

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

NI-ATCA FPGA Modules

2023-09-26

Contents

Getting Started. 4
Installed Examples. 4
Configuring a New ATCA-3671 FPGA Target. . . . . .
Configuring DRAM with FPGA Memory Items. 5
NI ATCA FPGA Modules Devices. 6
ATCA-3671 FPGA Module for ATCA. 6
ATCA-3671 Front Panel Diagram. 6
ATCA-3671 SOM Commands. . . . . .
ATCA-3671 Clocking Features. 9
ATCA-3671 User-Accessible Multi-Gigabit Transceivers (MGTs). 12
Deploying a Bitfile to the ATCA-3671. . . . . .
RTM-3661 PCI Express Rear Transition Module for ATCA. 24
RTM-3661 Front Panel Diagram. 24
RTM-3661 Socketed CLIP Interface. 24
AIO-3681 Digitizer Module for FMC. 28
AIO-3681 Socketed CLIP Interface. 30
AIO-3682 Signal Generator Module for FMC. 31
AIO-3682 Socketed CLIP Interface. 32
AIO-3691 High-Speed Serial Adapter Module. 33
AIO-3691 Socketed CLIP Interface. 34
Programming. 39
NI ATCA Modules VIs. 39
ATCA-3671 Open FPGA Reference. 40
ATCA-3671 Get FPGA ID. 41
ATCA-3671 Transfer Bitstream. 41
ATCA-3671 Configure Jitter Cleaner. 42
ATCA-3671 Configure RTM Line Rate. 44
ATCA-3671 Remote Bitstream Download. 45
3681 Generic Config. 46
3681 Gain Config. 47
3682 Generic Config. 48
3682 Gain Config. 49
Route GLCK. 50

Using the Standard Functionality for error in Parameters. 51
Using the Standard Functionality for error out Parameters. 52

Getting Started

To get started using your NI ATCA system and software, refer to the **ATCA-3671 Getting Started Guide**. The getting started guide provides information about the following tasks:

- Installing the software and driver
- Installing the adapter modules
- Configuring the hardware in MAX
- Programming the hardware

The getting started guide for your device available at ni.com/manuals.

Tip Complete the steps in the getting started guide for your hardware before using the information in this help file.

Related Topics

Installed Examples

Installed Examples

NI ATCA FPGA Modules includes several example applications for LabVIEW. These examples serve as interactive tools and programming models. Additionally, they can be used as building blocks in your own applications.

LabVIEW users can use the Example Finder to search or browse examples. NI ATCA examples are classified by keyword, so you can search for a particular device or measurements function. With LabVIEW running, select **Help»Find Examples** to launch the NI Example Finder. The NI Example Finder offers two ways to access all installed LabVIEW example VIs and their descriptions.

- Click the Browse tab to locate NI ATCA examples by task at **Hardware Input and Output»ATCA**.

- Click the Search tab to search all installed examples by keyword. For example, search for "ATCA" to locate all NI ATCA examples.

Examples also are available online that demonstrate NI ATCA basics, such as using DRAM, acquiring data from adapter modules, and performing high throughput streaming. Refer to ni.com/examples for these examples and for more information.

Configuring DRAM with FPGA Memory Items

The ATCA-3671 Getting Started DRAM example demonstrates multiple methods to access the external DRAM memory from the FPGA. Complete the following steps to configure DRAM with FPGA memory items using the ATCA-3671 Getting Started DRAM VI.

1. Within LabVIEW, select **Help»Find Examples** and search ATCA, then open ATCA 3671 - Getting Started DRAM.lvproj.
2. Open the ATCA 3671 - Getting Started DRAM (FPGA).vi from the project. This VI demonstrates the various methods for accessing the DRAM from the FPGA on the ATCA-3671.
3. Examine the front panel input controls.
4. Open the ATCA 3671 - Getting Started DRAM (Host).vi from the project. This Host VI controls the data that flows into and out of the different external memory access methods on the FPGA.
5. Examine the front panel input controls and follow the in-software instructions.

	**Note** To create new Memory or FIFO resources on your target, right-click on the target, select **New...** and follow the prompts.

6. Click the **Run** button to run the ATCA 3671 - Getting Started DRAM (Host).vi. Experiment with this example to determine your preferred method(s) of accessing the DRAM.

	**Note** You can copy segments of the code included in these examples into your application.

NI ATCA FPGA Modules Devices

Expand this book for information about the following devices:

- ATCA-3671 FPGA Module for ATCA
- RTM-3661 PCI Express Rear Transition Module for ATCA
- AIO-3681 Digitizer Module for FMC
- AIO-3682 Signal Generator Module for FMC
- AIO-3691 High-Speed Serial Adapter Module

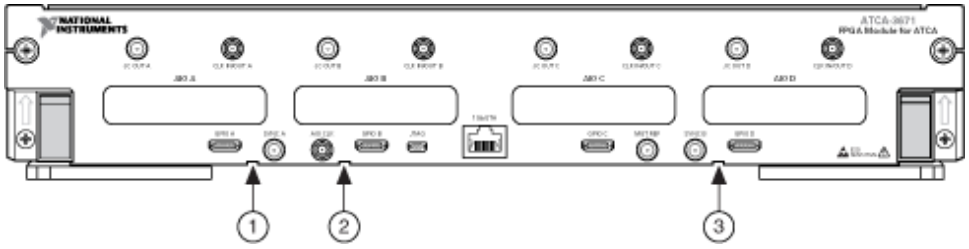
ATCA-3671 FPGA Module for ATCA

This section includes reference information about the ATCA-3671 module, including the clocking features and user-accessible multi-gigabit transceivers (MGTs).

Refer to the specifications document for your hardware at ni.com/manuals for complete specifications.

ATCA-3671 Front Panel Diagram

This section includes a diagram of the ATCA-3671 front panel and notes the locations of LED lights.



1. Red LED
2. Green LED
3. Blue LED

Boardctl Commands

Once you have accessed the System-on-Module (SOM) according to the instructions in SOM Commands, enter any of the boardctl commands listed in the following table.

Command	Direction
`boardctl power - 0` `boardctl power - 1`	Powers on or powers off the ATCA-3671 module. `-0`, turn blade off `-1`, turn blade on
`boardctl program <fpga> <BIN file>`	Programs the specified FPGA with the specified binary file. `<fpga>: A B C D` `<BIN file>`: File to be programmed
`boardctl reset <device> <duration>`	Resets the specified component with a reset pulse equivalent to the `<duration> in ms`. `<device>: A B C D S P` `A B C D`: FPGA IDs S: GbE switch P: GbE PHY `<duration>`: Duration of reset pulse in ms (default is 10 ms)
`boardctl sensors <option>...`	`-p, --power`: Read power usage `-t, --temperature`: Read temperatures `-v, --voltage <source>`: Read voltages for `<source>`; either the global rails (G), a given FPGA (`A B C D`), or the power sequencer (P).
`boardctl fault - c` `boardctl fault - f`	`-c`: clear any temperature faults `-f`: log fault info to a given file
`boardctl gclk <fpga> <output> <input>`	Routes GCLK output (0-3) on specified FPGA to be controlled by input FPGA. `<fpga>`: Target FPGA (`A B C D`) `<output>`: Target GCLK net (`0 1 2 3`) `<input>`: Source clock (`A B C D`)
`boardctl info`	Prints out ATCA-3671 module information.

`boardctl rtminfo`	Prints out RTM board information.

Minicom Commands

Use minicom commands to interact with the ATCA fixed logic microblaze directly. Once you have accessed the System-on-Module (SOM) according to the instructions in SOM Commands, start a minicom session by entering `minicom <fpga>` (where `<fpga>` is the letter of the desired FPGA—`A B C D`) in an SSH connection.

The following table lists available minicom commands. You can enter numeric input in base ten (0) or hexadecimal (x0).

Command	Usage
`jc_spi_write <data>`	Writes four bytes to the jitter cleaner. The last 5 bits of data specify the register's address.
`jc_spi_read <address>`	Reads four bytes from the jitter cleaner at given address.
`jc_show_reg_vals`	Prints out the register contents of the jitter clea ner.
`jc_set_input_freq <input freq (khz)> <pdf_val [min|max|freq in hz] (optional)> <cp_current [100|200|400|1600] (optional)>`	Specifies the input frequency to the JC. You can also specify the PLL1 PDF and cp current. PLL1 R and N dividers are set accordingly.
`jc_set_output_freq <gth/sma> <output freq (khz)>`	Sets output frequency for the GTH or SMA by m odifying output dividers. Valid values must be fa ctors of the VCO frequency.
`jc_set_vco_freq <vcxo freq (khz)> <vco_freq (khz)>`	Sets VCO/VCOX parameters for PLL2. The VCOX f requency specifies the external VCOX frequency and should always be set to 122.88 MHz. The VC O frequency may be set within the range of 2,75 0 MHz TO 3,072 MHz. PLL2 R and N dividers ARE automatically set to compatible values.
`jc_enable_sma <true/false>`	Enables/disables the jitter cleaner SMA output.

`jc_get_status`	Checks if the jitter cleaner reports locked.
`jc_disable_lmk`	Disables jitter cleaner. LMK can be re-enabled by setting the output frequency.
`version`	Displays code version.

ATCA-3671 Clocking Features

There are several ways to configure the onboard and external clocks on the ATCA-3671.

The clocking resources available on the system are summarized in the table below. If the software programming environment provides support for runtime configuration of clocking resources, refer to specific software product documentation, such as the LabVIEW Help, for more information.

The following table describes the ATCA-3671 clock sources.

Name	Direction	Description
Fixed 100 MHz	N/A	A single 100 Fixed MHz reference clock is distributed to all four FPGAs on the ATCA-3671. This clock is free-running and may be used to derive any other clock frequencies desired inside the FPGA.
DDR3 Clock	N/A	A single 400 Hz reference clock is distributed to all four FPGAs on the ATCA-3671 for explicit use by the soft double data rate type three (DDR3) memory controller logic in the FPGA. The actual frequency of the logic clock available on the user interface side of the memory controller may vary based on the software platform, but is typically 167 MHz for the maximum DDR3 dynamic random access memory (DRAM) speed of 1,333 MHz.

SYNC A SYNC B	Input	Each of these two independent signals can be used to drive a clock to all four FPGAs on the ATCA-3671. By default, the SMA input on the front panel of the system is enabled as the signal source. However, in environments where a full ATCA backplane is present, this setting can be overridden in firmware to select the corresponding ATCA backplane signals to use the source instead. When added to a LabVIEW project, these clock sources are defined as free-running base clocks which support derived clocks. Therefore, the external clock source must be connected and stable at the correct frequency before running the FPGA application, even if they are not explicitly used in the diagram. Otherwise, errors are reported. In addition, the SYNC A and SYNC B signals must not be used as clocks in the same application (and vice versa) if they are used as FPGA I/O.
CLK IN/OUT A CLK IN/OUT B CLK IN/OUT C CLK IN/OUT D	Input/output	Each of these connectors on the front panel are coupled to one FPGA on the ATCA-3671 (denoted by A, B, C, or D) as either an input or output. The direction of the signal is configured in software, and can be changed dynamically at runtime. In LabVIEW, this clock source for each FPGA is defined as an external clock. This implies that it may not always be stable or co

		nnected and therefore cannot b e used as the default top level c lock or to generate derived cloc ks.
JC OUT A JC OUT B JC OUT C JC OUT D	Output	Each FPGA on the ATCA-3671 ca n drive out an ultra-low phase n oise clock through a designated onboard jitter cleaning phase-l ocked loop (PLL). The exact fre quency configuration of this PL L is controlled by software. See Configuring the Jitter Cleaner f or Use as a Clock Source for mo re information.
MGT REF	Input	An input on the front panel use d to distribute a single multi-gig abit transceiver (MGT) reference clock to every user-accessible h igh-speed transceiver on all fou r FPGAs on the ATCA-3671.
AIO CLK	Input	An input on the front panel use d to drive an auxiliary external c lock into all four of the AIO slots on the front panel of the ATCA-3 671. This clock is not driven dir ectly into the FPGA. Therefore, i ts usage and availability are de pendent on the ATCA I/O modul e present in the system.

ATCA-3671 Jitter Cleaning Phase-Locked Loop (PLL)

Each FPGA has a designated TI LMK04808 jitter-cleaning dual-PLL device, which is used when an ultra-low phase noise clock must be generated from a clock source inside the FPGA.

The primary input to the jitter cleaner is driven by a differential pair on the FPGA. The jitter cleaner then outputs two different clocks back to the FPGA: one MGT reference clock, which is duplicated so that it can reach every MGT quad on the FPGA, and one SMA output clock.

The architecture of the LMK04808 uses a voltage-controlled crystal oscillator (VCXO) to drive the first-stage PLL. This VCXO is 122.88 MHz. This frequency was chosen to meet the requirements of many wireless telecommunications applications.

Note When choosing which output frequencies to generate, consider the VCXO frequency because it impacts the set of valid frequencies available for the internal voltage-controlled oscillator (VCO) and output clocks of the LMK04808.

For more information on the configuration and performance of the jitter cleaner, refer to the LMK0480X datasheet on the Texas Instruments website at www.ti.com.

Software features are provided to fully configure the jitter cleaner based on a user-defined set of input and output frequencies.

ATCA-3671 User-Accessible Multi-Gigabit Transceivers (MGTs)

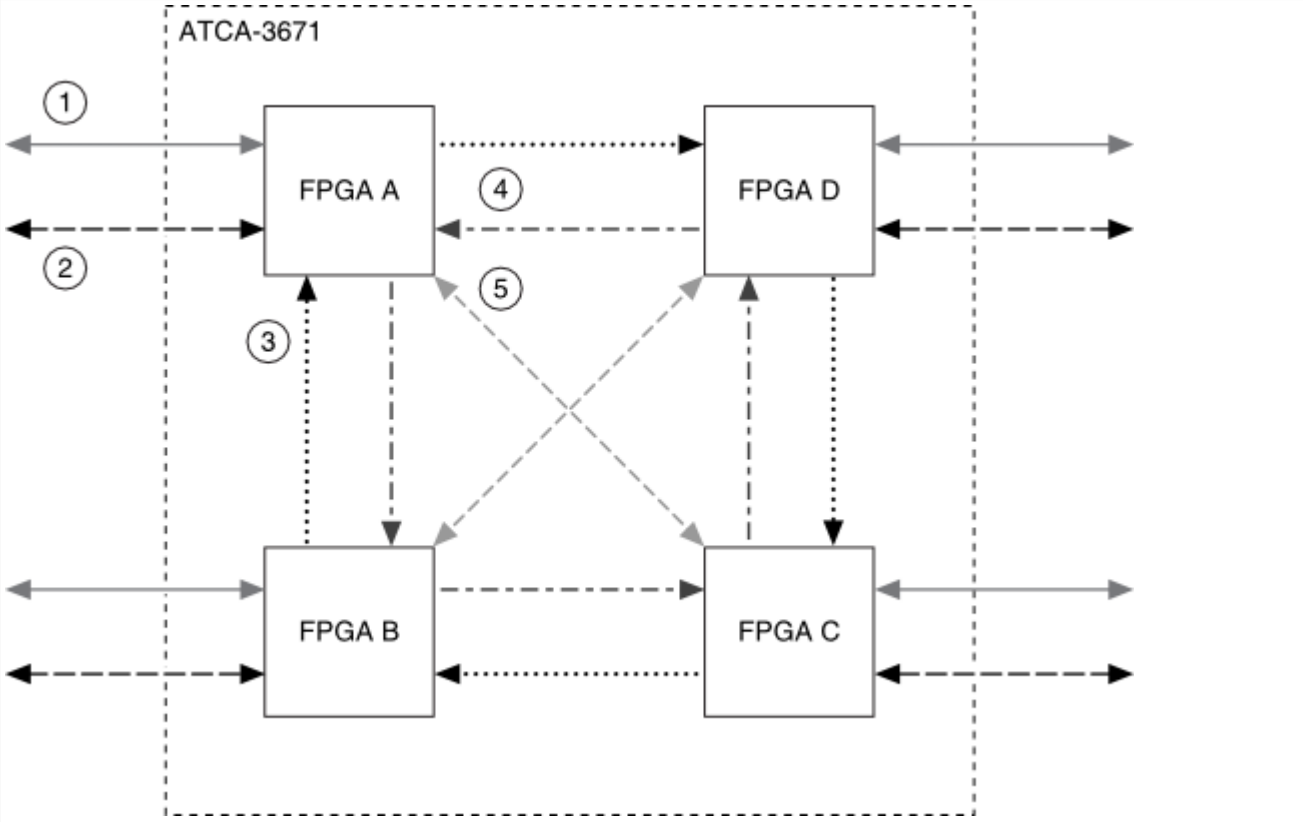
The ATCA-3671 hardware has 76 user-accessible MGTs available to the user for high-speed, low-latency data communication between devices.

The FPGAs on the ATCA-3671 feature a Xilinx GTH transceiver, which allows for supported line rates up to 12.5 Gbps on most interfaces.

The ATCA-3671 user-accessible MGTs are organized into five different groups:

- Three MGT groups connect the four FPGAs in a full mesh pattern
- One MGT group connects the FPGAs to the ATCA I/O (AIO) slot
- One MGT group connects the FPGAs to the Rear Transition Module (RTM) slot

The following figure shows the logical structure of the five transceiver groups.



1 AIO	3 FPGA Ring Up	5 FPGA Diagonal
2 RTM	4 FPGA Ring Down	

Each transceiver group is defined in the following table.

Name	Number of MGTs	Connection	Configuration
AIO	16	Each FPGA to a designated AIO module connector	Depends on which AIO module is populated in each slot
RTM	16	Each FPGA to the RTM	Depends on which RTM is populated in the system
FPGA Ring Up	16	Upstream, between FPGAs	Ring (A to D, B to A, C to B, and D to C)
FPGA Ring Down	16	Downstream, between FPGAs	Ring (A to B, B to C, C to D, and D to A)
FPGA Diagonal	12	Diagonal, between FPGAs	Diagonal (A to C, B to D, C to A, D to B)

MGT Reference Clocks

The ATCA-3671 family of devices provides a wide range of options for delivering reference clocks to each of the MGT interfaces in the system.

Three MGT reference clocks are distributed to reach every MGT bank (Quad) in every FPGA on the ATCA-3671, as shown in the following table.

MGT Reference Clock	Description
Fixed oscillator	A pair of fixed 312.5 MHz oscillators are used on the ATCA-3671 to deliver this common reference clock frequency to the entire board. One oscillator drives all reference clocks on FPGAs A and B, and the other drives all reference clocks on FPGAs C and D.
External SMA	An SMA connector on the front panel may be used to distribute an external reference clock to every transceiver on the entire board.
Jitter cleaner	Each FPGA has its own jitter-attenuating, dual-PLL device (LMK04808) that is capable of creating an ultra-low phase noise output that can be used as an MGT reference clock to all transceivers on the same FPGA.

Additional reference clock options are available on the AIO and RTM interfaces.

MGT Lane and Quad Mappings

The following table lists the lane and quad mappings for all MGTs on each ATCA-3671 FPGA.

GTH Usage	GTH Channel	Quad		Quad	GTH Channel	GTH Usage
Reserved	X0Y39	219		119	X1Y39	RingDown[15]
Reserved	X0Y38				X1Y38	
Reserved	X0Y37				X1Y37	RingDown[14]
Reserved	X0Y36				X1Y36	RingDown[13]

						RingDown[12]
Diag[11] Diag[10] Diag[9] Diag[8]	X0Y35 X0Y34 X0Y33 X0Y32	218		118	X1Y35 X1Y34 X1Y33 X1Y32	RingDown[11] RingDown[10] RingDown[9] RingDown[8]
Aio[15] Aio[14] Aio[13] Aio[12]	X0Y31 X0Y30 X0Y29 X0Y28	217		117	X1Y31 X1Y30 X1Y29 X1Y28	RingDown[7] RingDown[6] RingDown[5] RingDown[4]
Aio[11] Aio[10] Aio[9] Aio[8]	X0Y27 X0Y26 X0Y25 X0Y24	216		116	X1Y27 X1Y26 X1Y25 X1Y24	RingDown[3] RingDown[2] RingDown[1] RingDown[0]
Aio[7] Aio[6] Aio[5] Aio[4]	X0Y23 X0Y22 X0Y21 X0Y20	215		115	X1Y23 X1Y22 X1Y21 X1Y20	Rtm[15] Rtm[14] Rtm[13] Rtm[12]
Aio[3] Aio[2] Aio[1] Aio[0]	X0Y19 X0Y18 X0Y17 X0Y16	214		114	X1Y19 X1Y18 X1Y17 X1Y16	Rtm[11] Rtm[10] Rtm[9] Rtm[8]
RingUp[15] RingUp[14] RingUp[13]	X0Y15 X0Y14 X0Y13	213		113	X1Y15 X1Y14 X1Y13	Rtm[7] Rtm[6] Rtm[5]

RingUp[12]	X0Y12				X1Y12	Rtm[4]
RingUp[11]	X0Y11	212		112	X1Y11	Rtm[3]
RingUp[10]	X0Y10				X1Y10	Rtm[2]
RingUp[9]	X0Y9				X1Y9	Rtm[1]
RingUp[8]	X0Y8				X1Y8	Rtm[0]
RingUp[7]	X0Y7	211		111	X1Y7	Diag[7]
RingUp[6]	X0Y6				X1Y6	Diag[6]
RingUp[5]	X0Y5				X1Y5	Diag[5]
RingUp[4]	X0Y4				X1Y4	Diag[4]
RingUp[3]	X0Y3	210		110	X1Y3	Diag[3]
RingUp[2]	X0Y2				X1Y2	Diag[2]
RingUp[1]	X0Y1				X1Y1	Diag[1]
RingUp[0]	X0Y0				X1Y0	Diag[0]

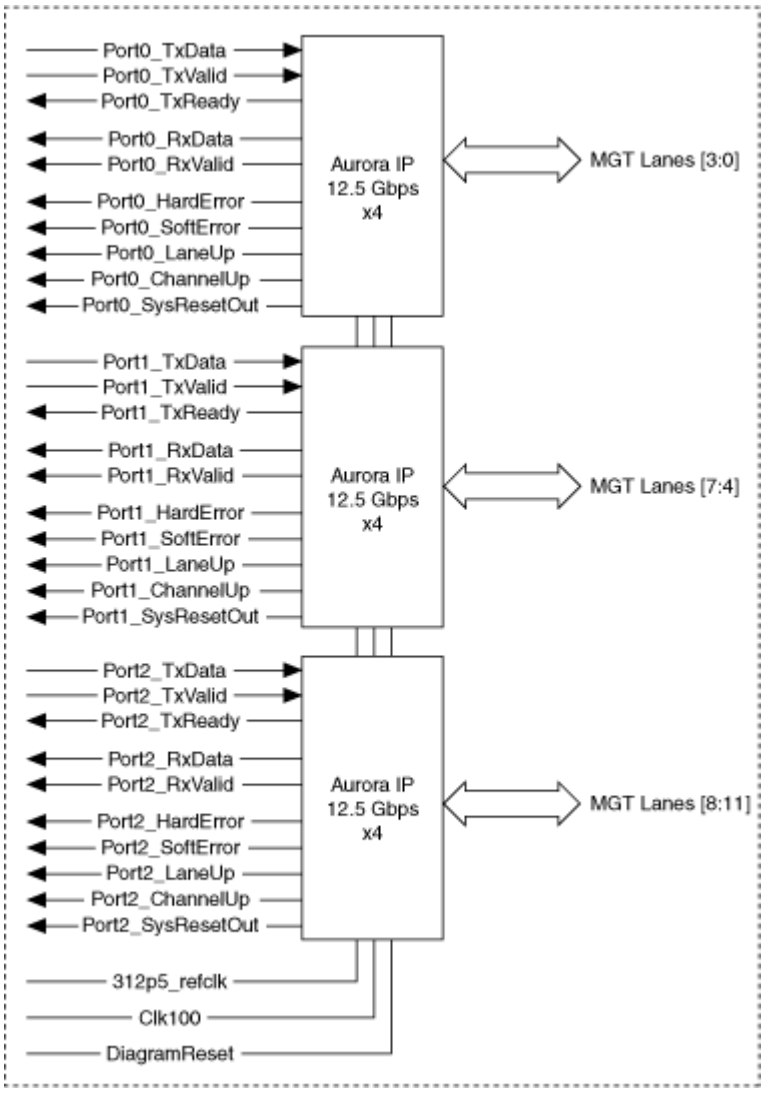
Inter-FPGA Socketed CLIP Interfaces

NI ATCA FPGA Modules includes Socketed CLIP examples for the inter-FPGA MGT groups.

The ATCA-3671 - Aurora Pattern example project includes an optimally hardware-efficient implementation of the Aurora protocol across all inter-FPGA lanes running at 12.5 Gbps. For the Ring Up and Ring Down groups, the CLIP consists of four x4 Aurora IP instances. For the Diagonal group, the CLIP consists of three x4 Aurora IP instances. Data is presented to the LabVIEW diagram in the same format as the Aurora IP structure, but each MGT group shares a single user clock between all instances.

Note Sharing a clock conserves a significant number of resources on the FPGA and prevents unnecessary clock domain crossing logic from being inferred, since all IP instances share the same external reference clock and do not require any additional buffering.

FPGA Diagonal Aurora CLIP Contents



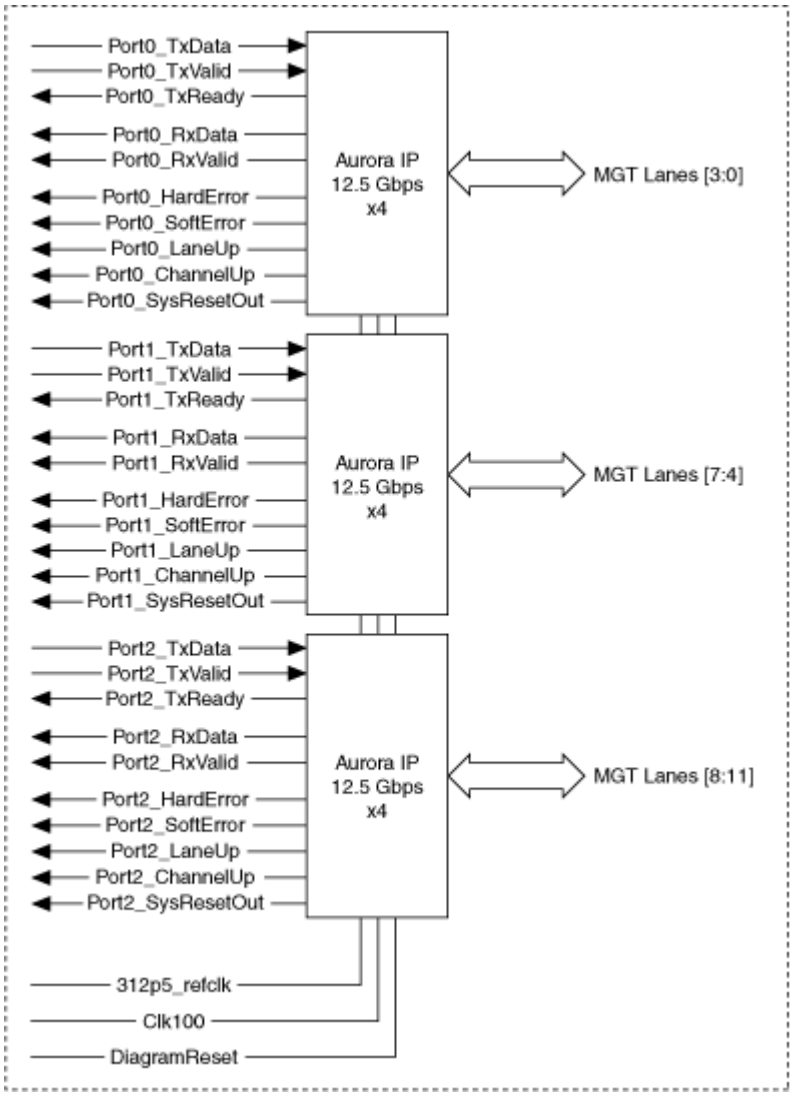
FPGA Diagonal Aurora CLIP Signals

Port	Direction	Clock Domain	Description
diag_txp(14:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.
diag_txn(14:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.

diag_rxp(14:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
diag_rxn(14:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
diag_312p5_refclk(3:0)	In	N/A	Reference clock inputs for the shared 312.5 MHz oscillator. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
diag_jc_refclk(3:0)	In	N/A	Reference clock inputs for the GTH reference clock generated by the jitter cleaner. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.

diag_sma_refclk(3:0)	In	N/A	Reference clock inputs for the shared front panel SMA connector. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
diag_fpga_ready	In	Async	Signal driven by fixed logic to indicate that all clocks and power supplies required by the FPGA are active and stable.

FPGA Ring Down and Ring Up CLIP Contents



FPGA Ring Up Aurora CLIP Signals

Port	Direction	Clock Domain	Description
ring_up_txp(15:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.
ring_up_txn(15:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.

ring_up_rxp(15:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
ring_up_rxn(15:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
ring_up_312p5_refclk(3:0)	In	N/A	Reference clock inputs for the shared 312.5 MHz oscillator. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
ring_up_jc_refclk(3:0)	In	N/A	Reference clock inputs for the GTH reference clock generated by the jitter cleaner. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.

ring_up_sma_refclk(3:0)	In	N/A	Reference clock inputs for the shared front panel SMA connector. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
ring_up_fpga_ready	In	Async	Signal driven by fixed logic to indicate that all clocks and power supplies required by the FPGA are active and stable.

FPGA Ring Down Aurora CLIP Signals

Port	Direction	Clock Domain	Description
ring_down_txp(15:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.
ring_down_txn(15:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.
ring_down_rxp(15:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
ring_down_rxn(15:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
ring_down_312p5_refclk(3:0)	In	N/A	Reference clock inputs for the shared 312.5 MH

			z oscillator. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
ring_down_jc_refclk(3:0)	In	N/A	Reference clock inputs for the GTH reference clock generated by the jitter cleaner. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
ring_down_sma_refclk(3:0)	In	N/A	Reference clock inputs for the shared front panel SMA connector. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each cl

			ock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
ring_down_fpga_ready	In	Async	Signal driven by fixed logic to indicate that all clocks and power supplies required by the FPGA are active and stable.

RTM-3661 PCI Express Rear Transition Module for ATCA

This section includes reference information about the RTM-3661 module, including the socketed CLIP interface.

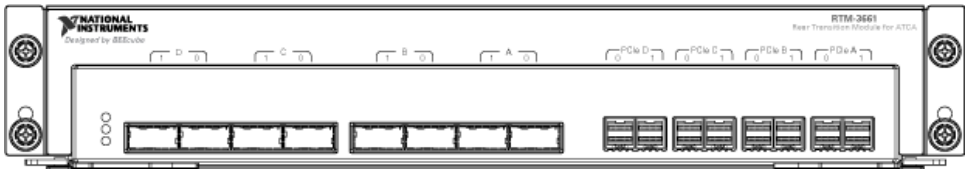
The RTM-3661 provides PCI Express Gen3 x8 host connectivity as well as two x4 Quad Small Form-factor Pluggable (QSFP) ports for every FPGA on the ATCA-3671.

The PCI Express ports support the latest Gen3 external cabling standard, and feature pass-through repeater devices on all transmit and receive lanes to support 8.0 Gbps line rates over copper cables up to 3 m in length. The QSFP ports feature active retimer devices in the transmit direction only and support up to 10.4167 Gbps line rates over a variety of cables.

Refer to the specifications document for your hardware at ni.com/manuals for complete specifications.

RTM-3661 Front Panel Diagram

This section includes a diagram of the RTM-3661 module.



RTM-3661 Socketed CLIP Interface

NI ATCA FPGA Modules includes a Socketed CLIP example for the RTM-3661.



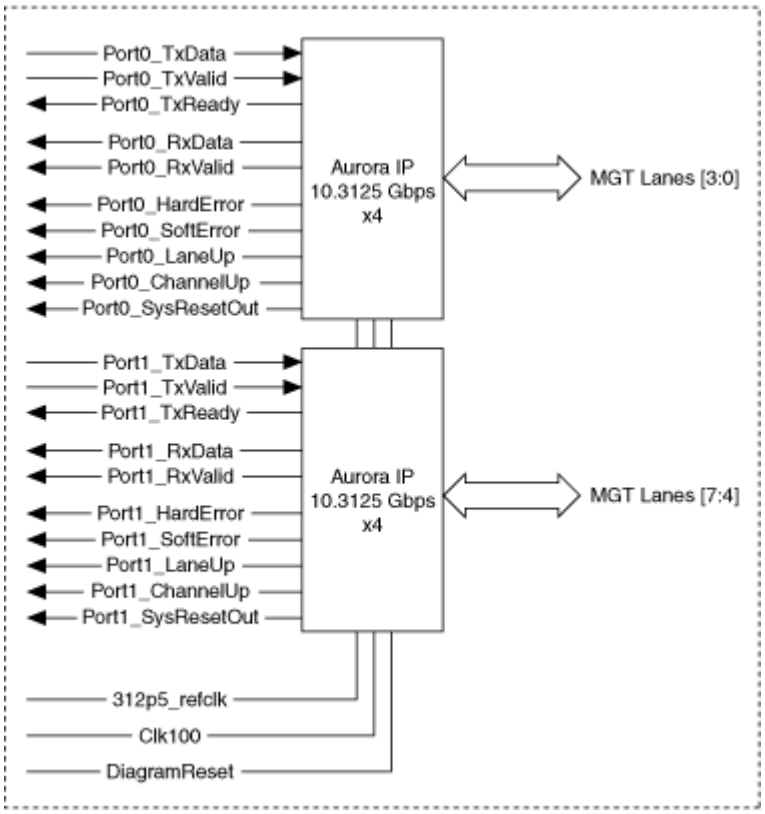
Note The x8 PCI Express host interface is critical to operation of the ATCA-3671 FPGA module in LabVIEW and is contained in the fixed logic of the FPGA. Therefore, those MGT lanes are not included in the Socketed CLIP definition. Eight lanes are available per FPGA (QSFP Port 0 and 1).

The provided example features an optimally hardware-efficient implementation of the Aurora protocol across all lanes running at 10.3125 GHz. Each QSFP port has its own x4 Aurora IP instance and data ports to the LabVIEW diagram, but both ports share a single user clock.



Note Sharing a clock conserves a significant number of resources on the FPGA and prevents unnecessary clock domain crossing logic from being inferred, since all IP instances share the same external reference clock and do not require any additional buffering.

RTM-3661 Aurora CLIP Contents



RTM-3661 Aurora CLIP Signals

Port	Direction	Clock Domain	Description
rtm_txp(7:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.
rtm_txn(7:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.
rtm_rxp(7:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
rtm_rxn(7:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
rtm_312p5_refclk(1:0)	In	N/A	Reference clock inputs for the shared 312.5 MH

			z oscillator. IBUFDS_GTE2 instances already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
rtm_jc_refclk(1:0)	In	N/A	Reference clock inputs for the GTH reference clock generated by the jitter cleaner. IBUFDS_GTE2 instances already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each clock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
rtm_sma_refclk(1:0)	In	N/A	Reference clock inputs for the shared front panel SMA connector. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and may be used directly in the CLIP. Each cl

			ock signal in the vector is mapped to a GTH Quad in the same order as the TX/RX pairs. For example, refclk(0) should be used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
rtm_jc_srcclk	Out	N/A	Optional output signal which can be driven out from the CLIP to be used as the source clock for the jitter cleaner (useful for recovered-clock applications where the interface IP produces a clock that must be externally filtered for reuse as a reference clock). The source clock to be driven out to the jitter cleaner is selected using the **External Clock Configuration** properties window.
rtm_fpga_ready	In	Async	Signal driven by fixed logic to indicate that all clocks and power supplies required by the FPGA are active and stable.

AIO-3681 Digitizer Module for FMC

This section includes reference information about the AIO-3681 module, including the block diagram and socketed CLIP interface.

The AIO-3681 is an expansion module that provides single- or dual-channel 12-bit analog-to-digital conversion at RF sampling rates up to 4 GS/s, depending on hardware configuration.

It features a TI ADC12D2000RF analog-to-digital converter (ADC) component and is shipped with software support for both dual-channel and interleaved dual-edge sampling (single-channel) modes of operation.

	**Note** For the ADC, 8 samples per channel are provided in the user diagram per data clock cycle. Therefore, the data clock always runs at one-eighth the frequency of the sampling clock. For example, at a sampling clock rate of 2.5 GHz, the data clock will run at 312.5 MHz.

Due to the complexity of the internal logic in the Socketed CLIP, and to provide more flexibility for cases when the external sampling clock is connected at runtime, a **CLIP Reset** signal is provided for explicit control over the reset behavior of the CLIP in the diagram.

	**Note** The self-calibration algorithm between the CLIP and ADC takes approximately 20 seconds to complete.

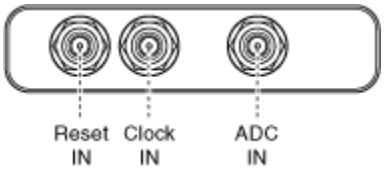
To help manage this initialization time, a **CLIP Ready** signal is also provided which can be monitored to determine when the ADC is ready to accept input data. Any interruptions to the external clock or software events, such as resetting the diagram or downloading a new bitstream, should always be followed by a reset to the CLIP.

	**Note** Rerun calibration after significantly altering the settings or conditions of the AIO-3681. To preform calibration, use the 3681 Generic Config VI to set the CAL bit (Addr: 0h; Bit 15) to low for a minimum of 1,280 sample clock cycles and then high for a minimum of 1,280 sample clock cycles. Visit `ni.com/info` and enter the Info Code exsq59 to access the specifications for the ADC12D2000RF and navigate to section 17.3.3 of the document for more information about the calibration feature.

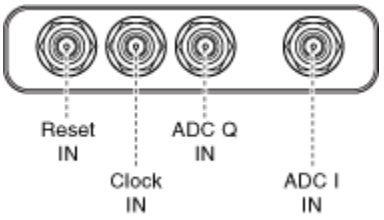
If the CLIP Calibration Status signal is true, AIO-3681 CLIP Calibration was successful. If CLIP Calibration Status signal is false after CLIP Initialization or Reset, calibration was unsuccessful and you should re-download the FPGA bitstream.

Note You can change gain range and configuration registers using 3681 Gain Config and 3681 Generic Config.

Single-Channel AIO-3681 Port Diagram



Dual-Channel AIO-3681 Port Diagram



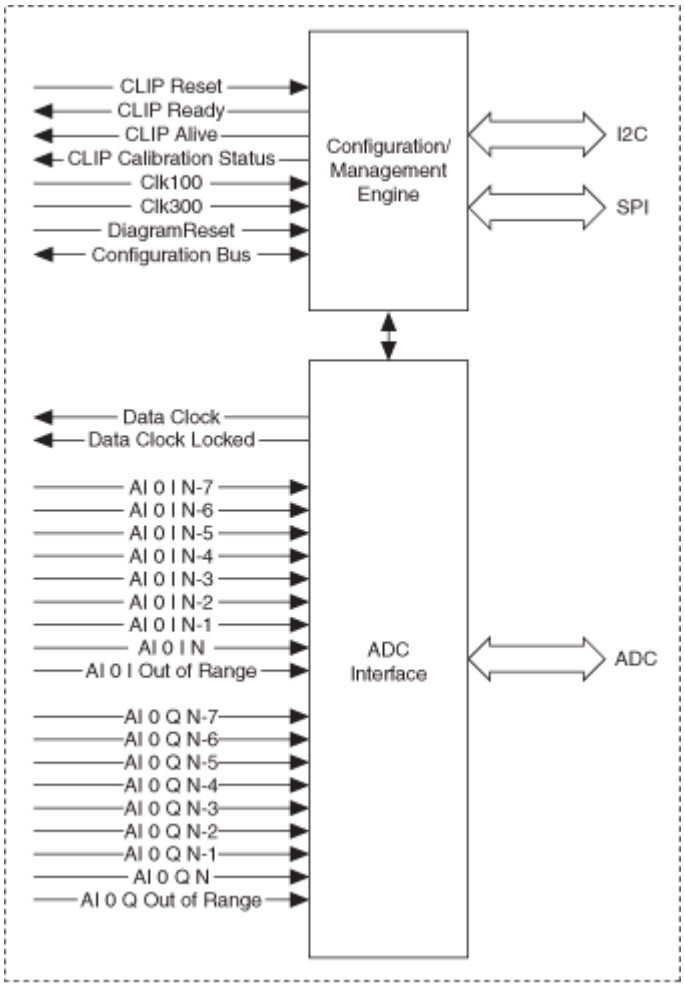
AIO-3681 Socketed CLIP Interface

LabVIEW support for the AIO-3681 includes two versions of the Socketed CLIP interface: one for dual-channel I/Q sampling, and one for single-channel dual-edge sampling (DES). Because the hardware assembly uses different components based on the desired sampling mode, always choose the appropriate Socketed CLIP to match the hardware configuration present in the system.

In dual-channel sampling mode, the same clock edge is used to sample both signals at the same time. In DES mode, opposite edges of the clock are used to sample each channel in an alternating fashion, effectively doubling the sample rate of the ADC.

For more information on the sampling modes supported by the ADC, refer to the TI ADC12D2000RF device data sheet on the Texas Instruments website at www.ti.com

AIO-3681 CLIP Contents



AIO-3682 Signal Generator Module for FMC

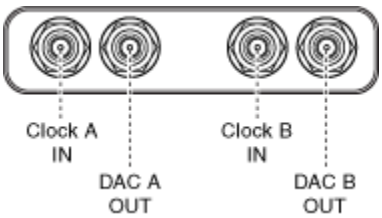
This section includes reference information about the AIO-3682 module, including the block diagram and socketed CLIP interface.

The AIO-3682 is an expansion module that provides dual-channel 14-bit digital-to-analog conversion at RF sampling rates up to 5.6 GS/s.

The module features two Analog Devices AD9129 digital-to-analog (DAC) components. Software support requires that both DACs be driven by the same external clock at all times, although the DACs can operate on separate sampling clock domains.

Refer to the specifications document for your hardware at ni.com/manuals for complete specifications.

AIO-3682 Port Diagram



AIO-3682 Socketed CLIP Interface

NI ATCA FPGA Modules includes a Socketed CLIP interface for using both DACs simultaneously with the same external sampling clock. The primary DAC (signal AO 0) must always be used in a LabVIEW design, and must be driven by a stable external sampling clock at all times. The secondary DAC (signal AO 1) is optional. It may be left out of a design, and its external clock input can be left disconnected when not in use. The logic clock generated by the CLIP for the diagram is derived purely from the output of the primary DAC (AO 0).

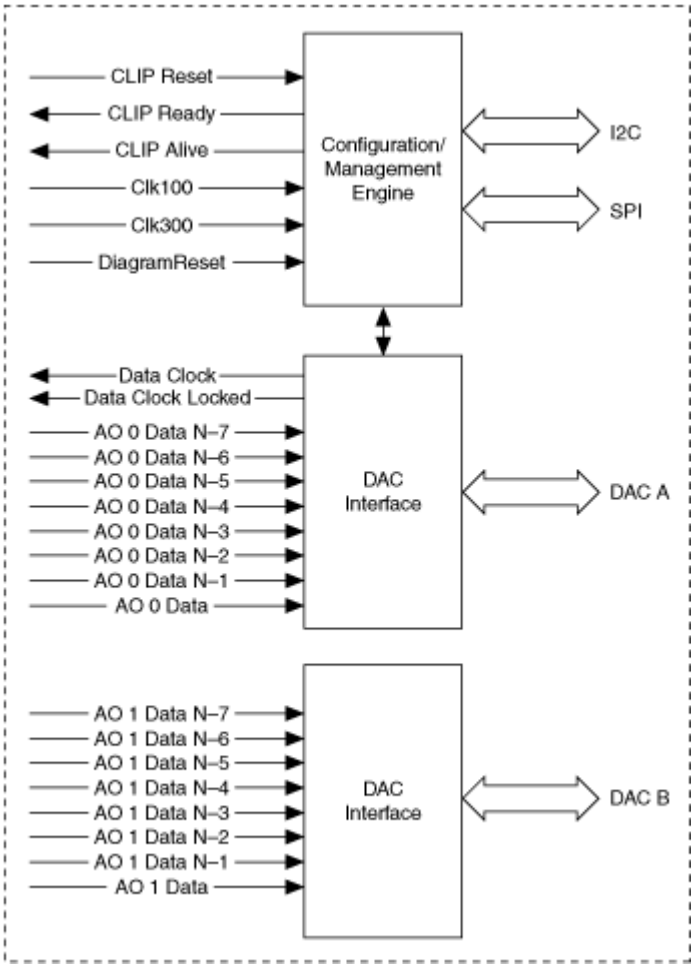
For each DAC, 8 samples are provided in the user diagram per data clock cycle. Therefore, the data clock always runs at one eighth the frequency of the sampling clock. For example, at a sampling clock rate of 2.5 GHz, the data clock will run at 312.5 MHz.

Due to the complexity of the internal logic in the Socketed CLIP, and to provide more flexibility for cases when the external sampling clock is connected at runtime, a CLIP Reset signal is provided for explicit control over the reset behavior of the CLIP in the diagram.

	Note The self-calibration algorithm between the CLIP and DAC takes approximately 20 seconds to complete.
	Note The initialization and reset sequence for this module generates a signal on the front panel DAC OUT ports.

To help manage this initialization time, a CLIP Ready signal is also provided which can be monitored to determine when the DAC is ready to accept output data. Any interruptions to the external clock or software events, such as resetting the diagram or downloading a new bitstream, should always be followed by a reset to the CLIP.

AIO-3682 CLIP Contents



AIO-3691 High-Speed Serial Adapter Module

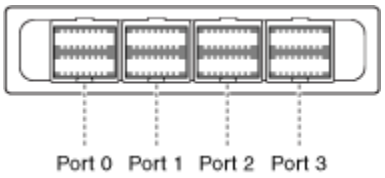
This section includes reference information about the AIO-3691 module, including the socketed CLIP interface.

The AIO-3691 is an expansion module that provides 16 lanes of high-speed serial connectivity between each FPGA and the front panel of the ATCA-3671.

It features four SFF-8644 (miniSAS-HD) connectors, which allow for quad x4 data interfaces to be implemented for communicating between ATCA-3671 FPGA modules or to other external high-speed serial equipment, such as the PXIe-6591 High-Speed Serial Instrument.

Refer to the specifications document for your hardware at ni.com/manuals for complete specifications.

AIO-3691 Port Diagram



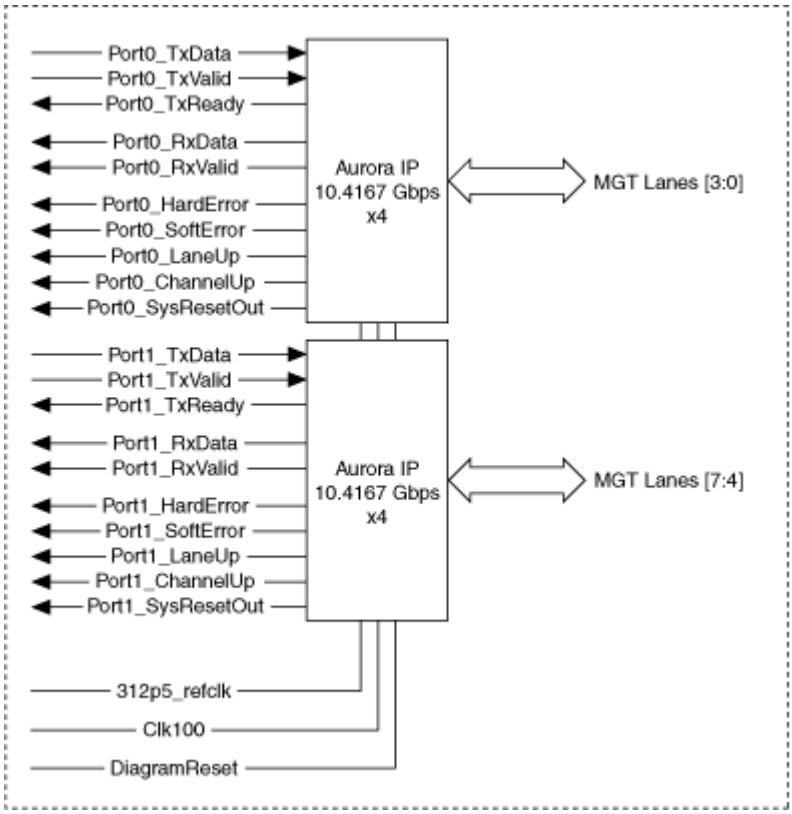
AIO-3691 Socketed CLIP Interface

NI ATCA FPGA Modules includes a Socketed CLIP example for the AIO-3691, which features an optimally hardware-efficient implementation of the Aurora protocol across all lanes running at 10.4167 GHz.

Each SFF-8644 port has its own x4 Aurora IP instance and data ports to the LabVIEW diagram, but all four ports share a single user clock.

	Note Sharing a clock conserves a significant number of resources on the FPGA and prevents unnecessary clock domain crossing logic from being inferred, since all IP instances share the same external reference clock and do not require any additional buffering.
	Note Since all four ports share a single user clock sourced by a single Aurora instance, Port 0 must always be connected to use this CLIP. Even if other ports are not in use, a stable link on Port 0 is required to maintain a valid user clock.

AIO-3691 Aurora CLIP Contents



AIO-3691 Aurora CLIP Signals

Port	Direction	Clock Domain	Description
fmc_dp_c2m_p(15:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.
fmc_dp_c2m_n(15:0)	Out (pad)	N/A	Dedicated high-speed serial transmit pins for GTH transceivers.
fmc_dp_m2c_p(15:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
fmc_dp_m2c_n(15:0)	In (pad)	N/A	Dedicated high-speed serial receive pins for GTH transceivers.
fmc_gtbclk0_m2c_p	In (pad)	N/A	Reference clock input pins from the FMC conne

			ctor. These ports are directly connected to the top-level ports of the FPGA and are unbuffered. Therefore, IBUFDS_GTE2 instances are required in the CLIP.
fmc_gtbclk0_m2c_n	In (pad)	N/A	Reference clock input pins from the FMC connector. These ports are directly connected to the top-level ports of the FPGA and are unbuffered. Therefore, IBUFDS_GTE2 instances are required in the CLIP.
fmc_gtbclk1_m2c_p	In (pad)	N/A	Reference clock input pins from the FMC connector. These ports are directly connected to the top-level ports of the FPGA and are unbuffered. Therefore, IBUFDS_GTE2 instances are required in the CLIP.
fmc_gtbclk1_m2c_n	In (pad)	N/A	Reference clock input pins from the FMC connector. These ports are directly connected to the top-level ports of the FPGA and are unbuffered. Therefore, IBUFDS_GTE2 instances are required in the CLIP.
fmc_312p5_refclk(3:0)	In	N/A	Reference clock inputs for the shared 312.5 MHz oscillator. IBUFDS_GTE2 instances are already present in fixed logic, so these input ports are already buffered and m

			ay be used directly in th e CLIP. Each clock signa l in the vector is mappe d to a GTH Quad in the same order as the TX/R X pairs. For example, re fclk(0) should be used f or the Quad containing txp/txn/rxp/rxn(3:0), an d so on.
fmc_jc_refclk(3:0)	In	N/A	Reference clock inputs for the GTH reference cl ock generated by the jit ter cleaner. IBUFDS_GT E2 instances are alread y present in fixed logic, so these input ports are already buffered and m ay be used directly in th e CLIP. Each clock signa l in the vector is mappe d to a GTH Quad in the same order as the TX/R X pairs. For example, re fclk(0) should be used f or the Quad containing txp/txn/rxp/rxn(3:0), an d so on.
fmc_sma_refclk(3:0)	In	N/A	Reference clock inputs for the shared front pa nel SMA connector. IBU FDS_GTE2 instances ar e already present in fixe d logic, so these input p orts are already buffere d and may be used dire ctly in the CLIP. Each cl ock signal in the vector is mapped to a GTH Qu ad in the same order as the TX/RX pairs. For exa mple, refclk(0) should b

			e used for the Quad containing txp/txn/rxp/rxn(3:0), and so on.
fmc_fpga_ready	In	Async	Signal driven by fixed logic to indicate that all clocks and power supplies required by the FPGA are active and stable.
fmc_jc_srcclk	Out	N/A	Optional output signal which can be driven out from the CLIP to be used as the source clock for the jitter cleaner (useful for recovered-clock applications where the interface IP produces a clock that must be externally filtered for reuse as a reference clock). The source clock to be driven out to the jitter cleaner is selected through the **External Clock Configuration** properties window.

Programming

This section provides reference information about programming the ATCA-3671 module and adapter modules.



Tip NI requires a 64-bit operating system and recommends that you have 8 GB or more of system memory for compiling LabVIEW FPGA VIs for the module.

NI ATCA FPGA Modules VIs

This section contains technical and programming support for using the NI ATCA LabVIEW API. This section provides reference material for the NI ATCA VIs.

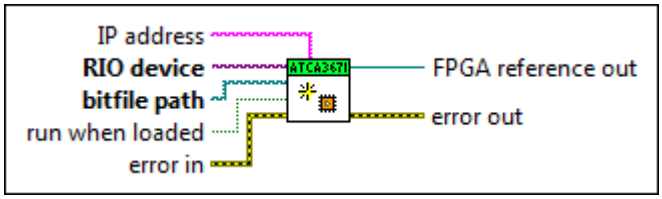
Use the NI ATCA VIs to develop applications for your ATCA-3671 module and adapter modules.

Palette Object	Description
ATCA-3671 Open FPGA Reference	Is required to connect to an FPGA from the host. It calls the transfer VI underneath to send over t he bitstream, then initiates a normal FPGA open call.
ATCA-3671 Get FPGA ID	Queries the ATCA-3671 FPGA ID of the RIO devic e specified by the resource string. It is called by several built-in VIs, but also may be used directl y to programmatically determine the location of a given FPGA device in large applications.
ATCA-3671 Transfer Bitstream	Initiates a connection to lvsrv and transfers the bitstream contents over the network when open and called by the ATCA-3671 Open FPGA Referen ce VI.
ATCA-3671 Configure Jitter Cleaner	Sends a command using lvsrv to set the jitter cle aner parameters and returns whether the PLL is locked after reprogramming.
ATCA-3671 Configure RTM Line Rate	Configures the retimers on the QSFP ports of th e RTM-3661 to run at the given line rate. Each ca

	ll affects a single lane, so the VI must be called i n a loop to configure entire ports at a time.
ATCA-3671 Remote Bitstream Download	Triggers a remote download of a `.lvbitx` file (LabVIEW FPGA bitstream) that is currently in me mory on the ATCA System-on-Module (SOM) ont o one of the ATCA-3671 FPGAs, reprogramming t he device.
3681 Generic Config	Reads/writes a configuration register of the atta ched AIO-3681 module.
3681 Gain Config	Reads/writes the gain range configuration settin g of an attached AIO-3681 module.
3682 Generic Config	Reads/writes a configuration register of the atta ched AIO-3682 module.
3682 Gain Config	Reads/writes the gain range configuration settin g of an attached AIO-3682 module.
Route GLCK	Configures the GCLK multiplexer routing betwe en the four ATCA FPGAs.

ATCA-3671 Open FPGA Reference VI

Opens a reference to an FPGA device on the ATCA-3671. It must be used in place of the standard Open FPGA Reference VI due to the unique nature of the ATCA-3671 hardware platform, which requires that the FPGA bitstream be transferred over the network to the embedded controller on the ATCA-3671 before reprogramming can occur.



RIO device specifies the RIO resource name of the ATCA-3671 FPGA module to be opened.

IP address specifies the IP address of the ATCA-3671 on the network.

bitfile path specifies the filename of the LabVIEW FPGA bitstream to be opened on the ATCA-3671 FPGA device.

run when loaded specifies whether or not to run the bitstream once it is transferred to the FPGA.

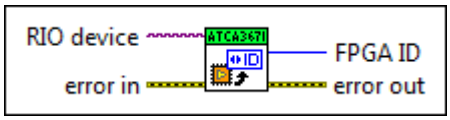
error in describes error conditions that occur before this node runs. This input provides standard error in functionality. If **error in** indicates an error, the node doesn't execute.

FPGA reference out returns a reference to a bitfile.

error out contains error information. This output provides standard error out functionality.

ATCA-3671 Get FPGA ID VI

Queries the ATCA-3671 FPGA ID of the RIO device specified by the resource string. It is called by several built-in VIs, but also may be used directly to programmatically determine the location of a given FPGA device in large applications.



RIO device specifies the RIO device name of the ATCA-3671 FPGA to query.

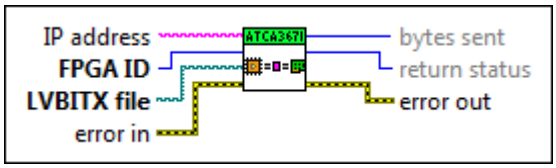
error in describes error conditions that occur before this node runs. This input provides standard error in functionality. If **error in** indicates an error, the node doesn't execute.

FPGA ID specifies the ATCA-3671 FPGA ID of the queried device.

error out contains error information. This output provides standard error out functionality.

ATCA-3671 Transfer Bitstream VI

Transfers a `.lvbitx` file (LabVIEW FPGA bitstream) onto the ATCA-3671 over the network. Transferring the file is a required operation before a new FPGA bitstream can be programmed onto the device. Therefore, this VI is called by ATCA-3671 Open FPGA Reference VI and should not be called separately in the user's diagram.



IP address specifies the IP address of the ATCA-3671 on the network.

FPGA ID specifies the ATCA-3671 FPGA ID of the device which should be assigned the new bitstream.

LVBITX file specifies the filename of the `.lvbitx` file used for transferring FPGA bitstream data.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality. If **error in** indicates an error, the node doesn't execute.

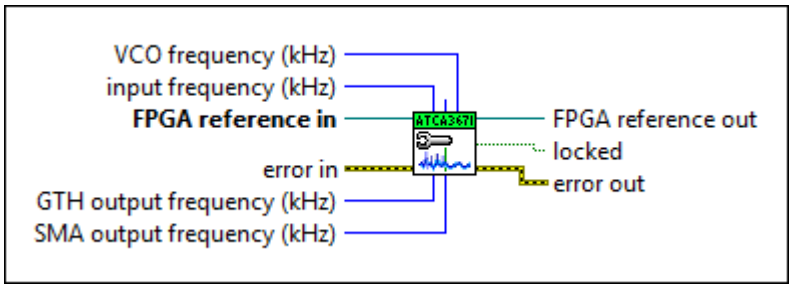
bytes sent reports the number of bytes successfully sent over the network. It is not necessary to check this output under normal circumstances, but it can provide information in case of networking problems.

return status reports the error status returned by the ATCA-3671 firmware. Non-zero values indicate invalid parameters, even if the operation itself did not return an error in **error out**.

error out contains error information. This output provides standard error out functionality.

ATCA-3671 Configure Jitter Cleaner VI

Used to configure the TI LMK04808 jitter cleaning PLL on the ATCA-3671 with a full set of input and output frequencies. All frequency values are provided in kHz, and a **locked** output is returned which indicates the lock state of the PLL after all corresponding register values have been programmed into the device. Ensure all values match the configuration of the FPGA target and any other externally-connected hardware.





VCO frequency (kHz) defines the frequency, in kHz, to be used for the voltage controlled oscillator (VCO) in the second stage PLL of the jitter cleaner. It must be within the valid range for the device type (for example, the LMK04808 requires a VCO between 2.75 GHz and 3.072 GHz) and achievable with a valid combination of multiplier/divider settings. The actual settings used on the device are automatically calculated by software.



input frequency (kHz) defines the input frequency, in kHz, being driven into the jitter cleaner. **input frequency (kHz)** should be set to match both parameters of the FPGA target in the External Clock Configuration properties and any other equipment connected to the hardware device.



FPGA reference in references the desired FPGA.



error in describes error conditions that occur before this node runs. This input provides

standard error in functionality. If **error in** indicates an error, the node doesn't execute.

GTH output frequency (kHz) determines the frequency, in kHz, to be generated for the GTH transceiver reference clock outputs on the device. Valid values are obtained by dividing the VCO frequency with an integer value between 1 and 1,045.

SMA output frequency (kHz) determines the frequency, in kHz, to be generated for the external SMA clock output on the device. Valid values are obtained by diving the VCO frequency with an integer value between 1 and 1,045.

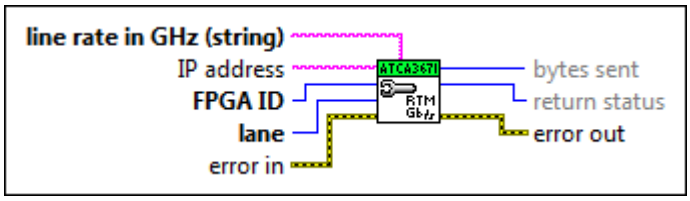
FPGA reference out references the desired FPGA.

locked reads the state of the lock detect pin on the jitter cleaner after completing all the register writes necessary to set the input and output frequencies. If the jitter cleaner is not locked, verify that the physical hardware connections and input values are correct.

error out contains error information. This output provides standard error out functionality.

ATCA-3671 Configure RTM Line Rate VI

Configures the retimers on the QSFP ports of the RTM-3661 to run at the given line rate. Each call only affects a single lane, so this VI must be called in a loop to configure entire ports at a time. Communication is performed directly with the ATCA-3671 over the network and does not involve the FPGA device itself.





IP address specifies the IP address of the ATCA-3671 on the network.



FPGA ID specifies the corresponding FPGA ID of the lane to be configured.



lane specifies the lane number to be configured (see documentation for mapping between lane numbers and ports).



line rate in GHz (string) specifies the line rate to be assigned to the retimer lane as a floating point string in GHz.



error in describes error conditions that occur before this node runs. This input provides standard error in functionality. If **error in** indicates an error, the node doesn't execute.



bytes sent reports the number of bytes successfully sent over the network. It is not necessary to check this output under normal circumstances, but it can provide information in case of networking problems.



return status reports the status code returned by the ATCA-3671 firmware. Non-zero values indicate invalid parameters, even if the operation itself did not return an error in **error out**.



error out contains error information. This output provides standard error out functionality.

ATCA-3671 Remote Bitstream Download VI

Triggers a remote download of a `.lvbitx` file (LabVIEW FPGA bitstream) that is currently in memory on the ATCA System-on-Module (SOM) onto one of the ATCA-3671 FPGAs, reprogramming the device. This VI should only be used when the ATCA is being used as a stand-alone target without a live PCIe host. Using this VI to reprogram FPGAs that have a live PCIe connection may lead to system instability.





IP address sets the IP address of the ATCA-3671 on the network.



FPGA ID specifies the ATCA-3671 FPGA ID of the device which should be assigned the new bitstream.



error in describes error conditions that occur before this node runs. This input provides standard error in functionality. If **error in** indicates an error, the node doesn't execute.



return status returns the status code of the ATCA-3671 firmware. Non-zero values indicate invalid parameters, even if the operation did not return an error from **error out**.



error out contains error information. This output provides standard error out functionality.

3681 Generic Config VI

Reads/writes a configuration register of the attached AIO-3681 module. This VI should be placed within an SCTL at or below 250 MHz. Accessing this setting requires multiple executions of the VI. Use hand shaking for valid execution. Only use one AIO Config VI per FPGA.





register specifies the address of the register to read/write to.



register value in specifies the value to write to the given register.



input valid allows you to specify TRUE to start a new register read/write transaction. A new read/write only occurs if this parameter is set to

TRUE when **ready for input** was TRUE the previous cycle.

write (T) Specifies whether the register value is modified. If TRUE, **register value in** will be written to the specified register on the AIO-3681 module. If FALSE, the existing register value will be read.

register value out indicates the actual register value of the AIO-3681 module.

output valid indicates whether the last requested register access has completed and the value of **register value out** is valid.

ready for input indicates when the VI is in an idle state and ready to perform a new register read/write.

3681 Gain Config VI

Reads/writes the gain range configuration setting of an attached AIO-3681 module. This VI should be placed within an SCTL at or below 250 MHz. Accessing this setting requires multiple executions of the VI. Use hand shaking for valid execution. Only use one AIO Config VI per FPGA. To access additional ADC configuration options, such as mode configuration and calibration, use the 3681 Generic Config VI.



gain range (register value) specifies the gain range setting to save to the AIO-3681 module. Use the 3681 Encode Gain Selection VI in order to convert a mV range to the appropriate register value.

channel specifies which input channel(s) to read/write the gain setting of.

input valid allows you to specify TRUE to start a new gain setting read/write transaction. A new read/write only occurs if this parameter is set to

TRUE when **ready for input** was TRUE the previous cycle.

write (T) specifies whether the register value is modified. If TRUE, **gain range (register value)** is written to the AIO-3681 gain range setting. If FALSE, the existing register value is read.

actual gain range (register value) indicates the actual gain range setting of the AIO-3681 module. Index 0 and 1 correspond to channels I and Q, respectively. Channels not specified for configuration report a value of 0. Use the 3681 Decode Gain Selection VI to convert the register values to the corresponding gain range in mV.

output valid indicates whether the last requested gain configuration has completed and whether the values of **actual gain range (register value)** are valid.

ready for input indicates when the VI is in an idle state and ready to perform a new gain configuration read/write.

3682 Generic Config VI

Reads/writes a configuration register of the attached AIO-3682 module. This VI should be placed within an SCTL at or below 250 MHz. Accessing this setting requires multiple executions of the VI. Use hand shaking for valid execution. Only use one AIO Config VI per FPGA.



register specifies the address of the register to read/write to.

channel specifies which channel of the AIO-3682 module the register is accessed from.

register value in specifies the value to write to the given register.



input valid allows you to specify TRUE to start a new gain setting read/write transaction. A new read/write will only occur if this parameter is set to TRUE when **ready for input** was TRUE the previous cycle.



write (T) specifies whether the register value is modified. If TRUE, **register value in** is written to the specified register on the AIO-3682 module. If FALSE, the existing register value is read.



register value out indicates the actual register values of the AIO-3682 module. Index 0 and 1 correspond to channels 0 and 1 respectively. Channels not specified for configuration will report a value of 0.



output valid indicates whether the last requested register access has completed and the value of **register value out** is valid.



ready for input indicates when the VI is in an idle state and ready to perform a new register read/write.

3682 Gain Config VI

Reads/writes the gain range configuration setting of an attached AIO-3682 module. This VI should be placed within an SCTL at or below 250 MHz. Accessing this setting requires multiple executions of the VI. Use hand shaking for valid execution. Only use one AIO Config VI per FPGA. To access additional AIO configuration options, use the 3682 Generic Config VI for all configuration purposes.





gain range (register value) specifies the gain range setting to save to the AIO-3682 module. Use 3682 Encode Current Selection VI in order to convert a mA range to the appropriate register value.



channel specifies which input channel(s) to read/write the gain setting of.



input valid specify TRUE in order to start a new gain setting read/write transaction. A new read/write will only occur if this parameter is set to TRUE when **ready for input** was TRUE the previous cycle.



write (T) specifies whether the register value is modified. If TRUE, **gain range (register value)** input is written to the AIO-3682 module gain range setting. If FALSE, the existing register value is read.



actual gain range (register value) indicates the actual gain range setting of the AIO-3682 module. Index 0 and 1 correspond to channels 0 and 1 respectively. Channels not specified for configuration will report a value of 0. Use the 3682 Decode Current Selection VI to convert the register values to the corresponding gain range in mA.



output valid indicates whether the last requested gain configuration has completed and the values of **actual gain range (register value)** are valid.



ready for input indicates when the VI is in an idle state and ready to perform a new gain configuration read/write.

Route GLCK VI

Configures the GCLK multiplexer routing between the four ATCA FPGAs.





remote credentials specifies the credentials used to access the ATCA SOM remotely.



source FPGA specifies the ATCA FPGA <A..D> that will drive the specified GCLK lane.



target FPGA specifies the ATCA FPGA <A..D> that will read the specified GCLK lane.



GLCK lane specifies the GCLK IO lane <0..3> to route.



error in describes error conditions that occur before this node runs. This input provides standard error in functionality. If **error in** indicates an error, the node doesn't execute.



error out contains error information. This output provides standard error out functionality.

Using the Standard Functionality for error in Parameters

Many LabVIEW nodes such as VIs contain **error in** parameters you can use to manage errors. These parameters typically provide the same, standard functionality. When a node exhibits different parameter functionality, the exceptions are documented in the reference material for that node. Standard **error in** behavior is as follows



Note Some nodes, such as error handling VIs, contain an **error in** parameter that does not provide standard error in functionality, but that contains an **error in** cluster that is standard.



error in describes error conditions that occur before this node runs. The default is `no error`. If an error occurred before this node runs, the node passes the **error in** value to **error out**. This node runs normally only if no error occurred before this node runs. If an error occurs while this node runs, it runs normally and sets its own error status in **error out**.

Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

The **error in** cluster contains the following cluster elements:



status is TRUE (X) if an error occurred before this node ran or FALSE (checkmark) to indicate a warning or that no error occurred before this node ran. The default is FALSE.



code is the error or warning code. The default is 0. If **status** is TRUE, **code** is an error code. If **status** is FALSE, **code** is 0 or a warning code.



source specifies the origin of the error or warning and is, in most cases, the name of the node that produced the error or warning. The default is an empty string.

Using the Standard Functionality for error out Parameters

Many LabVIEW nodes such as VIs contain an **error out** parameter you can use to manage errors. These parameters typically provide the same, standard functionality. When a node exhibits different parameter functionality, the exceptions are documented in the reference material for that node.

Standard **error out** functionality is as follows:



error out contains error information. If **error in** indicates that an error occurred before this VI ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** front panel indicator and select **Explain Error** from the shortcut menu for more information about the error.

error out contains the following cluster elements:



status is TRUE (X) if an error occurred before this node ran or during the running of this node or FALSE (checkmark) to indicate a warning or that no error occurred before this node ran or during the running of this node.



code is the error or warning code. If **status** is TRUE, **code** is an error code. If **status** is FALSE, **code** is 0 or a warning code.



source specifies the origin of the error or warning and is, in most cases, the name of the node that produced the error or warning.

© 2023 National Instruments Corporation.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP