FPGA Interface

June 2008, 371107E-01

The FPGA Interface provides programming features and functions to help you communicate with VIs or bitfiles that run on National Instruments FPGA targets, such as National Instruments Reconfigurable I/O (NI-RIO) devices. The VI that runs on the FPGA target is called the FPGA VI. The VI you use to communicate with FPGA VIs or bitfiles is called the host VI. The host VI can run on PCs or RT targets.

The FPGA Interface is available with FPGA target driver software. You can use the <u>FPGA Interface</u> functions in a host VI if you have an FPGA target and the appropriate driver software. You do not need the LabVIEW FPGA Module to use the FPGA Interface functions. You need the FPGA Module if you want to develop FPGA VIs.

- To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW Help* highlights this topic in the **Contents** tab so you can navigate the related topics.
- © 2003–2008 National Instruments Corporation. All rights reserved.

Using LabVIEW FPGA Interface without the FPGA Module (FPGA Interface)

If you have an FPGA target, you can use the LabVIEW FPGA Interface VIs and functions to interface with an FPGA bitfile even if you do not have the LabVIEW FPGA Module installed. Use the <u>Open FPGA VI Reference</u> function to <u>open a reference to the FPGA bitfile</u>. You then can use other FPGA Interface functions, such as <u>Read/Write Control</u>, <u>Invoke Method</u>, and <u>Close FPGA VI Reference</u>, to interface with the FPGA bitfile.



Note You can use an uncompiled FPGA VI only if you have the LabVIEW FPGA Module installed.

Communicating with FPGA VIs (FPGA Interface)

This book provides functions and procedures that enable host VIs to communicate with FPGA VIs.

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The *LabVIEW Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

Opening a Reference to an FPGA VI (FPGA Interface)

You can communicate with FPGA VIs or bitfiles running on FPGA targets using host VIs. Host VIs can run on PCs or RT targets. Each host VI must open a reference to the FPGA VI or bitfile that runs on the FPGA target. You can open a reference to an FPGA VI in the same LabVIEW project as the host VI, or you can open a reference to a bitfile.





Tip To make host VI development faster, keep the FPGA VI in memory.

Opening a Reference to an FPGA VI

Complete the following steps to open a reference to an FPGA VI in a host VI. You can open a reference to an FPGA VI if the host VI, the FPGA target, and the FPGA VI are in the same project. You cannot open a reference to an FPGA VI if you do not have the LabVIEW FPGA Module installed.

- 1. <u>Create a new project</u> or open an existing project.
- 2. <u>Add an FPGA target</u> to the project or verify the FPGA target appears in the <u>Project Explorer</u> window.
- 3. <u>Create a new FPGA VI</u> or verify the FPGA VI to which you want to open a reference appears in the **Project Explorer** window under the FPGA target.
- 4. Create a new host VI or open an existing host VI in the project. The host VI must be under **My Computer** or an RT target in the **Project Explorer** window.
- 5. Place an <u>Open FPGA VI Reference</u> function on the block diagram.

■ Place ■ Find

- 6. Drag the FPGA VI you want to open a reference to from the **Project Explorer** window to the Open FPGA VI Reference function. The FPGA VI icon appears in the Open FPGA VI Reference function. If the FPGA target in the project is associated with a physical target, the target name and resource appears under the Open FPGA VI Reference function.
- (Optional) Right-click the Open FPGA VI Reference function and select Configure Open FPGA VI Reference from the shortcut menu to display the <u>Configure Open FPGA VI Reference</u> dialog box, which you can use to select options for opening the reference.
- 8. (Optional) Wire a control or constant to the **resource name** input on the Open FPGA VI Reference function to specify an FPGA target on which to run the FPGA VI.

Note You must place and wire a <u>Close FPGA VI Reference</u> function for every Open FPGA VI Reference function in a host VI.

Opening a Reference to a Bitfile

Complete the following steps to open a reference to a bitfile in a host VI. The host VI does not need to be in a project.

- 1. Create a new host VI or open an existing host VI. If the host VI is in a project, the host VI must be under **My Computer** or an RT target in the **Project Explorer** window.
- 2. Place an <u>Open FPGA VI Reference</u> function on the block diagram.

Place Find

- 3. Right-click the Open FPGA VI Reference function and select **Configure Open FPGA VI Reference** from the shortcut menu.
- 4. Select the **Bitfile** option in the **Configure Open FPGA VI Reference** dialog box.
- 5. Navigate to the bitfile you want to open on an FPGA target.
- 6. (Optional) Use the **Configure Open FPGA VI Reference** dialog box to select additional options for opening the reference.
- 7. Click the **OK** button. The FPGA VI icon appears in the Open FPGA VI Reference function. A folder icon appears in the upper left corner of the Open FPGA VI Reference function to denote the bitfile.
- 8. Wire a control or constant to the **resource name** input on the Open FPGA VI Reference function to specify an FPGA target on which to run the FPGA VI.
- Note You must place and wire a <u>Close FPGA VI Reference</u> function for every Open FPGA VI Reference function in a host VI.

Communicating with an FPGA VI Running on a Development Computer (FPGA Interface)

The following list describes behavior to consider when you use a host VI to communicate with an FPGA VI that is <u>running on a development</u> <u>computer</u> with simulated I/O:

- You must run the host VI on a Windows development computer to communicate with the FPGA VI. You cannot run the host VI on an RT target.
- To execute different code on the host VI based on where the FPGA VI executes, use the <u>Invoke Method</u> function configured for the Get FPGA VI Execution Mode method.
- The Up Cast function and the Abort, Reset, and Download methods on the Invoke Method function do not support running the FPGA VI on the development computer. If you use the Up Cast function or these methods, the host VI returns a run-time error.
- <u>Direct Memory Access (DMA) FIFOs</u> are valid while either the host VI or FPGA VI are running. If both VIs stop running, DMA FIFOs lose all data.
- Interrupts are valid only when the FPGA VI is running. If the FPGA VI stops running, all interrupt data is lost and any host interface waits return immediately.
- If you use the Invoke Method function to read DMA FIFOs, the function might timeout more frequently because the FPGA VI is not running as fast on the development computer as it would on an FPGA target.
- You must close the front panel window of the FPGA VI before running the FPGA VI if you want to use the <u>Close FPGA VI</u> <u>Reference</u> function to close the reference, stop the FPGA VI, and reset the FPGA VI running on the development computer. You must open the front panel window of the FPGA VI before running the FPGA VI if you want to use the Close FPGA VI Reference function to close the host reference without resetting the FPGA VI running on the development computer.

Using Multiple FPGA VI References for the Same Target (FPGA Interface)

If you want to <u>open references</u> to different FPGA VIs or bitfiles on one target, open only one reference at a time, and close the reference before opening another.

You can have more than one FPGA VI reference simultaneously open on a target, as long as all the references correspond to the same FPGA VI or bitfile on the same target. You can take advantage of this if you want to structure an application so that different parts of the host VI communicate with certain parts of the FPGA VI. The host VI accesses certain resources from the FPGA. The resources can include controls and indicators, logical interrupts, and DMA channels. The best way to partition the host VI is to access one of the FPGA resources using one FPGA VI reference. The only type of access that you can always safely make using multiple FPGA VI references is reading of controls and indicators.

Reading FPGA VI Indicators (FPGA Interface)

Complete the following steps to read an indicator in an FPGA VI from a host VI. The FPGA VI must have front panel indicators.

- 1. <u>Open a reference</u> to the FPGA VI or bitfile.
 - Note The FPGA target, FPGA VI, and host VI must be in the same LabVIEW project if you want to open a reference to an FPGA VI. The host VI does not need to be in a project if you open a reference to a bitfile.
- 2. Place the <u>Read/Write Control</u> function on the block diagram. Notice that the Read/Write Control function contains one **Unselected** input.

■ Place ■ Find

- 3. Wire the **FPGA VI Reference Out** parameter of the <u>Open FPGA</u> <u>VI Reference</u> function to the **FPGA VI Reference In** parameter of the Read/Write Control function.
- 4. Click the **Unselected** input. The shortcut menu lists all front panel controls and indicators in the FPGA VI in the **Controls** submenu.
- 5. Select an indicator available in the FPGA VI from the shortcut menu. Notice that the **Unselected** input changes to an output and reflects the name of the indicator in the FPGA VI.

To read more indicators in the FPGA VI, right-click the Read/Write Control function and select **Add Element** from the shortcut menu and then customize the output as described in the previous step. You also can click the bottom line of the Read/Write Control function with the Positioning tool and drag the line down to add more controls and indicators. The read and write operations execute sequentially from top to bottom. Also, if the FPGA VI has subVIs and you want to access controls and indicators on the subVI, you must wire the subVI controls and indicators to the controls and indicators of the FPGA VI that is used in the Open FPGA VI Reference function.

Note You also can read FPGA VI controls. To read a control, rightclick the control input on the Read/Write Control function and select **Change to Read** from the shortcut menu.

Writing to FPGA VI Controls (FPGA Interface)

Complete the following steps to write to a control in an FPGA VI from a host VI. The FPGA VI must have front panel controls.

- 1. <u>Open a reference</u> to the FPGA VI or bitfile.
 - Note The FPGA target, FPGA VI, and host VI must be in the same LabVIEW project if you want to open a reference to an FPGA VI. The host VI does not need to be in a project if you open a reference to a bitfile.
- 2. Place the <u>Read/Write Control</u> function on the block diagram. Notice that the Read/Write Control function contains one **Unselected** input.

■ Place ■ Find

- 3. Wire the **FPGA VI Reference Out** output of the Open FPGA VI Reference function to the **FPGA VI Reference In** input of the Read/Write Control function.
- 4. Click the **Unselected** input. The shortcut menu lists all front panel controls and indicators in the FPGA VI.
- 5. Select a control available in the FPGA VI from the shortcut menu. Notice that the **Unselected** input changes to reflect the name of the control in the FPGA VI.

To write to more controls in the FPGA VI, right-click the Read/Write Control function and select **Add Element** from the shortcut menu and then customize the input as described in the previous step. You also can click the bottom line of the Read/Write Control function with the Positioning tool and drag the line down to add more controls and indicators. The read and write operations execute sequentially from top to bottom. Also, if the FPGA VI has subVIs and you want to access controls and indicators on the subVI, you must wire the subVI controls and indicators to the controls and indicators of the FPGA VI that is used in the Open FPGA VI Reference function.

 $\overline{\mathbb{N}}$

Note You also can write to FPGA VI indicators. To write to an indicator, right-click the indicator output and select **Change to Write** from the shortcut menu.

Reading DMA FIFOs from Host VIs (FPGA Interface)

You can <u>create Direct Memory Access (DMA) FIFOs</u> in FPGA VIs to transfer data from FPGA VIs to host VIs. Some FPGA targets do not support DMA. The FPGA targets that support DMA include a fixed number of DMA channels available for transferring data between the FPGA VI and the host VI. Refer to the specific FPGA target <u>hardware</u> <u>documentation</u> for information about the number of DMA channels available, if the FPGA target supports DMA.

Complete the following steps to read a DMA FIFO in an FPGA VI.

- 1. <u>Open a reference</u> to an FPGA VI or bitfile.
 - Note The FPGA target, FPGA VI, and host VI must be in the same LabVIEW project if you want to open a reference to an FPGA VI. The host VI does not need to be in a project if you open a reference to a bitfile. If you open a reference to an FPGA VI, the project must include a DMA FIFO item under the FPGA target and the FPGA VI must include a <u>FIFO Write</u> function on the block diagram that writes to the DMA FIFO item.
- 2. Place an <u>Invoke Method</u> function on the block diagram of the host VI in the data flow where you want the host VI to read the DMA FIFO. Make sure the host VI runs the FPGA VI before you read the DMA FIFO. Wire the **FPGA VI Reference In** input.

Place Find

- 3. Click the Invoke Method function and select *FIFO*»Read from the shortcut menu, where *FIFO* is the name of the FIFO item in the project. Wire the inputs and outputs as needed. The Read method returns **Data** when the **Number of Elements** is available or when the **Timeout** period ends. If the **Timeout** period ends before the **Number of Elements** is available, **Data** will be empty. The **Elements Remaining** output contains the number of elements remaining in the host memory part of the DMA FIFO.
- 4. Place the <u>Close FPGA VI Reference</u> function on the block diagram.

■ Place ■ Find

- 5. Wire the FPGA VI Reference Out output on the Invoke Node to the FPGA VI Reference In input on the Close FPGA VI Reference function.

Note You can read DMA FIFOs using only the Invoke Method function with the Read method. If you want more control over the DMA FIFO from the host VI, you also can configure, start, and stop the DMA FIFO using the optional Configure, Start, and Stop methods with the Invoke Method function.

Writing to DMA FIFOs from Host VIs (FPGA Interface)

You can <u>create Direct Memory Access (DMA) FIFOs</u> to transfer data from host VIs to FPGA VIs. Some FPGA targets do not support DMA. The FPGA targets that support DMA include a fixed number of DMA channels available for transferring data between the host VI and FPGA VI. Refer to the specific FPGA target <u>hardware documentation</u> for information about the number of DMA channels available, if the FPGA target supports DMA.

Complete the following steps to write to a DMA FIFO in an FPGA VI.

- 1. <u>Open a reference</u> to an FPGA VI or bitfile.
 - Note The FPGA target, FPGA VI, and host VI must be in the same LabVIEW project if you want to open a reference to an FPGA VI. The host VI does not need to be in a project if you open a reference to a bitfile. If you open a reference to an FPGA VI, the project must include a DMA FIFO item under the FPGA target and the FPGA VI must include a FIFO Read function on the block diagram that reads the DMA FIFO item.
- 2. Place an <u>Invoke Method</u> function on the block diagram of the host VI in the data flow where you want the host VI to write to the DMA FIFO. Wire the **FPGA VI Reference In** input.

🛯 Place 🔳 Find

- 3. Click the Invoke Method function and select *FIFO*»Write from the shortcut menu, where *FIFO* is the name of the FIFO item in the project. Wire the inputs and outputs as needed. The Write method returns **Empty Elements Remaining** when the data is written or when the **Timeout** period ends.
- 4. Place the <u>Close FPGA VI Reference</u> function on the block diagram.

Place Find

5. Wire the **FPGA VI Reference Out** output on the Invoke Node to the **FPGA VI Reference In** input on the Close FPGA VI Reference function.



Note You can write DMA FIFOs using only the Invoke Method function with the Write method. If you want more control over the DMA FIFO from the host VI, you also can <u>configure</u>, <u>start</u>, and <u>stop</u> the DMA FIFO using the optional Configure, Start, and Stop methods with the Invoke Method function.

Using SubVIs in Host VIs (FPGA Interface)

You can use subVIs in host VIs that communicate with FPGA VIs or bitfiles on FPGA targets. However, if you change the configuration of the <u>Open FPGA VI Reference</u> function or the <u>Up Cast</u> function, LabVIEW does not propagate the changes to the subsequent subVIs in the data flow unless you bind the reference output of the function to a type <u>definition</u>.

Complete the following steps to bind the reference output of the Open FPGA VI Reference function to a type definition.

- 1. Right-click the Open FPGA VI Reference function on the host VI block diagram and select **Configure Open FPGA VI Reference** from the shortcut menu.
- 2. Place a checkmark in the **Bind FPGA host reference to type definition** checkbox. The **Save As** dialog box appears.
- 3. Type a name for a new type definition in the **File name** text box or navigate to an existing type definition in the **Save As** dialog box. If you type a name for the type definition in the **File name** text box, LabVIEW creates the type definition for you.
- 4. Click the **Save** button to close the **Save As** dialog box.
- 5. Click the **OK** button to close the **Configure Open FPGA VI Reference** dialog box.

Complete the following steps to bind the reference output of the Up Cast function to a type definition.

- 1. Right-click the Up Cast function on the host VI block diagram and select **Bind Reference Output to Type Definition** from the shortcut menu. The **Save As** dialog box appears.
- 2. Type a name for a new type definition in the **File name** text box or navigate to an existing type definition in the **Save As** dialog box. If you type a name for the type definition in the **File name** text box, LabVIEW creates the type definition for you.
- 3. Click the **Save** button.

LabVIEW now automatically updates the type definition you created whenever changes to the Open FPGA VI Reference function or the FPGA VI necessitate changes to the type definition. LabVIEW also propagates these changes to any subVIs that use this type definition. This propagation allows the subVIs to remain consistent with the Open FPGA VI Reference function or Up Cast function.

If you do not bind the output of the Open FPGA VI Reference or Up Cast functions to a type definition and you make a change to the configuration of either of these functions, you must make the same change in any subVIs that use the reference output. Use the <u>Configure FPGA VI</u> <u>Reference</u> dialog box to update the reference.

Synchronizing FPGA VIs and Host VIs Using Interrupts (FPGA Interface)

Some FPGA targets allow you to generate interrupts from the FPGA VI to notify the host VI of events, such as data being ready, an error occurring, or a task finishing. Refer to the specific FPGA target hardware documentation for information about whether the FPGA target supports interrupts.

Waiting for and Acknowledging a Single Interrupt

Complete the following steps to wait for and acknowledge a single interrupt in a host VI.

- 1. <u>Open a reference</u> to the FPGA VI or bitfile that generates interrupts.
- Place the <u>Invoke Method</u> function on the block diagram of the host VI in the data flow where you want the host VI to wait for interrupts from the FPGA VI. Be sure to wire the FPGA VI Reference In input.

- 3. Right-click the Invoke Method function and select **Method**»Wait on IRQ from the shortcut menu.
- 4. Right-click the **IRQ Number(s)** input on the Invoke Method function and select **Create»Constant** from the shortcut menu. You also can create a control.
- 5. Enter the value of the logical interrupt you selected in the FPGA VI.
- 6. Wire the **Timeout** input if you want to specify the maximum time the host VI waits for the interrupt before continuing the data flow. By default, the host VI does not wait for an interrupt to occur and returns only interrupts already set by the FPGA VI. You also can wire a constant with a value of -1 to wait indefinitely for an interrupt. If you use the **Timeout** input, you can use the **Timed Out** output to determine whether the host VI continued data flow due to the timeout occurring or to the receipt of an interrupt. If a timeout occurs, the **Timed Out** output returns TRUE.
- Right-click the IRQ(s) Asserted output on the Invoke Method function and select Create»Indicator from the shortcut menu. LabVIEW creates a numeric indicator. A value of –1 indicates that the interrupt was not received.
- 8. Place the Invoke Method function on the block diagram of the host VI in the data flow where you want the host VI to acknowledge the interrupt from the FPGA VI. Place the Invoke Method function in a <u>Case Structure</u> if you want the function to execute only when the host VI receives an interrupt. If you wire a

Boolean constant of TRUE to the **Wait Until Cleared** input of the Interrupt VI, place the Invoke Method function in the data flow where you want to tell the Interrupt VI to stop waiting. Be sure to wire the **FPGA VI Reference In** input of the Invoke Method function.

- Right-click the Invoke Method function and select Method»Acknowledge IRQ from the shortcut menu. Use the Acknowledge IRQ method to acknowledge the logical interrupt returned by the Wait on IRQ method.
- 10. Wire the **IRQ(s)** Asserted output of the Wait on IRQ method directly to the **IRQ Number(s)** input of the Acknowledge IRQ method.

Waiting for and Acknowledging Multiple Interrupts

Complete the following steps to wait for and acknowledge multiple interrupts in a host VI.

- 1. <u>Open a reference</u> to the FPGA VI or bitfile that generates interrupts.
- Place the <u>Invoke Method</u> function on the block diagram of the host VI in the data flow where you want the host VI to wait for interrupts from the FPGA VI. Be sure to wire the FPGA VI Reference In input.

■ Place ■ Find

- 3. Right-click the Invoke Method function and select **Method**»Wait on IRQ from the shortcut menu.
- 4. Place an <u>Array Constant</u> on the block diagram. An array includes an index display on the left, an element display on the right, and an optional label.

■ Place ■ Find

5. Place a <u>Numeric Constant</u> in the array.

- 6. Using the Positioning tool, expand the array constant to the number of interrupts needed.
- 7. Enter the values of the logical interrupts.
- 8. Wire the array constant to the **IRQ Number(s)** input.
- 9. Wire the **Timeout** input if you want to specify the maximum time the host VI waits for the interrupt before continuing the data flow. By default, the host VI does not wait for an interrupt to occur and returns only interrupts already set by the FPGA VI. You also can wire a constant with a value of -1 to wait indefinitely for an interrupt. If you use the **Timeout** input, you can use the **Timed Out** output to determine whether the host VI continued data flow due to the timeout occurring or to the receipt of an interrupt. If a timeout occurs, the **Timed Out** output returns TRUE.
- 10. Right-click the IRQ(s) Asserted output on the Invoke Method

function and select **Create»Indicator** from the shortcut menu. LabVIEW creates an array indicator. The **IRQ(s)** Asserted front panel indicator displays the number(s) of the interrupts the FPGA target asserts. An empty array indicates that no interrupts were received.

11. Place the Invoke Method function on the block diagram of the host VI in the data flow where you want the host VI to acknowledge the interrupts from the FPGA VI. Place the Invoke Method function in a <u>Case Structure</u> if you want the function to execute only when the host VI receives an interrupt. If you wire a Boolean constant of TRUE to the **Wait Until Cleared** input of the Interrupt VI, place the Invoke Method function in the data flow where you want to tell the Interrupt VI to stop waiting. Be sure to wire the **FPGA VI Reference In** input of the Invoke Method function.

- 12. Right-click the Invoke Method function and select **Method**»Acknowledge IRQ from the shortcut menu. Use the Acknowledge IRQ method to acknowledge the logical interrupts returned by the Wait on IRQ method.
- 13. Wire the **IRQ(s)** Asserted output of the Wait on IRQ method directly to the **IRQ Number(s)** input of the Acknowledge IRQ method.

Forcing an FPGA VI to Download to an FPGA Target (FPGA Interface)

LabVIEW automatically downloads a compiled FPGA VI to the FPGA target when you invoke the <u>Open FPGA VI Reference</u> function. LabVIEW also automatically compiles, downloads, and runs an FPGA VI on the FPGA target when you click the **Run** button in an FPGA VI if the FPGA target supports <u>Interactive Front Panel Communication</u>. LabVIEW does not download the FPGA VI if the VI is already on the FPGA target or if the FPGA is reserved for other purposes.

You can force a download by right-clicking the FPGA VI in the <u>Project</u> <u>Explorer</u> window and selecting **Download** from the shortcut menu. FPGA target functionality varies depending on the specific target. Refer to the specific FPGA target <u>hardware documentation</u> for information about downloading options available for the target.

You also can programmatically force LabVIEW to download an FPGA VI or bitfile to an FPGA target from a host VI.

Complete the following steps to programmatically force an FPGA VI or bitfile to download.

- 1. <u>Open a reference</u> to the FPGA VI or bitfile.
 - Note The FPGA target, FPGA VI, and host VI must be in the same LabVIEW project if you want to open a reference to an FPGA VI. The host VI does not need to be in a project if you open a reference to a bitfile.
- 2. Place an <u>Invoke Method</u> function on the block diagram.

Place Find

- 3. Wire the **FPGA VI Reference Out** parameter of the Open FPGA VI Reference function to the **FPGA VI Reference In** parameter of the Invoke Method function.
- 4. Right-click the Invoke Method function and select **Method**»**Download** from the shortcut menu.

If you programmatically download the FPGA VI to the FPGA target with the Invoke Method function, you also must programmatically run the FPGA VI on the FPGA target with the Invoke Method function. Place another Invoke Method function on the block diagram and select **Method»Run** from the shortcut menu of the Invoke Method function to programmatically run an FPGA VI or bitfile.

Stopping, Aborting, and Resetting FPGA VIs (FPGA Interface)

If you create an FPGA VI that you want to stop and run again from a host VI, National Instruments recommends that you design the FPGA VI so that you can stop its execution by <u>setting a control</u> on the FPGA VI to a certain value. This design practice allows you to control how the FPGA VI is stopped and the values on the outputs on the FPGA target when the FPGA VI stops. You can stop the FPGA VI from the host VI by <u>writing the relevant FPGA VI control</u>. You then can run the FPGA VI by using the Invoke Method function with the <u>Run method</u>. If you do not need control of the FPGA target outputs, you can use the Invoke Method function with the <u>Reset method</u> to abort and restore the FPGA VI to the default state.

If stopping is not designed as part of the FPGA VI, you can reset and/or abort the FPGA VI from the host VI using the <u>Reset method</u> or <u>Abort</u> <u>method</u>. The Abort method does not reset the FPGA VI to the default state which might affect how the FPGA VI reacts to future executions of the Run method from the host VI.