BCI-1553

Advanced Operation Documentation

REV	DATE	DESCRIPTION	PREPARED BY	APPROVED BY
1.0	05/18/2016	Initial revision.	J. Ryan Vallieres	-

Table of Contents

List of Figures

1 Product Description

The Bloomy Controls BCI-1553 module is a redundant multi-function MIL-STD-1553 terminal designed for use in a National Instruments cRIO chassis. The BCI-1553 module supports operation as a Remote Terminal, Dual Remote Terminals, Bus Controller, or a Bus Monitor. Redundancy on the module is handled by having up to two transmission buses for any of the operation modes, each being independently configurable. This document specifically defines the operation of the Advanced mode of the BCI-1553.

1.1 Features

1.1.1 Direct Architecture Access

The Advanced operating mode gives tools for directly accessing the Holt 6130 series chip at the core of the BCI-1553 C-Series module.

1.1.2 Fully Customizable

With direct access to the core architecture, the BCI-1553 module can be configured for a vast array of highly custom applications. If the other pre-packaged operating modes do not offer the required flexibility for an application, the Advanced operation affords even greater flexibility to completely bend the BCI-15532 module to work for any complex use.

1.1.3 Low Utilization

The Advanced operation has a very small FPGA footprint, enabling the use of the module on a greater variety of FPGA targets and in larger-scale FPGA applications.

1.2 Setup & Installation

1.2.1 How to Setup Hardware for Use

The BCI-1553 module requires several external components for interfacing and power.

1. A compatible National Instruments cRIO chassis (Minimum Suggested: Kintex 7 70T FPGA).
2. An external power source capable of providing the voltage and current outlined in section 2.1.1.

After the BCI-1553 drivers have been installed, the BCI-1553 module can be placed in an empty slot in the cRIO chassis. Power can be connected to the front of the module and cabling run from the DB-15 interface onto the 1553 Bus. Once this is complete, the module can be discovered on a LabVIEW FPGA application.

1.2.2 How to Install Software for Use

When receiving a BCI-1553 module, drivers will accompany the module via flash drive, CD, or hyperlink. These various sources will contain an installer executable that will place the required drivers onto a target computer, allowing an installed version of LabVIEW to recognize the BCI-1553 module when it is present in a cRIO chassis.

1.2.2.1 Before Installing Drivers

Before attempting to install the BCI-1553 drivers, make sure that all versions of LabVIEW are closed on the target computer and no other install processes are actively running. Further, the following applications and software are required to successfully install the BCI-1553 drivers.

1. National Instruments LabVIEW 2015.
2. National Instruments LabVIEW 2015 FPGA and Real Time modules.

1.2.2.2 Installing BCI-1553 Drivers from a Flash Drive

1. Insert the flash drive into an available USB port on the target computer.
2. Navigate to the flash drive through Windows Explorer.
3. Run the *Setup.exe* file located at the root of the flash drive.
4. Follow the prompts presented by the installer.

1.2.2.3 BCI-1553 and LabVIEW FPGA Projects

Once the driver software is installed on a target computer, the BCI-1553 module can be discovered and/or added to an FPGA project. There are no slot requirements for the module and the number of modules allowable on a single system depends entirely on the FPGA space available. See Appendix C: FPGA Utilization for more information on the FPGA utilization of each operation mode.

The BCI-1553 module **requires a cRIO chassis to be running in FPGA mode** to be discovered or even added. In other words, the BCI-1553 module is **incompatible with National Instruments Scan Engine**.

1.2.2.4 Adding the BCI-1553 Module to a LabVIEW FPGA Project

The BCI-1553 module supports auto-discovery and manual addition to a project. The below steps outline the basic procedure for manually adding the BCI-1553 module to a LabVIEW FPGA project. Use the 'Existing target or device' option at step 2 for auto-discovery of the BCI-1553 module.

1. Add a new C Series module to the FPGA target.

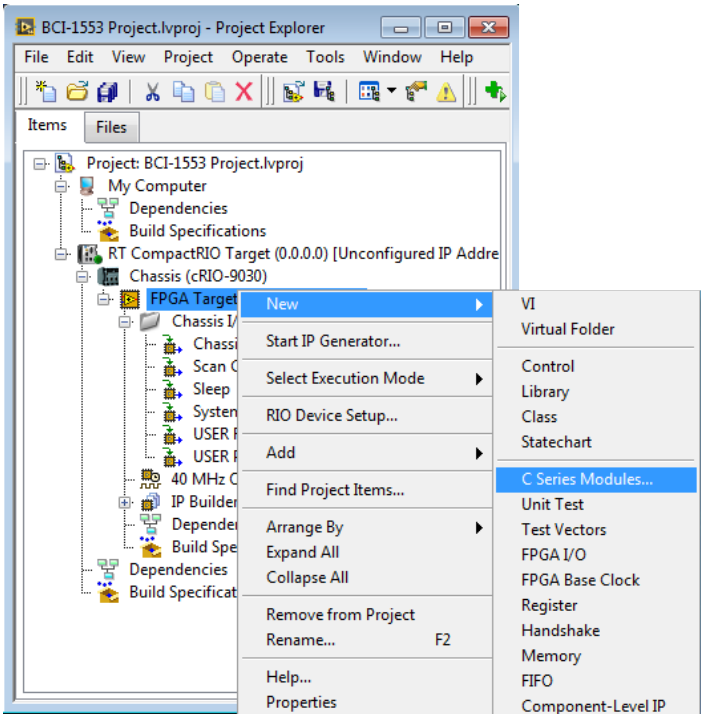

Figure 1-A: New C Series module.

2. Select to add a new target or device.

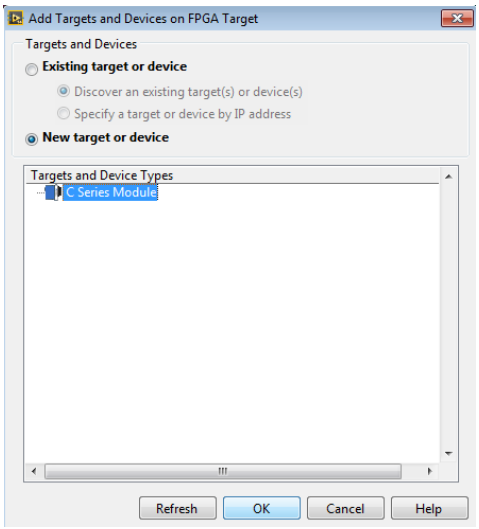

Figure 1-B: New target or device.

3. Select the module name, set the type to BCI-1553-BC, and set the slot.

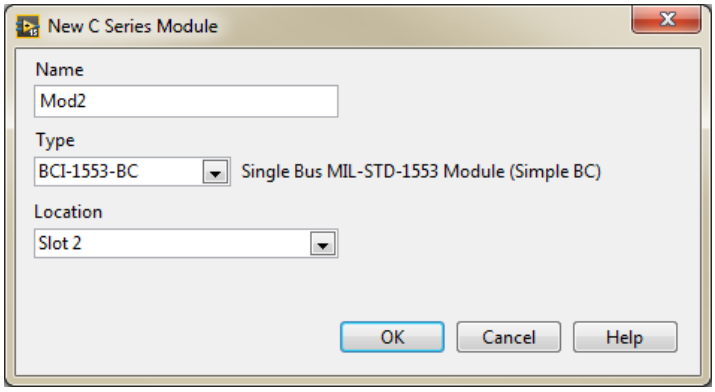

Figure 1-C: Setting the details of the BCI-1553 module to add.

4. Find the BCI-1553 module in the project.

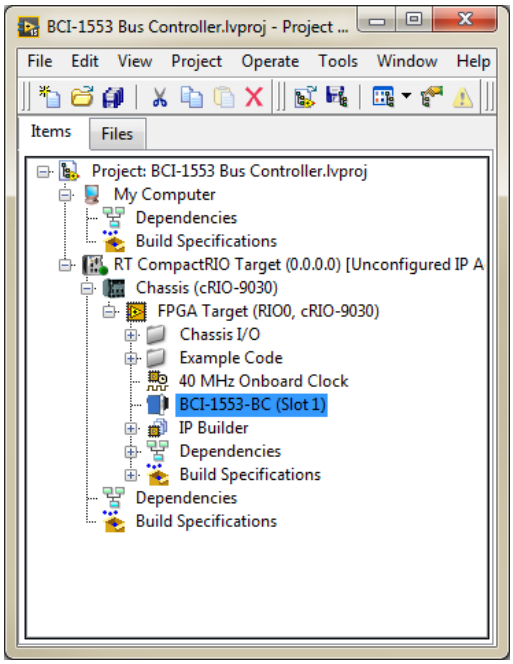

Figure 1-D: BCI-1553 module present in the project.

2 Hardware

2.1 Hardware General Overview

2.1.1 4-Contact Power Connector

This connector is the interface for an external power supply to the BCI-1553 module. An external power supply is required for the BCI-1553 to operate.

2.1.1.1 Voltage + (9V - 24V)

Pin(s): 4

The BCI-1553 module requires a stable supply of 9V - 24V for proper operation. Further, power required to maintain the operation of the module varies as the bus traffic increases relevant to the BCI-1553. When a remote terminal on the BCI-1553 is at near 100% bus load, the module requires a power supply that can maintain at least 270 mA at 24V (approx. 6.5 W).

2.1.1.2 Common

Pin(s): 3

2.1.1.3 No Connect

Pin(s): 1, 2

2.1.2 DB-15 1553 Interface

This is the primary interface to the MIL-STD-1553 buses on the BCI-1553. Additional utility functions may be accessed on this interface.

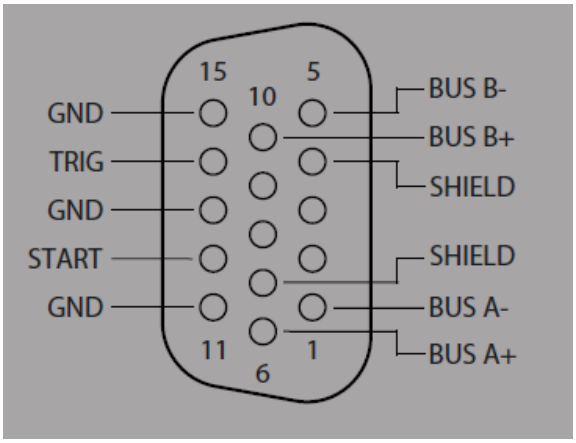

Figure 2-A: DB-15 output diagram for the BCI-1553.

2.1.2.1 Bus A+, Bus A-, Shield

Pin(s): 6, 1, 7

The primary MIL-STD-1553 bus on the BCI-1553 module. Depending on the operation mode of the BCI-1553 module, transmission on this bus can be performed selectively. Bus A is typically the main bus for communication with the BCI-1553 module. Behavior regarding activity on either Bus A or Bus B can be modified in software.

2.1.2.2 Bus B+, Bus B-, Shield

Pin(s): 10, 5, 6

The secondary MIL-STD-1553 bus on the BCI-1553 module. Depending on the operation mode of the BCI-1553 module, transmission on this bus can be selectively inhibited. Bus B is typically the redundant bus for communication with the BCI-1553 module. Behavior regarding activity on Bus B can be modified in software.

2.1.2.3 External Start

Pin(s): 12

Currently disabled for future implementation.

2.1.2.4 External Trigger

Pin(s): 14

Currently disabled for future implementation.

2.1.2.5 Ground

Pin(s): 11, 13, 15

2.1.3 LED Indicators

The LEDs present on the face of the BCI-1553 module are a quick way to discern the state of the module (Figure 2-B). As the module is interfaced through software, the state of each of these LEDs may change to indicate activity on the BCI-1553.

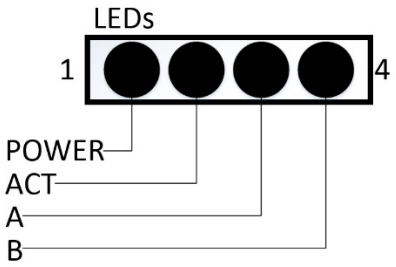

Figure 2-B: BCI-1553 LED Assignment.

2.1.3.1 Power (LED 1)

When the BCI-1553 module is connected to the backplane of a C-Series chassis and is receiving adequate power from the front connector, this LED will turn on. If there is no FPGA application running on the interfacing chassis, this LED will not turn on.

2.1.3.2 Act (LED 2)

The Act LED is used to indicate whether the BCI-1553 module is ready to be started for a particular operation mode. When in Bus Controller mode, this LED will turn on when the 'Initialize BC' (Figure 3-D) method is called and will turn off when the 'Release BC' (Figure 3-G) method is called.

2.1.3.3 Bus-A (LED 3), Bus-B (LED 4)

The Bus-A and Bus-B LEDs will flash on for brief periods of time. In general, these LEDs indicate that there is valid traffic on either bus.

3 Software

3.1 General Overview

The BCI-1553 module is capable of operating in any one of four modes: Remote Terminal, Bus Controller, Bus Monitor, and Advanced. It is not currently possible to operate a module in multiple operating modes at the same time. For example, the BCI-1553 module may not be in a remote terminal mode and a bus controller mode at the same time, while using the basic Bloomy drivers. The exception to this is when using the 'Advanced' operating mode. This mode allows for extremely precise control of the internal state of the BCI-1553 module, enabling the simultaneous operation of all three operating modes at once. When in one of the basic operating modes, it is still possible to populate a single chassis with several BCI-1553 modules, each in their own operating mode.

Each mode has its own set of API calls and its own variable FPGA utilization. To access the API for a module, it is necessary to place an I/O Device on the block diagram of an FPGA VI (Figure 3-A). Once placed, method and property nodes can be connected to the I/O Device refnum to access the API. Please note that a 1-second delay is required to accompany the I/O Device refnum instantiation to allow the BCI-1553 module the requisite time to finish internal initialization.

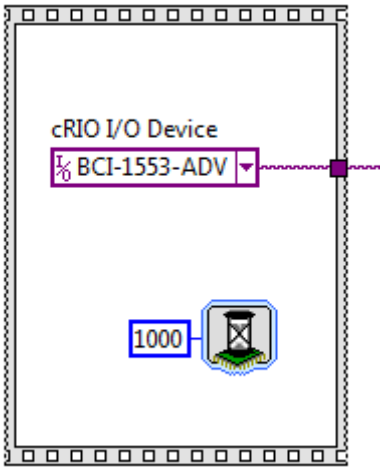

Figure 3-A: Creating a BCI-1553 Module Reference.

3.2 Common API

These methods and properties are constant across all operation modes. Most of these are read-only properties.

3.2.1 Serial Number (Property)

Returns the serial number associated with the BCI-1553 module present in the chassis (Figure 3-B).

3.2.2 Module ID (Property)

Returns the module ID associated with the BCI-1553 module present in the chassis (Figure 3-B).

3.2.3 Vendor ID (Property)

Returns the vendor ID associated with the BCI-1553 module present in the chassis (Figure 3-B).

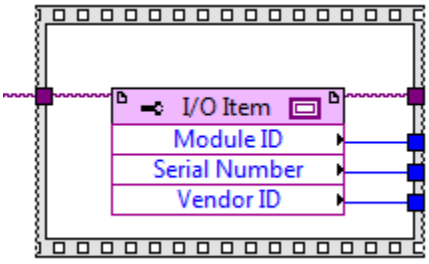

Figure 3-B: Reading the Module ID, Serial Number, and Vendor ID.

3.2.4 Change Mode (Method)

A utility for changing the operating mode of the module. This method should never be used in a final application without extreme caution.

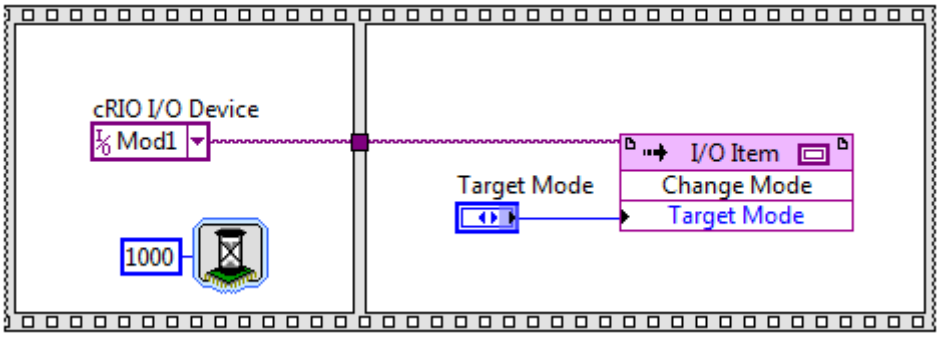

Figure 3-C: A typical use of the Change Mode method.

3.3 Advanced Operation

The Advanced mode sets up the BCI-1553 module for direct access to the core Holt 6130 configuration registers. Please refer to the *hi-6130_v-rev-o.pdf* document provided at installation for specific information on register behavior.

3.3.1 Read Non-Sequential Address (Method)

This function (Figure 3-D) is used to read from a singular register on the Holt 6130. This method is not optimized for doing sequential memory access, and should be leveraged when reading registers that are not adjacent.

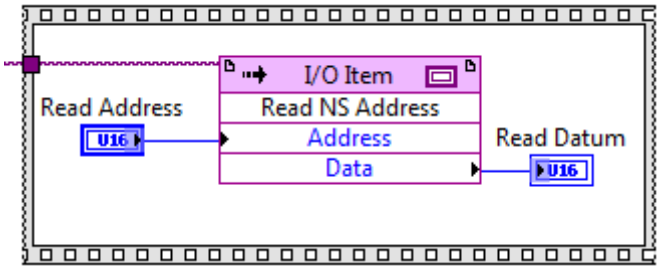

Figure 3-D: Reading data from a register.

3.3.2 Write Non-Sequential Address (Method)

This function is used to write data to a singular register on the Holt 6130. This method is not optimized for doing sequential memory access, and should be leveraged when configuring registers that are not adjacent.

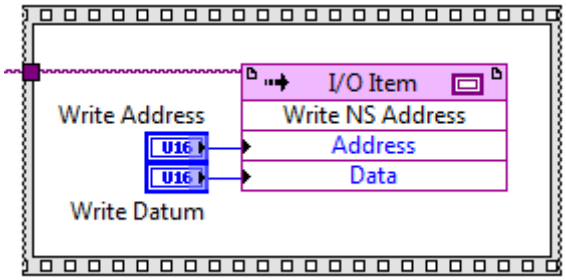

Figure 3-E: Writing data to a register.

3.3.3 Read Sequential Address

This function is similar to the *Read Non-Sequential Address* method. It is used for reading data from registers on the Holt 6130. Importantly, this method has been optimized for reading data from registers that are adjacent to each other. It is slightly less efficient when used for reading non-adjacent addresses. Figure 3-F displays using the method to read from 32 adjacent addresses. The first call to the method should always set the 'Initialize' argument to TRUE and provide an address to initialize to. The initialization call will return the data contained in the initialized address. The remaining consecutive calls, until the 'Initialize' argument is set TRUE again, will return the data stored at incrementing addresses from the initial address.

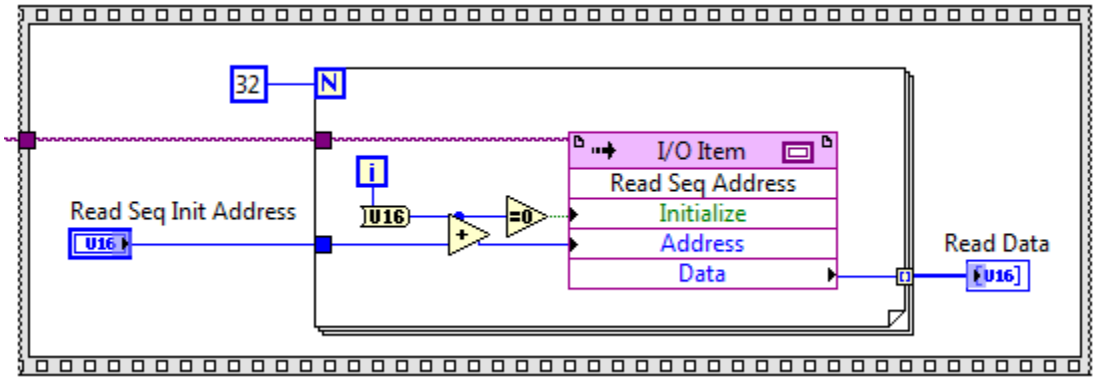

Figure 3-F: Performing a 32-address sequential read.

3.3.4 Write Sequential Address (Method)

This function is similar to the *Write Non-Sequential Address* method. It is used for writing data to registers on the Holt 6130. Importantly, this method has been optimized for writing data to registers that are adjacent to each other. It is slightly less efficient when used for writing non-adjacent addresses. Figure 3-G displays using the method to write to 32 adjacent addresses. The first call to the method should always set the 'Initialize' argument to TRUE and provide an address to initialize to. The initialization call will write the first data element to the initial address. The remaining consecutive calls, until the 'Initialize' argument is set TRUE again, will write data to each address following the initial address.

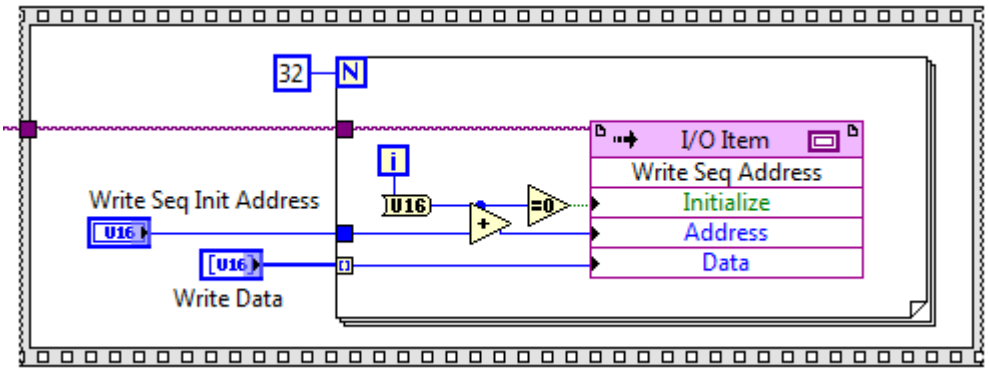

Figure 3-G: Writes 32 data points to 32 consecutive addresses.

3.4 Example Code

The BCI-1553 installer ships with several examples for using the BCI-1553. All of these examples assume the BCI-1553 is in the first slot of a National Instruments 9030 C-Series chassis. All examples can be found in <LabVIEW 2015 Directory>\examples\Bloomy Controls\BCI-1553-ADV.

4 Troubleshooting

4.1 Common Scenarios

General		
Issue	**Cause**	**Fix**
Power LED does not turn on.	FPGA code is not running on the host C-Series chassis.	When an application is not running on a C-Series FPGA, the chassis holds all modules in a sleep state. Running a simple application can verify proper power is being supplied to the BCI-1553.
	Incorrect power being supplied to the BCI-1553 module.	Make sure that the power supplied to the BCI-1553 module conforms to the details in section 2.1
BCI-1553 module does not transmit.	Incorrect bus termination.	Check that the MIL-STD-1553 bus has been setup properly.
	Module not initialized.	Call the initialize function relevant for the mode of operation the BCI-1553 module is in.
BCI-1553 module is not discoverable in a cRIO chassis	Incorrect or no power being supplied to module.	Make sure that the power supplied to the BCI-1553 module conforms to the details in section 2.1.
	Driver not installed.	Make sure that the BCI-1553 driver is installed on the target machine for the correct version of LabVIEW.
	Module not added to the FPGA of the C-Series chassis.	Make sure that the BCI-1553 module is added as a module to the FPGA resource. The module does not support being added to the chassis, which is a scan-engine interface.
Error 7 Encountered at FPGA compile time	Conflicting driver versions.	This error typically only occurs when two conflicting versions of the RT driver set are present on the same machine. This issue can be avoided completely by uninstalling the original driver ***BEFORE*** installing a new version. If encountering this issue, please check that only the latest version of the RT driver has been installed to this location: ..\ National Instruments\LabVIEW 2015\Targets\NI\FPGA\cRIO\other\ Delete the 'BCI-1553-BC' directory here and attempt to reinstall the driver.

Error 65221 Encountered at FPGA compile time	Conflicted method node state.	When updating from a previous driver version to a new one, LabVIEW does not immediately detect changes in the API. This results in behavior where method nodes and property nodes can have incorrect data wired to them, or indicators wired to input terminals, etc… To resolve this, replace the BCI-1553-ADV device node on the block diagram and reselect all affected methods. The methods can be selected by clicking on the method node and selecting the same method name (or the equivalent) from the dropdown list. Once this selection is made, you should be able to detect the error by a broken run arrow.
Module does not appear to perform any methods or properties properly.	Conflicting operating mode.	The vast majority of BCI-1553 modules are shipped in the Remote Terminal operating mode and must be transitioned to the target operating mode. To do this, verify the current operating mode of the module by using auto-discovery in your LabVIEW project. Once the current operating mode has been determined, use the driver for that operating mode to transition the module to the desired operating mode. Verify the module transitioned mode as intended by using auto-discovery in the LabVIEW project again.
Data written to addresses is inconsistent or incorrect.	Resource conflict between methods.	When designing the FPGA application, make sure that multiple instances of the 'Read Sequential Address' and 'Write Sequential Address' cannot execute at the same time for the same module. Since these reads and writes require several iterations to operate, another request of the same kind could alter the referenced initial address. 'Read Sequential Address' and 'Write Sequential Address' should be able to execute completely in parallel without causing harm. Two instances of 'Write Sequential Address' should not execute in parallel at the same time.

5 Appendix A: API Summary

Remote Terminal (*Bold = Output*)			
Function	**I/O**	**I/O Data Type**	**Description Section**
Read NS Address	Address	U16	Section 3.3.1
	Data	**U16**	
Write NS Address	Address	U16	Section 3.3.2
	Data	U16	
Read Seq Address	Initialize	Bool	Section **Error! Reference source not found.**
	Address	U16	
	Data	U16	
Write Seq Address	Initialize	Bool	Section 3.3.4
	Address	U16	
	Data	**U16**	

6 Appendix C: FPGA Utilization

Below are the typical results of using the entire API for a single BCI-1553 module in the Advanced operating mode. These results may vary based on the optimizations used when compiling FPGA code with the BCI-1553 module present. When using multiple BCI-1553 modules in a single cRIO chassis, it is expected that FPGA utilization will increase linearly, assuming full API usage.

Just having the module present in a LabVIEW project will result in a baseline impact on FPGA utilization (approx. 50% of the below estimates). For any FPGA applications not using the BCI-1553, it is strongly suggested to make sure it is removed from the LabVIEW project before compiling FPGA applications.

Remote Terminal			
FPGA Component	**Number Used**	**% Utilized Kintex 7 70T FPGA**	**% Utilized Kintex 7 160T FPGA**
Total Slices	1603	14.8	6.3
Slice Registers	5144	6.3	2.5
Slice LUTs	4378	10.7	4.3
Block RAMs	3	2.2	0.9
DSP48s	0	0	0