs while avoiding resource conflicts.

To create a VI-scoped FIFO, place a VI-Scoped FIFO Configuration node on the block diagram, right-click the node, and select **Properties** from the shortcut menu to display the **FPGA FIFO Properties** dialog box. Use the **FPGA FIFO Properties** dialog box to configure a FIFO. After you configure the FIFO, place a FIFO Read or FIFO Write function on the block diagram. Right-click the function and select **Select FIFO»*x*** from the shortcut menu, where ***x*** is the name of the FIFO on the block diagram. LabVIEW prepends VI:: to the name of VI-scoped FIFOs.

Support for 64-bit Integers

The LabVIEW FPGA Module 8.2 introduces support for 64-bit integers. You can use 64-bit signed and unsigned integers with all arithmetic operations and comparisons, the Saturation Arithmetic VIs, memory, and the HDL Interface Node.

Improved Saturation Arithmetic VIs

In the FPGA Module 8.0 and earlier, you must use the configuration dialog box to select the data type for inputs on the Saturation Arithmetic VIs. In the FPGA Module 8.2, you can wire 8-, 16-, 32-, and 64-bit data types directly to the inputs of the Saturation Add and Saturation Subtract VIs without using the configuration dialog box. You can wire 8-, 16-, and 32-bit data types directly to the inputs of the Saturation Multiply VI without using the configuration dialog box.

Use the configuration dialog box for the Saturation Arithmetic VIs to select the width of the output data type. You can select **Match input**, **Avoid overflow**, or **Fixed** as the output width. The default output width for the Saturation Add and Saturation Subtract VIs is **Match input**. The default output width for the Saturation Multiply VI is **Avoid overflow**.

HDL Interface Node Enhancements

The FPGA Module 8.2 includes the following enhancements to the HDL Interface Node:

- The HDL Interface Node is no longer limited to 28 input and output parameters.
- The HDL Interface Node includes a new look, which displays the names of input and output parameters.

Easier to Understand Code Generation Errors

In the FPGA Module 8.0 and earlier, the **Code Generation Errors** window usually refers to objects that contain errors using internal names, which makes it difficult to determine where and how to correct an error. In the FPGA Module 8.2, the **Code Generation Errors** window returns errors using the visible names of objects, provides suggestions for correcting the errors, and highlights the location of the errors. Click the **Show Error** button in the **Code Generation Errors** window to highlight the location of the error in the VI or **Project Explorer** window.

Resolving Timing Violations in Single-Cycle Timed Loops

The FPGA Module 8.2 returns additional information about compilation failures due to timing violations. LabVIEW cannot always detect the location of the violation. If LabVIEW can detect the location of the violation and the violation occurs within a single-cycle Timed Loop, the **Compilation Failure** dialog box returns the name of the single-cycle Timed Loop. You can use this information to identify where you need to make changes in the FPGA VI to resolve the violation.

Upgrade and Compatibility Issues

Refer to the following sections for upgrade and compatibility issues specific to different versions of the FPGA Module.

Upgrading from FPGA Module 8.0

You might encounter the following compatibility issues when you upgrade to FPGA Module 8.2 from FPGA Module 8.0.

Memory Read and Memory Write Functions

If you use the Memory Read or Memory Write functions in an FPGA VI from LabVIEW 8.0 or earlier, the FPGA Module 8.2 does not update the Memory Read and Memory Write functions to 8.2 functionality. All Memory Read and Memory Write functions from 8.0 or earlier continue to work in the FPGA Module 8.2.

If you created memory blocks using the FPGA Module Memory Extension Utility on `ni.com`, the FPGA Module 8.2 does not convert the memory blocks to use an 8.2 memory item. The memory blocks continue to work in the FPGA Module 8.2. However, the memory blocks might not be supported in a future release of the FPGA Module. To replace a memory block, create a new memory item with the same configuration as the memory block and configure the Memory Read and Memory Write functions to access the new memory item.

Wait on Occurrence Function

In the FPGA Module 8.0 and earlier, when you use the Wait on Occurrence function in an FPGA VI, the function uses ticks as the unit for the **ms timeout** parameter. When you use the Wait on Occurrence function in a Windows VI, the unit is milliseconds. In the FPGA Module 8.2, the Wait on Occurrence function uses milliseconds in both FPGA and Windows VIs. The FPGA Module 8.2 also includes a new Wait on Occurrence with Timeout in Ticks function.

When you open an FPGA VI that is saved in LabVIEW 8.0 or earlier in the application instance for an FPGA target, the FPGA Module 8.2 replaces the Wait on Occurrence function with the new Wait on Occurrence with Timeout in Ticks function. If you use the **Import FPGA Files from LabVIEW FPGA Module 1.X** utility, the FPGA Module 8.2 replaces the Wait on Occurrence function with the new Wait on Occurrence with Timeout in Ticks function in all FPGA VIs.

However, the FPGA Module might not replace the Wait on Occurrence function in the following cases:

- The FPGA VI was last saved on a non-FPGA target in LabVIEW 8.0 and you open and save (or mass compile) the VI in a non-FPGA target in LabVIEW 8.2.
- The FPGA VI was last saved in LabVIEW 7.*x* and you open and save (or mass compile) the VI in LabVIEW 8.2 in a non-FPGA target.
- The FPGA VI was saved or mass compiled in LabVIEW 8.2 before installing the FPGA Module.

In the cases listed above, you must manually replace the Wait on Occurrence function with the Wait on Occurrence with Timeout in Ticks function to continue using the ticks unit.

Tunnels and Shift Registers on For Loops

In the FPGA Module 8.0 and earlier, if the value wired to the count (**N**) terminal on a For Loop is `0`, the outputs from the tunnels and shift registers are undefined. In the FPGA Module 8.2, the output tunnels and shift register terminal on the right side of the For Loop contain an extra MUX to handle a `0` value wired to the count terminal in a manner consistent with LabVIEW for Windows. As a result, FPGA VIs that use the output tunnels or right shift register terminal in a For Loop might use more FPGA resources and/or compile at a slightly lower clock rate.

Upgrading from FPGA Module 1.*x*

You might encounter the following compatibility issues when you upgrade to FPGA Module 8.2 from FPGA Module 1.*x*. Refer to the *Upgrading from FPGA Module 8.0* section of this document for information about other upgrade issues you might encounter.

Refer to `ni.com/info` and enter the info code `ex8kk4` for more information about how to perform FPGA Module 1.*x* tasks with the FPGA Module 8.2.

Refer to the *LabVIEW FPGA Module Release and Upgrade Notes* for each version of the LabVIEW FPGA Module between versions 1.*x* and 8.2 at `ni.com/manuals` for more information about the new features and changes in each version.

FPGA Device I/O Functions

The FPGA Module no longer supports the FPGA Device I/O functions. Use the FPGA I/O Node configured to target-specific I/O resources instead.

Embedded Project Manager

The FPGA Module no longer supports the Embedded Project Manager. Use the **Project Explorer** window instead.

Upgrading FPGA VIs, Host VIs, and Embedded Projects from the LabVIEW FPGA Module 1.*x* to 8.2

If you created FPGA VIs or host VIs using LabVIEW 7.*x* and the FPGA Module 1.*x*, you can update the VIs for use with LabVIEW 8.2 and the FPGA Module 8.2. Complete the following steps to update the VIs for use with LabVIEW 8.2 and the FPGA Module 8.2.

Note NI recommends you back up your files before you begin updating them for use with the FPGA Module 8.2.

1. Launch LabVIEW 8.2.
2. Select **Tools»FPGA Module»Import FPGA Files from LabVIEW FPGA Module 1.X** in the **Getting Started** window or in the **Project Explorer** window of an open project to launch the **Import FPGA Files from LabVIEW FPGA Module 1.X** utility. If you do not have a project open, LabVIEW prompts you to create a new project.

- Upgrading from the FPGA Module 1.1 only.

3. Complete the following steps to add the LEP file you want to import to the **LabVIEW Embedded Project (*.lep) Files** list.
   a. Click the **Add** button below the **LabVIEW Embedded Project (*.lep) Files** list. The **Select an Embedded Project to Add** dialog box appears.
   b. Navigate to the location of the LEP file you want to import.
   c. Click the **Add** button. LabVIEW adds the LEP file and all the VIs associated with the LEP file to the **Import FPGA Files from LabVIEW FPGA Module 1.X** utility. You can see only the LEP file in the **LabVIEW Embedded Project (*.lep) Files** list.
   d. Proceed to step 5.

- Upgrading from the FPGA Module 1.0 only.

4. Complete the following steps to add the FPGA VIs you want to import to the **FPGA 1.0 VIs** list.
   a. Click the **Add** button below the **FPGA 1.0 VIs** list. The **Select a VI to Add** dialog box appears.
   b. Navigate to the location of the FPGA VI you want to import.

c. Click the **Add** button. LabVIEW adds the FPGA VI to the **Import FPGA Files from LabVIEW FPGA Module 1.X** utility.

d. Proceed to step 5.

5. Add host VIs from the FPGA Module 1.*x* you want to import to the **Host VIs** list.
6. Click the **Continue** button. The **Import Status** dialog box appears. When LabVIEW finishes importing the 1.*x* files, the **Save VIs** dialog box appears.
7. Select a location in the **Save Location** section to save the imported VIs. Select **Next to Project** if you want to save the new VIs in the same location as the new project. National Instruments recommends this option. Select **Choose a Location** if you want to specify a different location. Select **Defer Decision - Leave VIs open in memory** if you want to wait until later to save the VIs.
8. If you select **Choose a Location** or **Next to Project** and you have not saved the project, click the **Continue** button. If you select **Choose a Location**, the **Location to Save** dialog box appears. If you select **Next to Project**, the **Name the Project** dialog box appears. Proceed to step 9.

   If you have saved the project or if you select **Defer Decision - Leave VIs open in memory**, click the **Finish** button. LabVIEW returns you to the project. If you select **Defer Decision - Leave VIs open in memory**, LabVIEW also opens the front panel windows of the top-level VIs.
9. If you have the **Location to Save** dialog box open, choose an existing folder or create a new folder in which to save the new project and click the **Save** button. If you have the **Name the Project** dialog box open, type a name for the new LabVIEW project file (`.lvproj`) and click the **Save** button to save the new project.

The project file now includes all the VI and I/O information, as well as the FPGA target(s) on which the FPGA VI runs.

Where to Go from Here

National Instruments provides many resources to help you succeed with your NI products. Use the following resources as you start exploring LabVIEW and the FPGA Module.

Examples

The driver software for many FPGA targets includes corresponding examples. Refer to the specific hardware documentation for information about whether the FPGA target you use comes with corresponding examples.

You can start with an existing example and use it as a framework for developing FPGA VIs and host VIs. From LabVIEW, launch the NI Example Finder by selecting **Help»Find Examples**. Browse the examples by directory or by task.

Related Documentation

Refer to the *LabVIEW Help* for information about LabVIEW and the FPGA Module, and refer to the **Context Help** window for information about using VIs and functions.

- *LabVIEW Help*—Available by selecting **Help»Search the LabVIEW Help** in LabVIEW. Browse the **FPGA Module** book in the **Contents** tab for an overview of the FPGA Module and hardware-specific information included in the *LabVIEW Help*. Browse the **FPGA Interface** book in the **Contents** tab for an overview of the FPGA Interface information included in the *LabVIEW Help*.
- **Context Help** window—Available by selecting **Help»Show Context Help**.

Refer to the specific hardware documentation for information about using the FPGA target with LabVIEW and the FPGA Module and for information about hardware specifications.

NI Web Site

Visit `ni.com/fpga` for the latest NI Developer Zone articles, examples, and support information for the FPGA Module.

Refer to `ni.com/info` and enter the info code `fpgatrn` to access online training for the FPGA Module.

Known Issues

Refer to `readme_fpga.html` on the LabVIEW FPGA Module installation CD for information about known issues with the FPGA Module. You also can find the `readme_fpga.html` file installed in the `LabVIEW 8.2\readme` directory. Select **Start»All Programs»National Instruments»LabVIEW 8.2»Readme** to open the `LabVIEW 8.2\readme` directory.

371572B-01 Jul06