Getting Started with the NI LabVIEW™ Embedded Module for ADI Blackfin Processors

Version 2.5

The NI LabVIEW Embedded Module for ADI Blackfin Processors is a comprehensive graphical development environment for embedded design. Jointly developed by Analog Devices and National Instruments, this module seamlessly integrates the LabVIEW development environment and Blackfin embedded processors.

This module builds on NI LabVIEW Embedded technology, which facilitates dataflow graphical programming for embedded systems and includes hundreds of analysis and signal processing functions, integrated I/O, and an interactive debugging interface. With the Embedded Module for Blackfin Processors, you can enable cache, optimize linking, and view live front panel updates using JTAG, serial, or TCP/IP, as well as use VisualDSP++ compiler options through LabVIEW. The Embedded Module for Blackfin Processors includes the LabVIEW C Code Generator, which generates C code from the LabVIEW block diagram.

Engineers and scientists can lower development costs, achieve faster development times, and still deliver a high performance embedded processing solution with the Embedded Module for Blackfin Processors.

August 2007
371656C-01

NATIONAL INSTRUMENTS™

Contents

What's New in the Embedded Module for Blackfin Processors

The Embedded Module for Blackfin Processors 2.5 includes the following new features:

- **VisualDSP++ 5.0 support**—Incorporates new features and enhancements to the VisualDSP++ toolchain.
- **New Blackfin processor support**—Allows you to target the ADSP-BF542, ADSP-BF544, and ADSP-BF548, in addition to the ADSP-BF53*x* processors.

- **New device drivers**—Provide support for the following:
  - Real-Time Clock (RTC), Timer, Flag, and File System services—VIs are located on the **Blackfin»Device»Services** palette.
  - BF548 EZ-Kit Lite Peripherals—VIs for the keypad, LCD, rotary controller, and USB services are located on the **Blackfin»Device»BF54x** palette.
  - AD7877 Touch Screen—VIs are located on the **Blackfin»Device»Peripherals»AD7877** palette.
- **Elemental I/O support**—Takes advantage of the Elemental I/O framework to access analog input, digital I/O, and PWM abilities on the Blackfin target.
- **VI-specific C code generation options**—Allow you to set code generations options on a per VI basis. You can specify VI-level C code generation options either through the **C Code Generation Properties** page in the **VI Properties** dialog box or the **Source File Settings** page in the **Build Specification Properties** dialog box. Refer to the *LabVIEW Help* for information about specifying VI-level C code generation options.
- **Expression folding**—Generates better performing and more efficient code by enabling expression folding, which collapses groups of nodes into single expressions that are easily recognized by C compilers. To enable expression folding for your target, place a checkmark in the **Use expression folding** checkbox in the **Build Specification Properties** dialog box.

After you install LabVIEW, refer to the *LabVIEW Upgrade Notes*, available by selecting **Start»All Programs»National Instruments»LabVIEW»LabVIEW Manuals** and opening `LV_Upgrade_Notes.pdf`, for information about new features in LabVIEW 8.5. Refer to the *LabVIEW 8.5 Readme for Windows*, available by selecting **Start»All Programs»National Instruments»LabVIEW»Readme** and opening `readme.html` for information about features and changes in LabVIEW 8.5.

System Requirements

The Embedded Module for Blackfin Processors has the following requirements:

- A desktop computer with Windows Vista/XP/2000
- Analog Devices VisualDSP++ 5.0 (included)
- LabVIEW 8.5 with embedded support (included)
- One of the following EZ-KITs:
  - EZ-KIT Lite for Analog Devices ADSP-BF537 Blackfin Processor (included)
  - EZ-KIT Lite for Analog Devices ADSP-BF533 (not included)
  - EZ-KIT Lite for Analog Devices ADSP-BF538 (not included)
  - EZ-KIT Lite for Analog Devices ADSP-BF548 (not included)

Refer to the *LabVIEW Release Notes*, available by selecting **Start»All Programs»National Instruments»LabVIEW»LabVIEW Manuals** and opening `LV_Release_Notes.pdf`, for more information about LabVIEW Development System requirements.

Installing the Embedded Module for Blackfin Processors

If you have LabVIEW 8.5 installed, you can install LabVIEW 8.5 with embedded support without first uninstalling LabVIEW 8.5. However, you must install VisualDSP++ 5.0 before you install the Embedded Module for Blackin Processors.

Complete the following steps to install VisualDSP++ 5.0 and the Embedded Module for Blackfin Processors.

1. Log in as an administrator or as a user with administrator privileges.
2. Insert the NI LabVIEW Embedded Module for ADI Blackfin Processors Installation DVD and select to install ADI VisualDSP++ 5.0.

Tip If the installer does not automatically begin, double-click `VisualDSP++5.0.exe` on the DVD to begin installation of VisualDSP++ 5.0.

3. Follow the instructions on the screen for installing VisualDSP++ 5.0.

   When the VisualDSP++ 5.0 installation finishes, you can install the Embedded Module for Blackfin Processors.

4. Select to install the Embedded Module for Blackfin Processors. If the installer welcome screen is not visible, double-click `setup.exe` on the DVD to begin installation of the Embedded Module for Blackfin Processors.
5. Follow the instructions on the screen. The installation DVD installs both LabVIEW 8.5 with embedded support and the Embedded Module for Blackfin Processors.
6. Follow the activation instructions that appear on the screen.

   You also can use the NI License Manager, available by selecting **Start»All Programs»National Instruments»NI License Manager**, to activate National Instruments products. Refer to the *National Instruments License Manager Help*, available by selecting **Help» Contents** in the NI License Manager, for more information about activating NI products.
7. Restart the computer when the installer prompts you and log on as an administrator or as a user with administrator privileges.

Installing the EZ-KIT Lite or Emulator

Caution Be careful when removing the board from the package and handling the board to avoid the discharge of static electricity, which might damage some components.

The EZ-KIT Lite or USB-based ICE board is designed to run as a stand-alone unit. You do not have to open the computer case.

Note You must install VisualDSP++ before you can install the Blackfin target. Refer to the *Installing the Embedded Module for Blackfin Processors* section for information about installing VisualDSP++.

Complete the following steps to install the EZ-KIT Lite or USB-based ICE.

1. Plug the power supply for the board into a surge-protected outlet. Connect the USB assembly for the board to the USB port on the host computer using the provided USB cable.

   Figure 1 shows the location of the A/C adaptor and USB port on the Blackfin target. Refer to the *ADSP-BF537 EZ-KIT Lite Evaluation System Manual* in the EZ-KIT box for more detailed information about the ADSP-BF537 EZ-KIT Lite hardware.

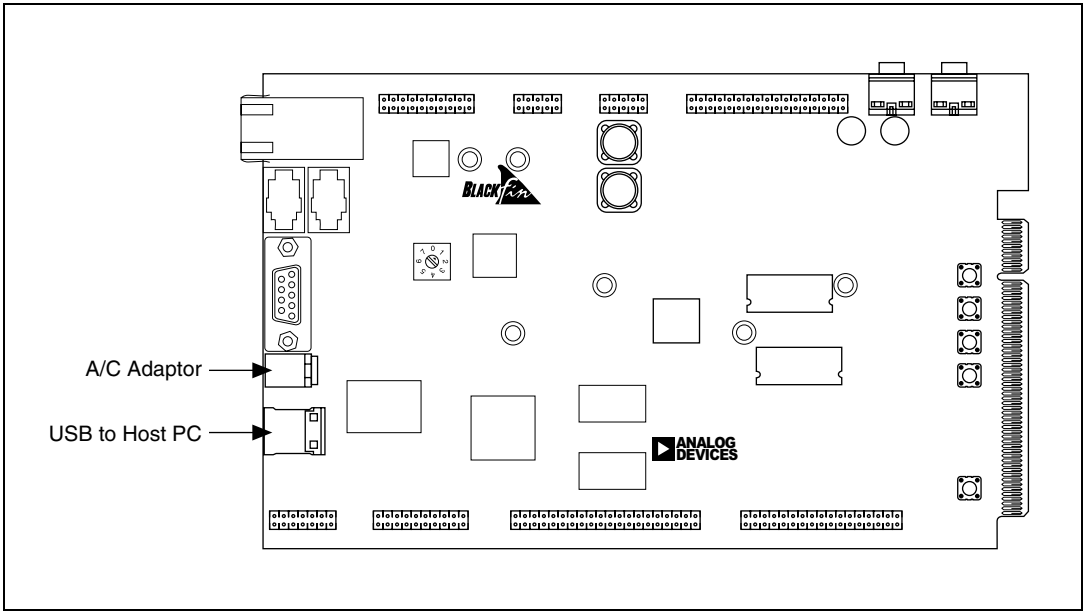

Figure 1. Locating the A/C Adaptor and USB Port on the Blackfin Target

On the board, the power LED illuminates, and you might see other visible activity, such as blinking LEDs. The connection activates the Windows Found New Hardware Wizard.

2. Follow the instructions on the screen to install the software automatically. A Windows message notifies you when the new device is ready for use and the hardware installation is complete.
3. Verify that the USB monitor LED is lit. The LED is in close proximity to the USB connector. The lit LED signifies that the board is communicating properly with the host computer and is ready to run. Refer to the *ADSP-BF537 EZ-KIT Lite Evaluation System Manual* in the EZ-KIT box for more detailed information about the USB monitor LED.
4. For a USB-based ICE, attach the JTAG cable to the emulation target. Refer to the emulator documentation for more information about USB-based ICE.

Upgrading from the Embedded Module for Blackfin Processors 2.0

Blackfin VIs and projects you created using the Embedded Module for Blackfin Processors 2.0 are compatible with the Embedded Module for Blackfin Processors 2.5.

Upgrading from the Embedded Module for Blackfin Processors 1.0

This section describes upgrade and compatibility issues for upgrading from the Embedded Module for Blackfin Processors 1.0 to 2.5.

Embedded Project Manager

The Embedded Module for Blackfin Processors no longer supports the Embedded Project Manager. Use the **Project Explorer** window instead. Refer to the *Creating the LabVIEW Project* section for information about using the **Project Explorer** window. Refer to the *Upgrading Blackfin VIs and Embedded Projects* section for information about converting embedded project (`.lep`) files.

Upgrading Blackfin VIs and Embedded Projects

If you created Blackfin VIs using the Embedded Module for Blackfin Processors 1.0, you must create projects and update the VIs for use with LabVIEW 8.5 and the Embedded Module for Blackfin Processors 2.5.

Caution National Instruments recommends you back up your files before you begin updating your Blackfin VIs for use with the Embedded Module for Blackfin Processors 2.5. The Embedded Module for Blackfin Processors 1.0 cannot open VIs you save in version 2.5.

The Embedded Module for Blackfin Processors includes a utility to help you convert LabVIEW 7.1 embedded project (`.lep`) files to LabVIEW 8.5 project (`.lvproj`) files. Complete the following steps to update embedded project files for use with LabVIEW 8.5 and the Embedded Module for Blackfin Processors 2.5.

1. Launch LabVIEW 8.5.
2. Select **File»Open** in the **Getting Started** window and navigate to the `labview\Targets\ADI\Embedded\vdk\utils\Blackfin LEP Converter\` directory.

3. Select `Blackfin_LEP_Converter.vi` and click the **OK** button to run the Blackfin LEP converter utility.
4. 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 Blackfin LEP converter utility. You can see only the LEP file in the **LabVIEW Embedded Project (*.lep) Files** list.
5. Click the **Continue** button. LabVIEW prompts you to specify the location and name of the new project.
6. The **Import Status** dialog box appears. When LabVIEW finishes importing the 1.0 files, the **Project Explorer** window appears with the Blackfin target and VIs.
7. Complete the following steps to verify the target configuration and build specifications.
    a. Right-click the Blackfin target in the **Project Explorer** window and select **Configure Target** from the shortcut menu.
    b. Verify the settings in the **VisualDSP++ Target Configuration** dialog box and click the **OK** button. Refer to the *Configuring the Target and Debugging Options* section for information about configuring the Blackfin target.
    c. Right-click the build specification under the Blackfin target in the **Project Explorer** window and select **Properties** from the shortcut menu to open the **Build Specification Properties** dialog box.
    d. Verify the settings in the **Application Information** and **Source Files** categories and click the **OK** button. Refer to the *Creating the ADSP-BF537 Build Specification* section for information about editing a build specification.
8. Select **File»Save** in the **Project Explorer** window to save the project. LabVIEW saves a copy of all the VIs for the project in the same folder as the `.lvproj` file.

Refer to the *Building, Downloading, and Running a Blackfin Application* section for information about building a Blackfin VI into a Blackfin application and running the application on the target.

Migrating Custom Linker Description Files

If you used previous version of the Embedded Module for Blackfin Processors 1.0 to move to custom hardware, you modified the Linker Description File (LDF) to match the hardware layout of the custom hardware memory layout. The `.ldf` file defines the system to the VisualDSP++ linker and specifies how the linker creates executable code.

You must rewrite custom `.ldf` files to use them with the Embedded Module for Blackfin Processors 2.5 because the format of `.ldf` files changed from previous versions of VisualDSP++ to VisualDSP++ 5.0.

Refer to the *Moving to Custom Hardware* topic in the *LabVIEW Help,* available by selecting **Help»Search the LabVIEW Help** in LabVIEW, for information about modifying `.ldf` files for a custom target.

Tip When searching for a specific topic in the *LabVIEW Help* from the **Search** tab, place a checkmark in the **Search titles only** checkbox and enter the topic title in the **Type in the word(s) to search for** text box. This option limits the number of topics that the *LabVIEW Help* finds so you can locate the topic more quickly.

Getting Started with Blackfin VIs, Applications, and Elemental I/O Tutorial

Use this example to learn how to create a LabVIEW project and build, run, and debug a Blackfin application.

Note You can create a project and Blackfin VI without connecting a Blackfin target to the host computer. However, you must connect a Blackfin target to the host computer before you can test a Blackfin application. Refer to the *Installing the EZ-KIT Lite or Emulator* section for information about installing the Blackfin target.

Creating the LabVIEW Project

Use LabVIEW projects to group together LabVIEW files and non-LabVIEW files, create build specifications for building a Blackfin VI into a Blackfin application, and run the application on a Blackfin target. You must use a project to build a Blackfin VI into a Blackfin application.

LabVIEW project files have a `.lvproj` file extension. Project files contain target-specific build options and other information necessary for the LabVIEW C Code Generator to generate C code from the VIs.

Complete the following steps to launch LabVIEW and create a project with an ADSP-BF537 target and a blank VI.

1. Launch LabVIEW.
2. Select **Empty Project** in the **Getting Started** window to open an empty LabVIEW project as shown in Figure 2. LabVIEW creates an empty project with a Windows target, shown in Figure 2 as **My Computer**.

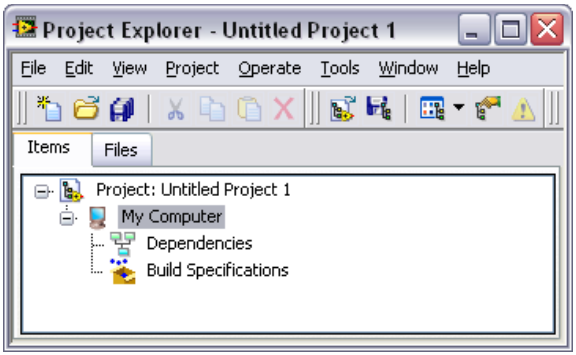

Figure 2. Targeting My Computer in the Project Explorer Window

3. Right-click **Project:Untitled Project 1** in the **Project Explorer** window and select **New»Targets and Devices** from the shortcut menu to open the **Add Targets and Devices** dialog box as shown in Figure 3.

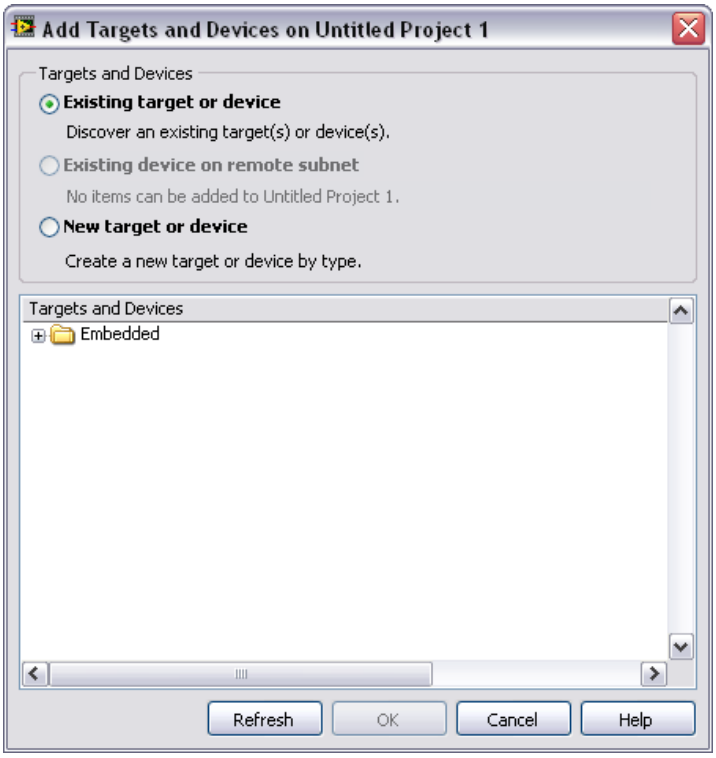

Figure 3. Add Targets and Devices Dialog Box

4. Expand the **Embedded** folder and select **Analog Devices ADSP-BF537** as shown in Figure 4.

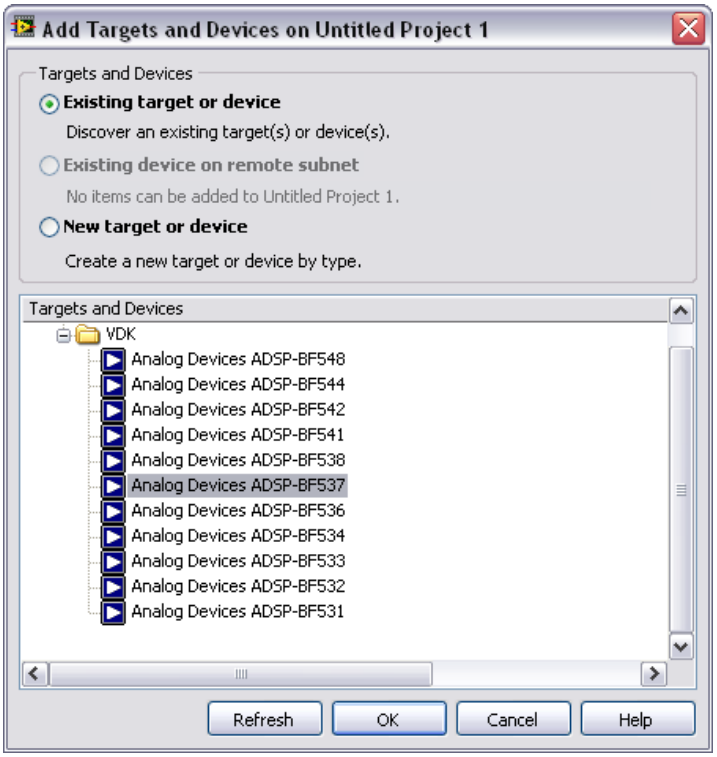

Figure 4. Adding a New Device

5. Click the **OK** button to add the target to the project. The target appears in the **Project Explorer** window as shown in Figure 5.

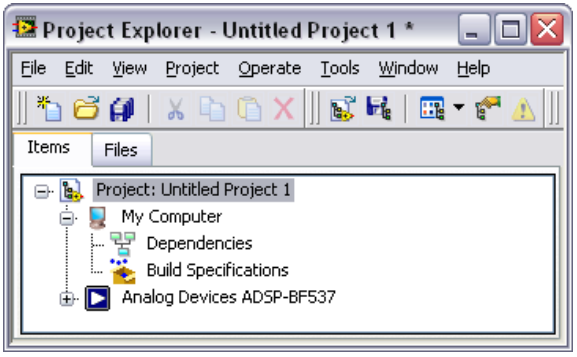

Figure 5. Adding the ADSP-BF537 to the Project Explorer Window

6. Expand the ADSP-BF537 target as shown in Figure 6. LabVIEW automatically adds **Dependencies** and **Build Specifications** under the target. SubVIs appear under **Dependencies** when you add a VI that contains subVIs to a project. Build specifications you create for a target in a project appear under **Build Specifications** in the **Project Explorer** window. Refer to the *Creating the ADSP-BF537 Build Specification* section for information about how to create a build specification.

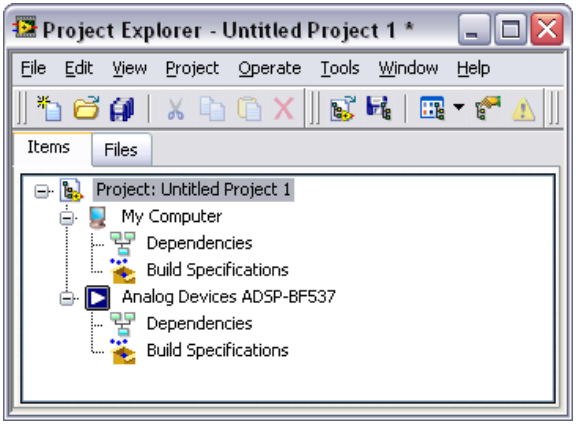

Figure 6. Expanding the Blackfin Target in the Project Explorer Window

7. Right-click the ADSP-BF537 target and select **New»VI** from the shortcut menu. The new, untitled VI appears in the **Project Explorer** window under the ADSP-BF537 target as shown in Figure 7.

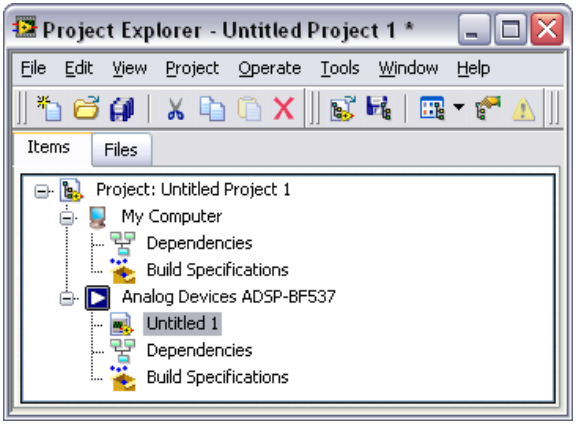

Figure 7. Adding a New VI to the Project

The front panel window and block diagram of the new VI open. The front panel window, or user interface, appears with a gray background and includes controls and indicators. The block diagram appears with a white background and includes VIs, functions, and structures that control the front panel window objects.

Tip You also can use the New Blackfin Project Wizard to create a project with a Blackfin target. Refer to the *Creating Blackfin Projects* book in the *LabVIEW Help* for more information about creating a project with a Blackfin target.

Creating the Front Panel

The front panel is the user interface for a VI. You can use the front panel as a debugging interface for Blackfin applications you create with LabVIEW. This tutorial includes an LED indicator that lights if the input exceeds a threshold value you define.

Complete the following steps to create the front panel.

1. Add the following controls to the front panel window as shown in Figure 8:
   - Two numeric controls located on the **Numeric** palette.
   - One numeric indicator located on the **Numeric** palette.
   - One round LED located on the **Boolean** palette.

Tip If you cannot find the object you are looking for, click the **Search** button on the **Controls** or **Functions** palette toolbar. Type the name of the object for which you want to search. LabVIEW searches as you type and displays any matches in the search results text box.

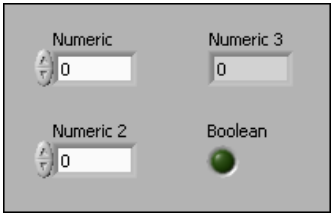

Figure 8. Creating the Front Panel

2. Rename the controls by double-clicking the labels and entering new names as shown in Figure 9.
   - Change one of the numeric controls to **input**.
   - Change the other numeric control to **threshold**.
   - Change the numeric indicator to **output**.
   - Change the round LED to **threshold exceeded?**.

Tip Double-click to select a single word in a label. Triple-click to select the entire label.

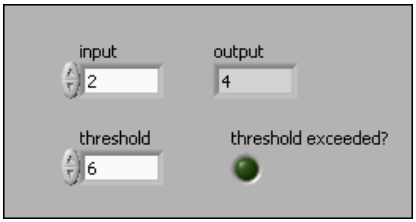

Figure 9. Changing the Labels

3. Change the numeric representation of the numeric controls and the **output** numeric indicator to a 32-bit signed integer by right-clicking each control and indicator and selecting **Representation»I32** from the shortcut menu. By default LabVIEW creates numeric controls as a double precision, floating-point data type. However, the Blackfin processors must emulate floating-point data types. so you can make the Blackfin application more efficient by changing the representation of the numeric controls to an integer-based data type.

Creating the Block Diagram

The block diagram is the source code for a VI and contains a pictorial description or representation of an application. Wires carry data between the objects, or nodes, on the block diagram. The controls and indicators you added in the *Creating the Front Panel* section appear as terminals on the block diagram.

Complete the following steps to create the block diagram.

1. Switch to the block diagram by clicking the block diagram if it is visible or selecting **Window»Show Block Diagram**.

Tip You also can switch to the block diagram by pressing the <Ctrl-E> keys.

2. Select **Help»Show Context Help** to display the **Context Help** window. The **Context Help** window displays basic information about LabVIEW objects when you move the cursor over each object.

Tip You also can press the <Ctrl-H> keys to open and close the **Context Help** window.

3. Create the block diagram as shown in Figure 10.
   a. Place a While Loop located on the **Structures** palette around the controls and indicator on the block diagram. While Loops repeat the subdiagram inside it until the conditional terminal, which is an input terminal, receives a particular Boolean value. Right-click the conditional terminal, shown at left, in the lower right corner of the While Loop and select **Create Constant** from the shortcut menu. The default Boolean constant in the While Loop is FALSE.

   b. Place a Multiply function located on the **Numeric** palette on the block diagram inside the While Loop.
   c. Wire the **input** control to the **x** input of the Multiply function.
   d. Right-click the **y** input of the Multiply function and select **Create»Constant** from the shortcut menu. Enter `2` to multiply the value of the **input** control by two.
   e. Place a Greater? function located on the **Comparison** palette on the block diagram.
   f. Wire the **x*y** output of the Multiply function to the **x** input of the Greater? function.
   g. Wire the **threshold** control to the **y** input of the Greater? function.
   h. Wire the **x > y?** output of the Greater? function to the **threshold exceeded** indicator.
   i. Wire the **output** indicator to the wire connecting the Multiply function and the Greater? function.
   j. Place a Wait Until Next ms Multiple function located on the **Time, Dialog & Error** palette inside the While Loop.

k. Right-click the **millisecond multiple** input and select **Create»Constant** from the shortcut menu. Enter `100` to wait 100 milliseconds.

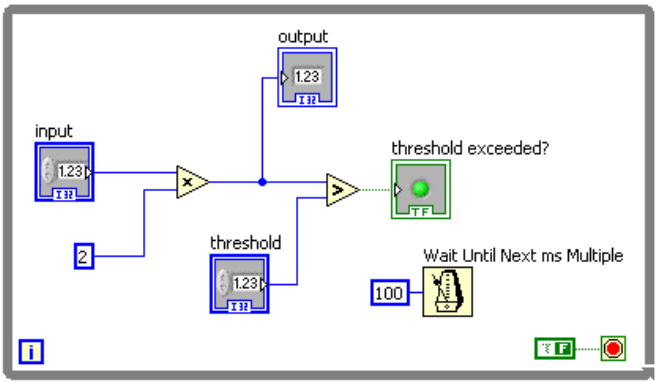

Figure 10. Creating the Block Diagram

4. Save the VI as `Blackfin Tutorial`.

Configuring the Target and Debugging Options

In LabVIEW you must specify how the Blackfin target connects to the host computer using the **VisualDSP++ Target Configuration** dialog box. The EZ-KIT Lite is connected to the host computer through the USB port, which also is known as a debug agent.

Note You only have to configure the target once unless you change how you connect the target to the host computer.

Complete the following steps to configure the target options.

1. Right-click **Analog Devices ADSP-BF537** in the **Project Explorer** window and select **Configure Target** from the shortcut menu to open the **VisualDSP++ Target Configuration** dialog box, shown in Figure 11.

The **VisualDSP++ version** pull-down menu and **VisualDSP++ location** text box display the version of VisualDSP++ on the host computer. Correct the version and location if necessary.

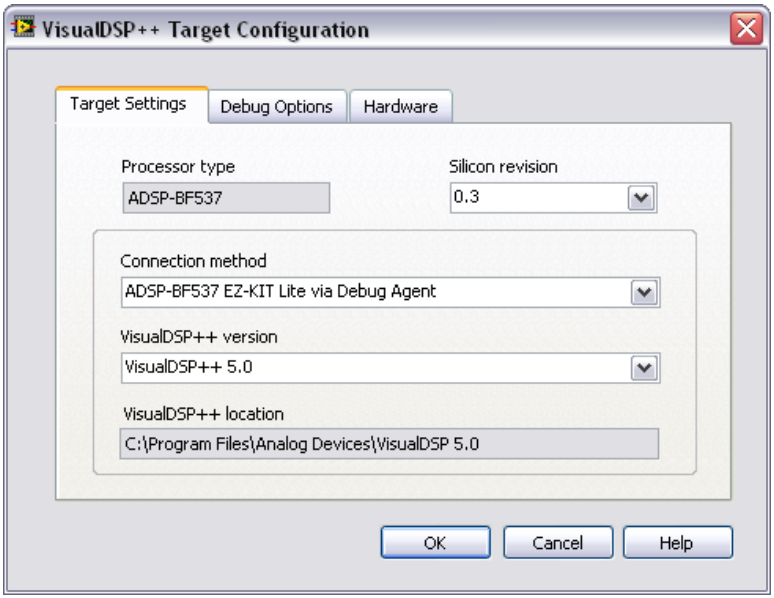

Figure 11. Configuring the Target Settings

2. Select **ADSP-BF537 EZ-KIT Lite via Debug Agent** from the **Connection method** pull-down menu.
3. Select the silicon revision in the **Silicon revision** pull-down menu that matches the silicon revision on the hardware. You can find the silicon revision on the Blackfin processor of the target.
4. Click the **Debug Options** tab to configure the debug options you can use while debugging a Blackfin application on the Blackfin target.

Tip Click the **Help** button to open the *LabVIEW Help* and read a description of each debug setting.

5. Change the **Front Panel / Probe Update Period (ms)** to 100 by moving the slider or typing `100` in the numeric control under the slider

as shown in Figure 12. This setting configures how often the front panel updates with data from the Blackfin application.

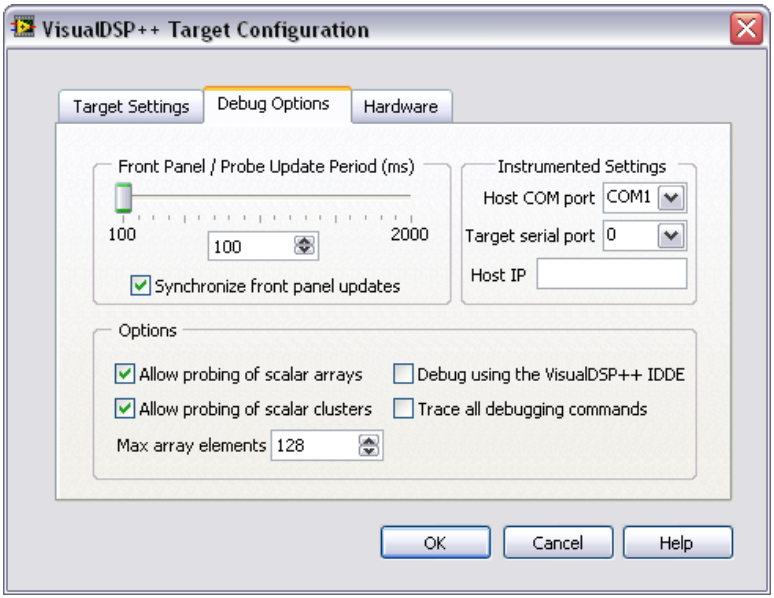

Figure 12. Configuring the Debugging Options

6. Click the **OK** button to close the **VisualDSP++ Target Configuration** dialog box.
7. Click the **OK** button in the dialog box that appears reminding you to rebuild the project.

Creating the ADSP-BF537 Build Specification

Build specifications specify how the LabVIEW C Code Generator generates the C code and how to build the Blackfin VI into a Blackfin application. You can create the build specification when you create a project or wait until you are ready to build the Blackfin VI into a Blackfin application.

You can have multiple build specifications for the same target. For example, you might want one build specification that generates debugging information and another build specification that does not generate this extra information.

Complete the following steps to create a build specification.

1. Right-click **Build Specifications** under the ADSP-BF537 target and select **New»VDK Application** from the shortcut menu.

2. LabVIEW prompts you to save the project. Click the **Save** button when prompted and save the project as `Blackfin Tutorial`. The **Build Specification Properties** dialog box opens.
3. (Optional) LabVIEW might prompt you to configure the target. Click the **Yes** button and refer to the *Configuring the Target and Debugging Options* section for information about configuring the target.
4. Select **Debug** in the **Build Configuration** section of the **General** tab. The debug build configuration does not apply any compiler optimizations, which makes the embedded application larger. Use the debug build configuration when you want C source-level debugging.
5. Select an option in the **Debug Mode** section. You can debug a Blackfin application in the following ways:
    - **Instrumented debugging using a serial port**—Single-stepping and probes are faster than non-instrumented debugging, but using a serial port requires the COM port on the Blackfin target to be connected to the host PC, is intrusive on real-time performance, and uses a larger amount of memory on the Blackfin target. The LabVIEW C Code Generator adds a communication layer to the generated C code for synchronization and data transfer. You must remove the checkmark from the **Redirect stdout to serial port** checkbox on the **Advanced** tab before you can select the serial port debug option.

Note You must connect the Blackfin target to the host computer with a serial cable to use the serial port debug option.

- **Instrumented debugging using a TCP port**—Single-stepping and probes are faster than non-instrumented debugging, but using a TCP port requires that the Blackfin target is connected to an Ethernet port, is intrusive on real-time performance, and uses a larger amount of memory on the Blackfin target. The LabVIEW C Code Generator adds a communication layer to the generated C code for synchronization and data transfer. You must place a checkmark in the **Enable lwIP TCP/IP support** checkbox on the **Advanced** tab before you can select the TCP port debug option.

Note You must connect the Blackfin target to the host computer with an Ethernet cable to use lwIP TCP/IP support.

- **Non-instrumented debugging using a JTAG/EZ-KIT USB connection**—Single-stepping and probes are slower than instrumented debugging, but using JTAG or USB does not require a network connection.

Note You still must connect the Blackfin target to the host computer using a JTAG or USB connection to download a Blackfin application to a Blackfin target, run the application on the Blackfin target, reset the processor, and so on, even if you use a serial port or TCP port for debugging.

This tutorial uses the non-instrumented debug mode as shown in Figure 13, but you can select any debug mode.

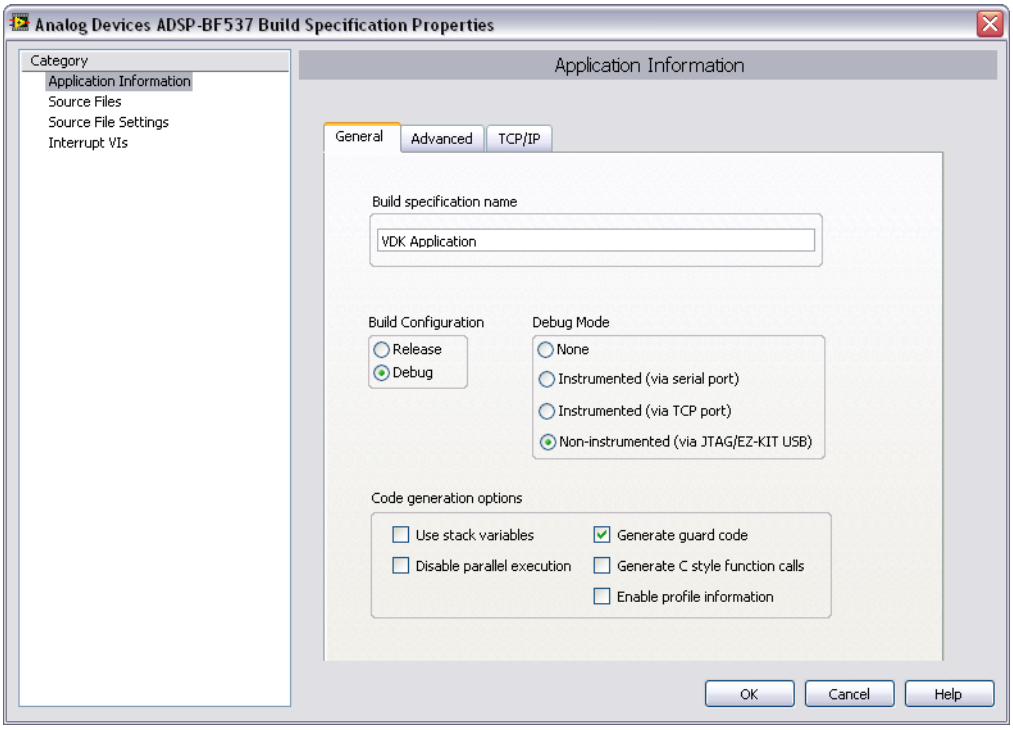

Figure 13. Configuring the Build Options

6. Select **Source Files** from the **Category** list and select **Blackfin Tutorial.vi** in the source files list. Click the blue right arrow button, shown at left, to move the VI from the source files list to the **Top-level VI** text box as shown in Figure 14.

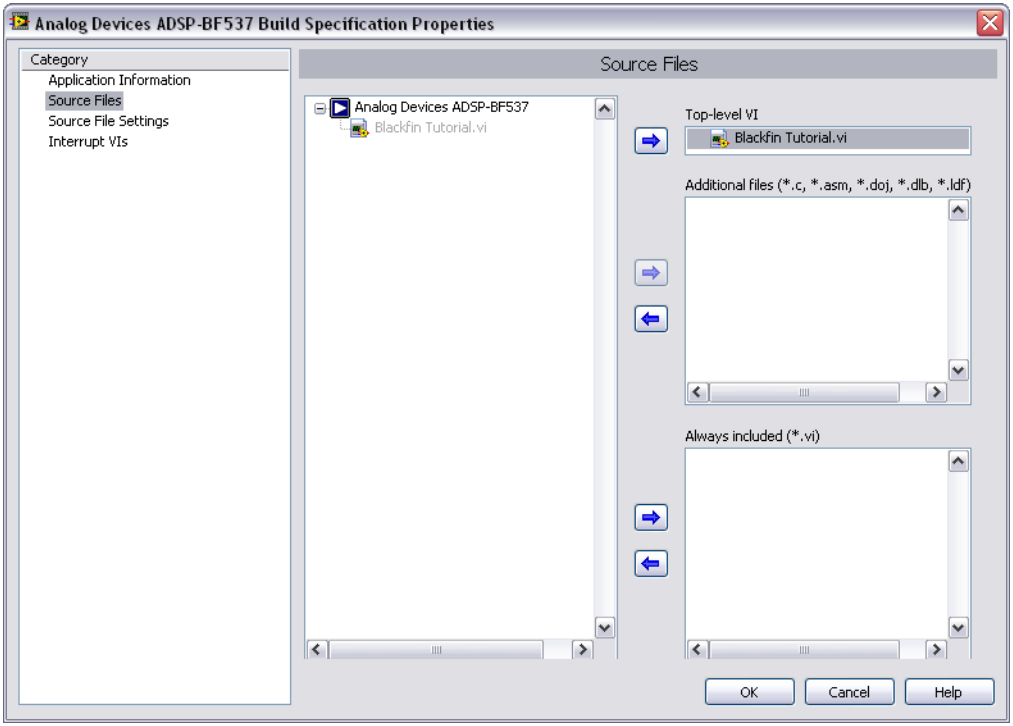

Figure 14. Selecting the Top-Level VI for the Build Specification

7. Click the **OK** button and expand the **Build Specifications** item in the **Project Explorer** window. The build specification you just created appears under the Blackfin target as shown in Figure 15.

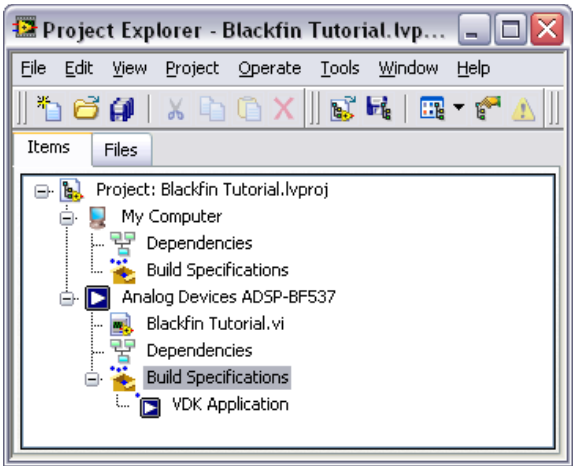

Figure 15. Blackfin Build Specification in the Project Explorer Window

8. Triple-click **VDK Application** and rename the build specification to `Debug configuration` as shown in Figure 16.

Tip You also can right-click **VDK Application** and select **Rename** from the shortcut menu to edit the build specification name.

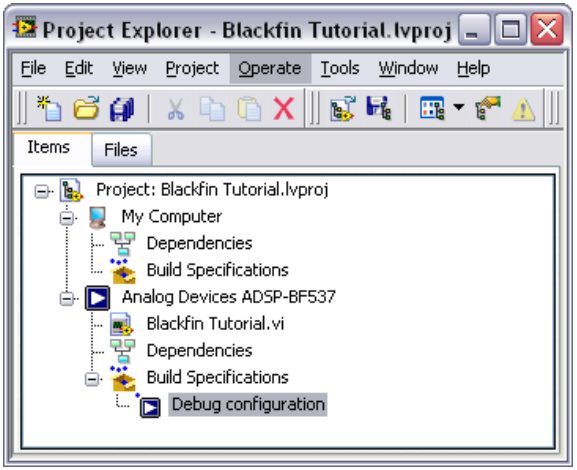

Figure 16. Renaming the Blackfin Build Specification

9. Select **File»Save Project** in the **Project Explorer** window to save the project. Build specifications are saved with the project.

Building, Downloading, and Running a Blackfin Application

After you develop the Blackfin VI on the host computer, you build the Blackfin VI into a Blackfin application you can run on a Blackfin target. When you build a Blackfin application, the LabVIEW C Code Generator generates C code from the LabVIEW block diagram using the settings you configure.

Note Before you can build, or compile, a Blackfin VI into a Blackfin application, you must configure the build specifications, the target settings, and the debug settings. Refer to the *Creating the ADSP-BF537 Build Specification* and *Configuring the Target and Debugging Options* sections for information about configuring specifications and settings.

1. Right-click **Debug configuration** in the **Project Explorer** window and select **Build** from the shortcut menu to build the Blackfin VI into a Blackfin application. LabVIEW displays the status of the building and linking process.
2. Right-click **Debug configuration** again and select **Debug** from the shortcut menu to deploy the application to the Blackfin target. The application automatically runs on the Blackfin target when you select **Debug** from the shortcut menu.
3. Enter a value in the **threshold** front panel numeric control of the Blackfin Tutorial VI on the host computer.
4. Enter different values in the **input** numeric control. In Figure 17, the front panel on the left does not exceed the threshold value. If you enter a number greater than the threshold, the **threshold exceeded?** LED lights as shown in the front panel on the right in Figure 17.

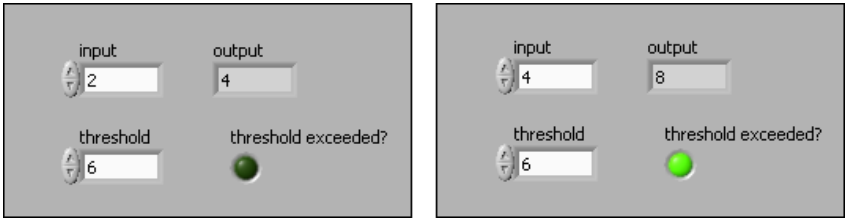

Figure 17. Increasing Input Causes Output to Exceed Threshold and Lights the LED

5. Click the **Abort Execution** button, shown at left, to stop the Blackfin application.

Debugging with Breakpoints and Probes

Complete the following steps to debug the Blackfin tutorial application with breakpoints and probes.

1. Switch to the block diagram if it is not visible.
2. Right-click the Multiply function and select **Set Breakpoint** from the shortcut menu. The breakpoint is highlighted with a red border around the function. When you run the Blackfin application, execution pauses just before the function executes. If you are using JTAG or USB/EZ-KIT for debugging, LabVIEW might prompt you to halt the processor.
3. Right-click **Debug configuration** in the **Project Explorer** window and select **Debug** from the shortcut menu. LabVIEW prompts you to save changes to the VI. LabVIEW also prompts you if you need to rebuild or redownload the Blackfin application to the Blackfin target.

Tip LabVIEW uses default values for controls and indicators when building a Blackfin VI into a Blackfin application. To change the initial values, enter the new values in the front panel controls and then select **Edit»Make Current Values Default** to change the initial values. You must rebuild the Blackfin application after you change the initial values of the controls.

The Blackfin tutorial application begins running on the Blackfin target. When you reach a breakpoint during execution, the Blackfin target halts all operation, the application pauses, and the **Pause** button, shown at left, appears red and changes to a **Continue** button.

4. Add a probe to see the values on the wire coming into the Multiply function.
   a. Click the wire coming into the **x** input.
   b. Click the wire coming into the **y** input.

   A floating **Probe** window appears after you create each probe. LabVIEW numbers the **Probe** windows automatically and displays the same number in a glyph on the wire you click as shown in Figure 18.

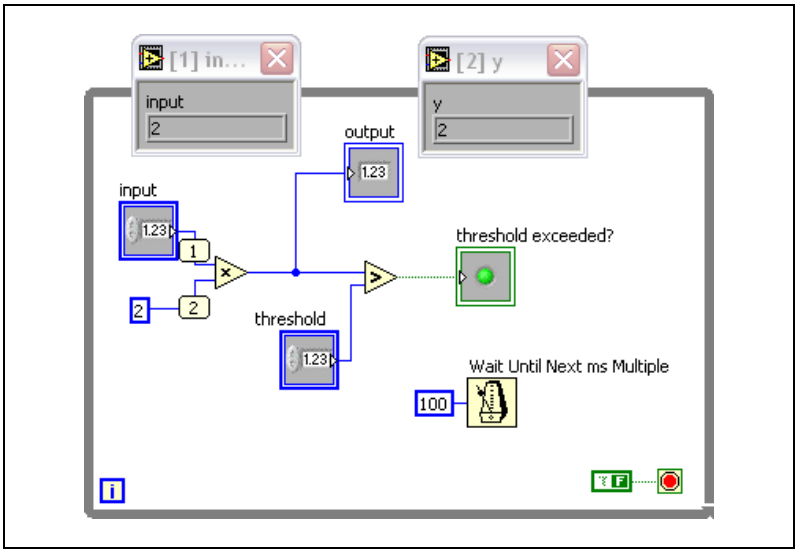

Figure 18. Creating Probes

5. Enter a different value in the **input** numeric control.
6. Click the **Continue** button, shown at left, a few times to see the value in the **Probe** windows change as the Blackfin application executes additional iterations of the While Loop.
7. Click the **Step Over** button, shown at left, to execute a node and pause at the next node. The node blinks when it is ready to execute.
8. Continue clicking the **Step Over** button to step through the rest of the block diagram.
9. Click the **Abort Execution** button to stop the application.

Using Elemental I/O

Complete the following steps to use Elemental I/O to light an LED on the Blackfin target when the threshold is exceeded.

Selecting the Elemental I/O Target

You first must select the Elemental I/O for the Blackfin target before you can add Elemental I/O items to the project. Complete the following steps to select the Elemental I/O target.

1. Right-click **Analog Devices ADSP-BF537** in the **Project Explorer** window and select **Select Elemental I/O** from the shortcut menu to open the **Select Elemental I/O** dialog box.

   Use this dialog box to specify the target I/O. After you select the I/O, you can create new Elemental I/O items in the **Project Explorer** window and use Elemental I/O in a Blackfin VI.

2. Select BF537 in the **Select Elemental I/O Device** list to select the EZ-KIT I/O and click the **OK** button. After you select the I/O and click the **OK** button, this dialog box becomes unavailable to the project. You must create a new project if you need to select a different I/O for the Blackfin target.

Adding Elemental I/O Items to the Project

You must add Elemental I/O items to the project before you can use Elemental I/O in a Blackfin VI. Complete the following steps to add Elemental I/O items to the project.

1. Right-click **Analog Devices ADSP-BF537** in the **Project Explorer** window and select **New»Elemental I/O** from the shortcut menu to open the **New Elemental I/O** dialog box.
2. Expand **Digital Output** in the **Available Resources** tree.
3. Hold down the <CTRL> key and click LED1 and LED2 to select both resources.
4. Click the **Add** button to add LED1 and LED2 to the **New Elemental I/O** list.
5. Click the **OK** button to add the Elemental I/O items to the LabVIEW project.

   Many pins on the Blackfin target can have multiple configurations. For example, on the Blackfin BF537 EZ-KIT, LED1 and the pulse width modulation (PWM) timer TMR3 both use pin PF6. Therefore, you cannot use both LED1 and TMR3 in the same application.

After you add Elemental I/O items to the project, LabVIEW filters the available resources in the **New Elemental I/O** dialog box to remove resources with pin conflicts. In this example, if you right-click **Analog Devices ADSP-BF537** and select **New»Elemental I/O** from the shortcut menu, you notice that TMR3 is not available in the **Available Resources** list because you already added LED1 to the project.

Using Elemental I/O on the Block Diagram

You can use Elemental I/O on the block diagram after you add Elemental I/O items to the project. Complete the following steps to use Elemental I/O in a Blackfin VI.

1. Drag LED1 from the Project Explorer window to the block diagram above the **threshold exceeded?** indicator.
2. Expand the Elemental I/O Node by dragging the bottom handle until you see LED1 and LED2.
3. Wire the **x > y?** output of the Greater? function to the LED1 and LED2 items in the Elemental I/O Node.

   Refer to the *Using Elemental I/O Nodes* topic in the *LabVIEW Help* for more information about using Elemental I/O Nodes.

Building and Running the Application with Elemental I/O

Complete the following steps to run the Blackfin application with Elemental I/O.

1. Save the VI so you can rebuild the project.
2. Right-click **Debug configuration** in the **Project Explorer** window and select **Rebuild All** from the shortcut menu.
3. When the application is finished rebuilding, right-click **Debug configuration** and select **Debug** from the shortcut menu.

Tip You also can click the **Pause** button and then the **Run** button in the front panel or block diagram window to run the Blackfin application in debug mode. Refer to the *Using the Run Button with Blackfin VIs* topic in the *LabVIEW Help* for more tips on using the **Run** button.

4. Enter different values in the **input** numeric control until the **threshold exceeded?** indicator lights. When the **threshold exceeded?** LED lights, LED1 and LED2 on the Blackfin target also light.
5. Click the **Abort Execution** button to stop the Blackfin application.

Where to Go from Here

National Instruments provides many resources to help you succeed with your NI products. Use the following related documentation as you start exploring LabVIEW and the Embedded Module for Blackfin Processors.

- *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help** in LabVIEW, provides information about LabVIEW programming, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools. Refer to the **Embedded Module for Blackfin Processors** book on the **Contents** tab of the *LabVIEW Help* for information specific to the Embedded Module for Blackfin Processors and Blackfin applications.
- Context help provides brief descriptions of VIs and functions with a link to the complete reference for a VI or function. Select **Help» Show Context Help** to open the **Context Help** window.
- Examples are available in the `labview\examples\lvemb\Blackfin` directory and can help you get started creating Blackfin VIs.
- The readme file, available by selecting **Start»All Programs»National Instruments»LabVIEW»Readme** and opening `readme_BLACKFIN.html`, contains known issues and last-minute information.
- The *Getting Started with ADSP-BF537 EZ-KIT Lite* manual, available in the EZ-KIT box, familiarizes you with the hardware capabilities of the EZ-KIT.
- The *ADSP-BF537 EZ-KIT Lite Evaluation System Manual*, available in the EZ-KIT box, describes the operation and configuration of the board components and provides a schematic for reference.