PXIe-6943
32 Channel Digital Test Instrument

Publication No. 981052   Rev. A (03-08-2021)

Astronics Test Systems Inc.
4 Goodyear, Irvine, CA 92618
Tel: (877) 388-1952, (949) 859-8999; Fax: (949) 859-7139

https://www.ni.com/en-us/contact-us.html

THANK YOU FOR PURCHASING THIS

ASTRONICS TEST SYSTEMS PRODUCT

For this product, or any other Astronics Test Systems product that incorporates software drivers, you may access our web site to verify and/or download the latest driver versions. The web address for driver downloads is:

http://www.astronicstestsystems.com/support/downloads

If you have any questions about software driver downloads or our privacy policy, please contact us at:

atsinfo@astronics.com

WARRANTY STATEMENT

All Astronics Test Systems products are designed to exacting standards and manufactured in full compliance to our AS9100 Quality Management System processes.

This warranty does not apply to defects resulting from any modification(s) of any product or part without Astronics Test Systems express written consent, or misuse of any product or part. The warranty also does not apply to fuses, software, non-rechargeable batteries, damage from battery leakage, or problems arising from normal wear, such as mechanical relay life, or failure to follow instructions.

This warranty is in lieu of all other warranties, expressed or implied, including any implied warranty of merchantability or fitness for a particular use. The remedies provided herein are buyer's sole and exclusive remedies.

For the specific terms of your standard warranty, contact Customer Support. Please have the following information available to facilitate service.

1. Product serial number
2. Product model number
3. Your company and contact information

You may contact Customer Support by:

E-Mail:	https://www.ni.com/en-us/contact-us.html	
Telephone:	+1 877 388 1952	(USA)

RETURN OF PRODUCT

Authorization is required from Astronics Test Systems before you send us your product or sub-assembly for service or calibration. Call or contact Customer Support at 1-877-388-1952 or fax to 1-888-484-4436. We can also be reached at: https://www.ni.com/en-us/contact-us.html.

If the original packing material is unavailable, ship the product or sub-assembly in an ESD shielding bag and use appropriate packing materials to surround and protect the product.

PROPRIETARY NOTICE

TRADEMARKS AND SERVICE MARKS

DISCLAIMER

FOR YOUR SAFETY

Before undertaking any troubleshooting, maintenance or exploratory procedure, read carefully the **WARNINGS** and **CAUTION** notices.

This equipment contains voltage hazardous to human life and safety, and is capable of inflicting personal injury.

If this instrument is to be powered from the AC line (mains) through an autotransformer, ensure the common connector is connected to the neutral (earth pole) of the power supply.

Before operating the unit, ensure the conductor (green wire) is connected to the ground (earth) conductor of the power outlet. Do not use a two-conductor extension cord or a three-prong/two-prong adapter. This will defeat the protective feature of the third conductor in the power cord.

Maintenance and calibration procedures sometimes call for operation of the unit with power applied and protective covers removed. Read the procedures and heed warnings to avoid “live” circuit points.

Before operating this instrument:

1. Ensure the proper fuse is in place for the power source to operate.
2. Ensure all other devices connected to or in proximity to this instrument are properly grounded or connected to the protective third-wire earth ground.

If the instrument:

- fails to operate satisfactorily
- shows visible damage
- has been stored under unfavorable conditions
- has sustained stress

Do not operate until performance is checked by qualified personnel.

Table of Contents

List of Figures

List of Tables

Document Change History

Revision	Date	Description of Change
A	03/08/2021	Initial Release

This page was left intentionally blank.

Chapter 1
Overview and Features

This manual provides information necessary to set up and operate the PXIe 6943 32-Channel module. A single PXIe 6943 is referred to as a Digital Test Instrument (**DTI**). Up to thirteen DTIs can be coupled and synchronized as a Digital Test Suite (**DTS**).

Throughout this manual, "DTI" is used to refer to the PXIe 6943 and "DTS" is used to refer to two or more PXIe 6943s coupled together.

Introduction

The PXIe-6943 DTI (see Figure 1-1) is a high performance, 32-channel, 50 MHz digital I/O system. The instrument is designed to be used in a PXIe compatible mainframe and conforms to PXI Express Hardware Specification Revision 1.1.

Figure 1-1 PXIe-6943 DTI

The DTI provides a high speed data sequencer and 32 high-performance digital I/O channels in a space-saving, single-wide PXIe module. The DTI operates at data rates up to 50 MHz with 1 ns edge placement and less than 3 ns channel-to-channel skew.

Designed for High Reliability

The comprehensive thermal design ensures reliability with excellent cooling, monitoring, and protection. Each high-power module is equipped with a custom-designed heat sink to provide optimal cooling. An on-board temperature monitor protects the pin electronics devices from overheating and provides over-temperature shutdown.

Advanced Features for Modern Digital Test Development

The DTI is designed for today's challenging digital test system applications through innovative design. The flexible Field Programmable Gate Array (FPGA) design enables the DTI to meet special user and legacy requirements. The high-speed Data Sequencer provides control over test patterns, timing, and format.

Innovative Software Tools Speed Test Development

The API driver and Digital Test Instrument Layer support third-party test development tools to ease development and integration into popular test environments

Scalable Design

Built-in scalability and modular design enable linking of up to 13 modules to create a Digital Test Suite (DTS). Individual modules in the DTS can operate as an independent digital instrument or be combined as a digital subsystem.

High-Speed Data Sequencer

The high-speed data sequencer provides state-of-the-art control over digital test patterns. Sequencer logic supports full unit under test (UUT) handshaking and controls timing, format, pattern data, looping, and conditional testing. The sequencer includes definable, standby, and idle sequences.

Triggering and Synchronization

The DTI features extensive control over digital testing to synchronize with other test instruments and control digital test sequencing. The DTI accepts triggers from the PXI TTL Trigger Bus, front panel Auxiliary inputs, or from any channel, and provides two sync outputs per DTI. Triggers can be used to synchronize the DTI with other instruments and as a test input for test sequence control. Sync outputs can be offset with reference to the start of a sequence or a sequence step.

Instrument Soft Front Panel

The soft front panel software provides interactive control of the DTI. The intuitive graphical interface enables setup and configuration, calibration, and sequencer control. Channels may be set up either individually or in groups.

Automatic Test Program Generation (ATPG)

The optional ATPG provides an interface to IEEE-Std-1445 formatted files that can be generated from automatic test program generators such as LASAR to seamlessly integrate with the DTI. This interface provides the capability for the system to utilize the various features of IEEE-Std-1445 to support guided probe, fault dictionary, and complex patterns and timing set(s).

Migration Tools and Translators

The optional Migration Tools and Translators support many legacy test systems from a variety of manufacturers. Test programs from supported systems are easily translated without extensive code rewriting.

I/O Features

The DTI I/O features:

- Channels: 32 single-ended variable voltage.
- Dual threshold or differential comparator mode.
- Voltage range: -2 V to +7 V with an output swing of up to 9 V
- Relay Isolation on all 32 I/O channels.
- PMU Capabilties all 32 I/O channels.
- Programmable current load with dual commutating voltages
- Four Selectable output slew rates (0.2, 0.6, 0.9 and 1.2 V/ns typical)
- 50 Ohm output impedance
- Over-temperature detection/protection
- Over-voltage detection/protection
- Auxiliary channels (12):
    - Eight LVTTL
    - Four LVDS

Basic Elements of the DTI Module

As illustrated in Figure 1-1, the DTI module is comprised of the following major components; front panel, Digital Board, and a Driver/Receiver Board.

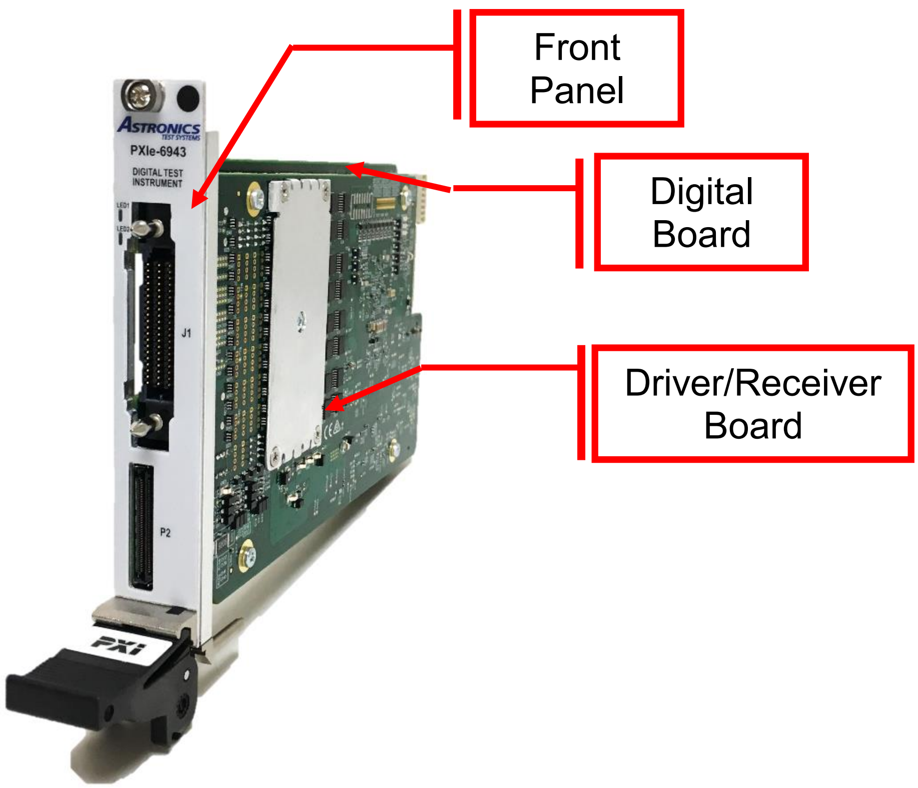


Figure 1-1 DTI Elements

Front Panel

The front panel contains two connectors (J1 and P2) as well as two LEDs (LED1 and LED2). J1 contains all the I/O from the driver/receiver card. P2 links multiple DTIs together using the External Timing Bus boards. LED1 and LED2 provides module status information.

Figure 1-2 illustrates the front panel's connectors and LED indicators.

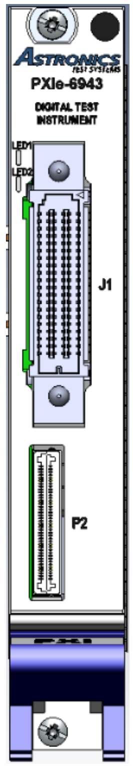


Figure 1-2 Front Panel Connectors and Indicators

Connectors

The J1 connector contains all the digital I/O test signals

The P2 connector connects multiple DTI modules together using the ETB accessory boards.

LED Indicators

LED	Indication
LED1	Red. Illuminated indicates module failure.
LED2	Green. Solid indicates FPGA loaded and module initialized. Flashing indicates module programming FPGA.

Table 1-1 PXIe-2461 LED Indicators

Digital Board

The digital board, hereafter referred to as DB, contains the connectors and headers required for routing signals to/from the PXIe backplane as well as the driver/receiver logic. The DB logic is comprised of the following major components.

DB Control

This contains the logic to communicate and program DTI resources.

PCI Express Bridge

The PCI Express Bridge maintains the interface with the backplane. The DTI is programmed through BAR0 register access.

External Timing Bus Control

In a multi-module system, the external timing bus (ETB) synchronizes the modules. The external timing bus control logic routes and terminates these signals.

Data Sequencer

The data sequencer provides the timing, memory and control for the driver/receiver board channels.

The Data Sequencer logic consists of the following:

- Timing Data (Phase Assert, Phase Return, Window Open, Window Close)
- Stimulus Format Code (Non Return, Return to Zero, etc.)
- Pattern Data (Output Levels, Input Compare, CRC Enable)
- Sequence Data (Pattern Period, Pattern Order, Looping, Conditional Testing)
- Result Data (Error Flags, Error Count, CRC per Channel, Record Memory)

Driver/Receiver Board

The driver receiver board, hereafter referred to as DR, contains all the driver/receiver logic, relays, sensors and termination circuitry for the I/O and AUX channels.

Accessories

The following section lists the accessories available and the order number.

ETB Links, Cables and Adapters

The following table lists the ETB links, cables and adapters and the ordering part number.

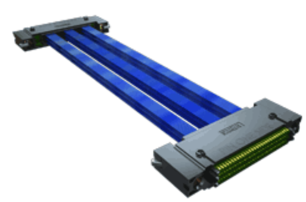
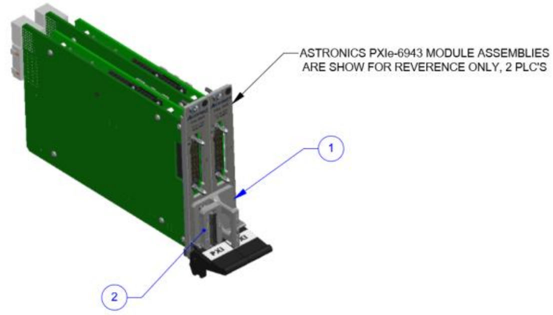
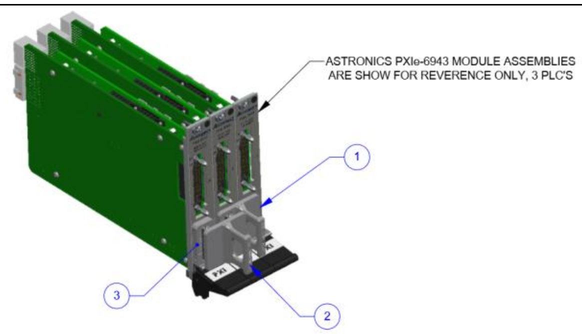
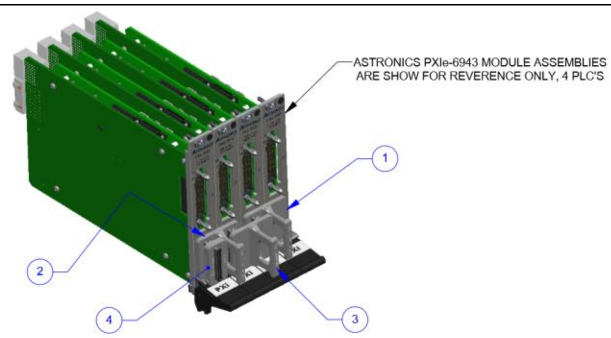
Description	Order #	Image
ETB primary link.	408812-001	
ETB secondary link.	408812-002	
ETB bridge link	408812-003	
ETB terminator link	408812-004	
50 ohm coax cable. Note 1	SEAC-020-06-XX.X-TU-TU	
50 ohm MCX breakout adapter.	405654	NA
VHDCI Dual Stack breakout adapter.	TBD	NA

Table 1-2 ETB Links, Cables and Adapter Accessories

Note 1: *In Table 1-2, XX.X is the length in inches. 6.0 Minimum.*

ETB Kits

The following table lists the ETB Kits and the ordering part number.

Number of Modules	Order #	Kit Contents
2	408886-002	 1. 408812-001 2. 408812-004
3	408886-003	 1. 408812-001 2. 408812-003 3. 408812-004
4	408886-004	 1. 408812-001 2. 408812-002 3. 408812-003 4. 408812-004

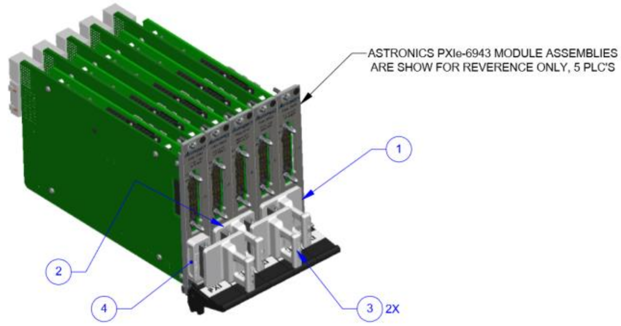
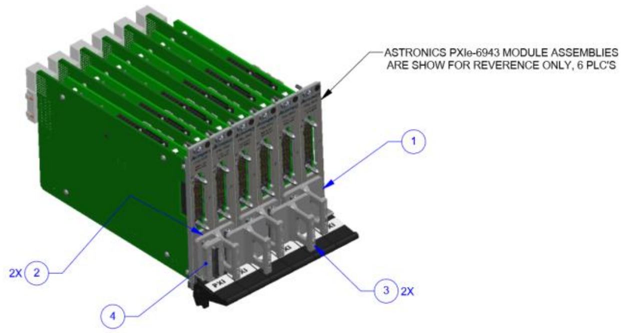
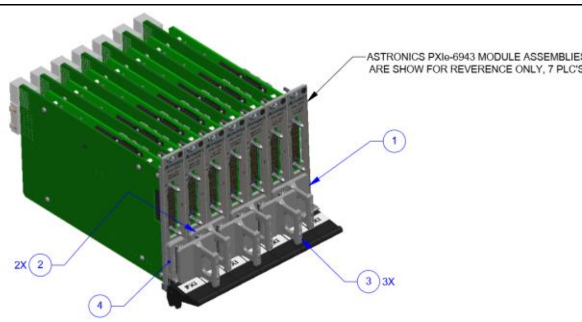
Number of Modules	Order #	Kit Contents
5	408886-005	 1. 408812-001 2. 408812-002 3. 408812-003 4. 408812-004
6	408886-006	 1. 408812-001 2. 408812-002 3. 408812-003 4. 408812-004
7	408886-007	 1. 408812-001 2. 408812-002 3. 408812-003 4. 408812-004

Table 1-3 ETB Kit Descriptions

Contact the factory for ETB kits for 8 to 13 modules.

This page was left intentionally blank.

Chapter 2
Getting Started

Unpacking and Inspection

WARNING

Use standard ESD procedures including ground straps and static-safe work surfaces whenever handling the PXIe-6943 module.

Remove the PXIe-6943 module and inspect it for damage. If any damage is apparent, inform the carrier immediately. Retain shipping carton and packing material for the carrier's inspection.

Verify that the pieces in the package you received contain the correct module option. Notify our Customer Support department (see front pages for contact information) if the module appears damaged in any way. Do not attempt to install a damaged module into a PXIe chassis.

The module is shipped in an anti-static bag to prevent electrostatic damage to the module. Do not remove the module from the anti-static bag unless it is in a static-controlled area.

Installing the Module(s) into a PXIe Chassis

WARNING

The PXIe-6943 module is NOT hot-swappable. The power to the PXIe chassis must be turned off before installing a PXIe-6943. Plugging the module in with chassis power on may result in damage.

The PXIe-6943 may be installed in any PXIe chassis hybrid or PXIe slot.

1. Insert module in the chassis with the PXI latch in the lower and open position.

Figure 2-1 Module Installation Step 1

2. Push the module forward, then lift the PXI latch to lock the module in place.

Figure 2-2 Module Installation Step 2

3. Tighten the retaining screws on the top and bottom of the module with a Philips screwdriver.

Figure 2-3 Module Installation Step 3

4. Install any remaining modules, making sure that modules that will be linked using the ETB connectors are in adjacent slots.

Figure 2-4 Module Installation Step 4

Installing the External Timing Bus (ETB) Links

Install the ETB links with the power off. ETB configuration is performed once on power up.

1. Install all first level links making sure all links are seated completely, see Figure 2-5 below.

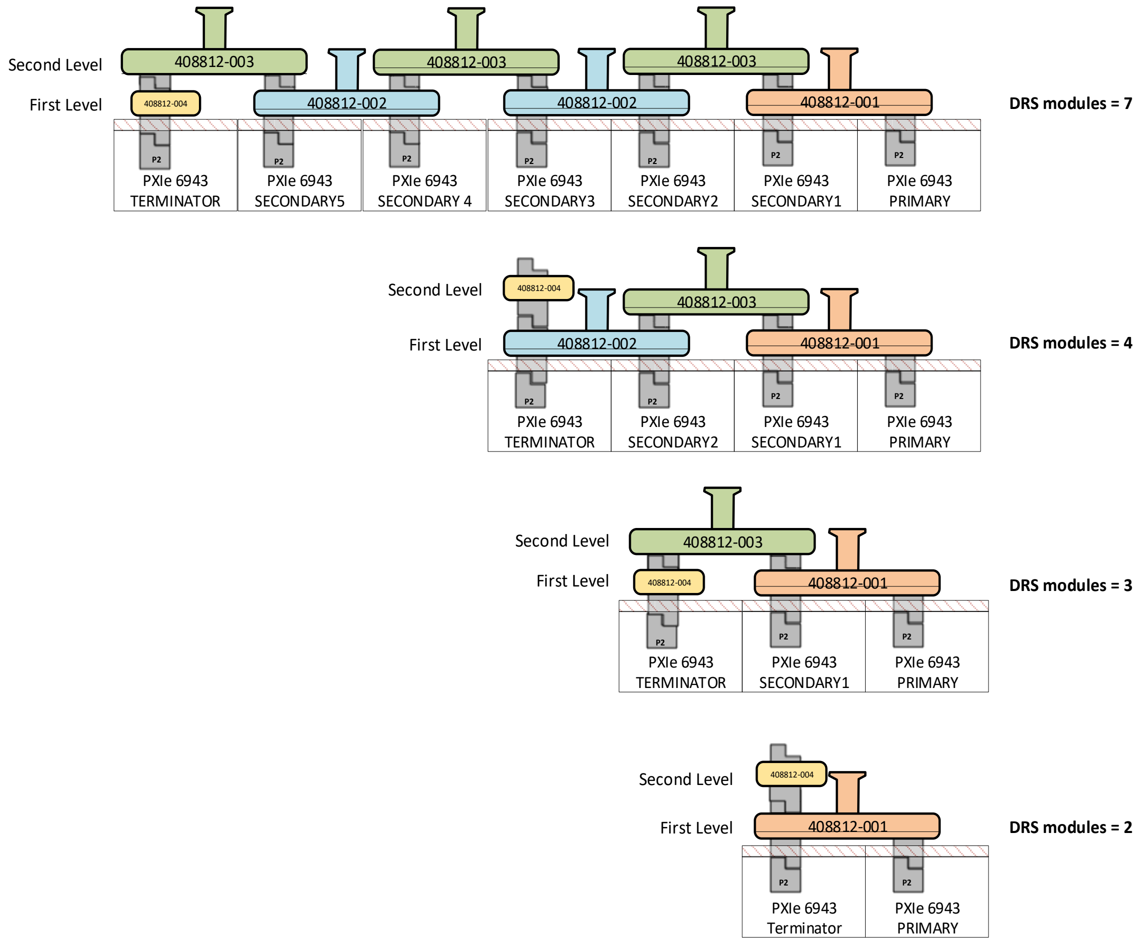


Figure 2-5 ETB Link Installation Levels

2. Install all second level ETB links making sure all links are seated completely.

Installing the J1 Cable

The J1 cable connects the 6943 I/O channels the device under test (DUT) via an adapter board. The cable has two 120 pin connectors that must be connected correctly, as shown in Figure 2-6, in order to prevent damage to the module or adapter.

1. Insert the cable connector with the “6943” label that contains the green PCBs into the J1 slot on the DTI module.

2. Insert the cable connector with the “UUT” label that contains the red PCBs into the adapter connector.

Figure 2-6 J1 Cable Connectors

WARNING

The J1 cable connector labeled “6943” with the green PCBs must be connected to the DTI and not the connector labeled “UUT” in order to prevent damage.

Initial Power On

Drivers must be installed prior to hardware installation (see Software Installation).

1. Once the module(s) and ETB boards are installed for the desired DTS configuration, turn on the chassis power.
2. Turn on or re-start the computer connected to the chassis.
3. The module will initialize and perform the internal power on self-test (POST). The green LED2 will flash during this step for ~14s.

   If the module fails POST then the green LED2 will continue the flash and the red LED1 will illuminate.

   If the module passes POST the green LED2 will stop flashing and be illuminated and the red LED1 will not be illuminated.
4. If installed, run the Soft Front Panel (SFP) program. The SFP will query the POST results and display any error codes for all installed DTI modules.

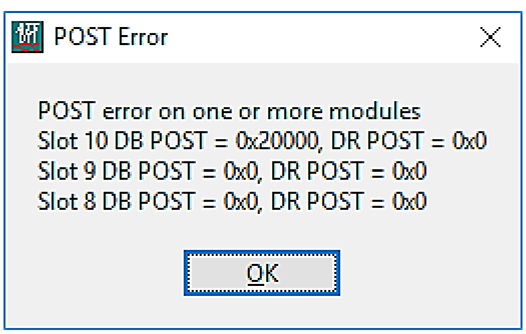


Figure 2-7 SFP POST Error Panel

The Digital Board (DB) POST codes are documented in Table 2-1 below.

DB POST Bit Number	Failure
0	I2C Bus 1. I2C Bus 1 controls: Frequency Synthesizers Voltage/Current Monitors Temperature Monitor
1	Frequency Synthesizer I/O
2	Temperature Monitor I/O.
3	Voltage/Current Monitor Initialization.
4	I2C Bus 1. I2C Bus 1 controls: Sequencer FPGA Temperature Monitor
5	Sequencer FPGA Temperature Monitor I/O.
6	SPI Flash I/O.
7	Sequencer FPGA Program.
8	Voltage/Current Monitor Update.
9	DR +12V Voltage/Current Level while reset Voltage < 11.0V Current > 1.0A
10	DR +3.3V Voltage/Current Level while reset Voltage < 3.0V Current > 0.3A
11	ETB FPGA Program.
12	Voltage/Current Monitor Update.
13	DR +12V Voltage/Current Level after reset Voltage < 11.0V Current > 2.0A
14	DR +3.3V Voltage/Current Level while reset Voltage < 3.0V Current > 4.1A
15	NVM Flash I/O
16	Frequency Synthesizer I/O
17	500MHz Clock
18	Sequencer Input Delay Control
19	Pattern Memory Bank 1 Request
20	Pattern Memory Bank 2 Request
21	Pattern Memory Bank 3 Request
22	Pattern Memory Bank 4 Request
23	Probe Memory Request
24	Driver Board I/O

Table 2-1 DB POST Error Description

The Driver Board (DR) POST codes are documented in Table 2-2 below.

DR POST Bit Number	Failure
0	I2C Bus. I2C Bus controls: Voltage/Current Monitors VEE DAC HV_VCC DAC
1	ADC SPI Initialization
2	SPI Flash Initialization.

DR POST Bit Number	Failure
3	Voltage/Current Monitor Initialization.
4	Voltage/Current Monitor Update.
5	DR +3.3V Voltage/Current Level Voltage < 3.2V Current > 1.0A
6	DR +5.0V Voltage/Current Level Voltage < 4.75V Current > 1.0A
7	VDD Regulator
8	CH1-CH4 Chip
9	CH5-CH8 Chip
10	CH9-CH12 Chip
11	CH13-CH16 Chip
12	CH17-CH20 Chip
13	CH21-CH24 Chip
14	CH25-CH28 Chip
15	CH29-CH32 Chip
16	Calibration Flash I/O
17	NVM Flash I/O
22	FPGA/Processor

Table 2-2 DR POST Error Description

Should the module fail POST, turn the chassis power off, re-install or make certain the PXIe-6943 is properly installed in the chassis, and turn the chassis power back on. If the module continues to fail, contact Customer Support.

Software Installation

Prior to hardware installation of the PXIe-6943, install the following four software drivers:

- VISA Driver
- C API Instrument Driver
- LabView Instrument Driver
- Low Level VISA Driver

VISA Driver

The C Legacy API and LabView Instrument Drivers use the VISA communication library to operate the instrument. The VISA library must be installed prior to installing the instrument driver and is supplied by the PCI Express interface manufacturer.

Installing the VISA Driver

Follow the manufactures setup instructions.

C API Instrument Driver

The C API instrument driver links the communication interface and an application development environment (ADE). It provides a high level, abstract view of the instrument. It also provides ADE-specific information that supports the capabilities of the ADE, such as a graphical representation.

Listed below are some of the ADEs that are recommend for use with this driver:

- Agilent Technologies Agilent VEE
- Microsoft Visual Basic
- Microsoft Visual C/C++
- Microsoft Visual C#/.Net
- National Instruments LabWindows/CVI

Included with the instrument driver is the Soft Front Panel (SFP) software. The soft front panel is a graphical user interface for the PXIe-6943. Use it to verify communications and functionality when the PXIe-6943 is first integrated into the system.

Download the C API Instrument driver from:

http://www.ni.com/gate/gb/GB_EVALTLKTFTICASTRONICS/US

Installing the API Driver

1. Open the file ri6943e_vXYZ.zip on the computer with the PCI Express interface. The XYZ refers to the driver version Z.YZ. For example, ri6943e_v100.zip is version 1.00 of the API driver.
2. Double-click the “setup.exe” file to execute the installer.
3. Follow the setup directions.

The following files are installed into the directory determined by the VXIPNPPATH windows environment variable.

- ANSI C source code for the Instrument Driver and Soft Front Panel, i.e., .c and .h files.
- MS Windows 32 bit DLL library, i.e., ri6943e_32.dll file.
- Microsoft 32 bit DLL import library, i.e., ri6943e.lib file.
- LabWindows/CVI function panel file, i.e., ri6943e.fp file.
- Driver help file, i.e., ri6943e.doc file.

LabView Instrument Driver

The LabView instrument driver supports LabView 2014 and later. Contact customer support for information on driver support for earlier versions of LabView.

Download the LabView Instrument driver from:

http://www.ni.com/gate/gb/GB_EVALTLKTFTICASTRONICS/US

Installing the LabView Instrument Driver

1. Open the download file on the computer with the PCI Express interface.
2. Double-click the “setup.exe” file to execute the installer.
3. Follow the setup directions.

Listed below are the example vi’s included with the LabView Driver in the Examples folder of the installation directory.

- ri6943e Main.vi – Soft Front Panel application.
- ri6943e Example configure and read 5 measurements.vi
- ri6943e Example Positive Pulsewidth Measurement.vi

Low Level VISA Driver

A low level VISA driver is required for the specific PCI Express interface. The NI-VISA low level driver is automatically installed with the LabView instrument driver installation.

Contact customer service if the PCI Express interface is not supported by NI-VISA

Installing the Low Level VISA Driver

If the LabView driver is not installed then the low level driver must be manually installed.

The low level driver is installed in a subfolder named “INF” in the API driver installation location.

You must have administrator privileges to install the low level driver.

1. From the explorer window, open the “INF” folder in the API installation location. There will be four files:

   - ats6943e.cat
   - ats6943e.inf
   - ats6943e.ini
   - ats6943e.txt

   The .cat file contains the driver signing data.

   The .inf file is the low level driver.

   The .ini is the PXI Module Description File

   The .txt file contains installation and removal instructions.

2. Right click on the .inf file and select **Install**.
3. Re-boot the computer when driver install is complete.

Chapter 3
Hardware Description

The PXIe-6943 DTI is comprised of two PCB's, the digital board (DB) and the driver/receiver (DR) and a front panel.

The block diagram in Figure 3-1 shows the DTI hardware connection to each other and the functional elements within each.

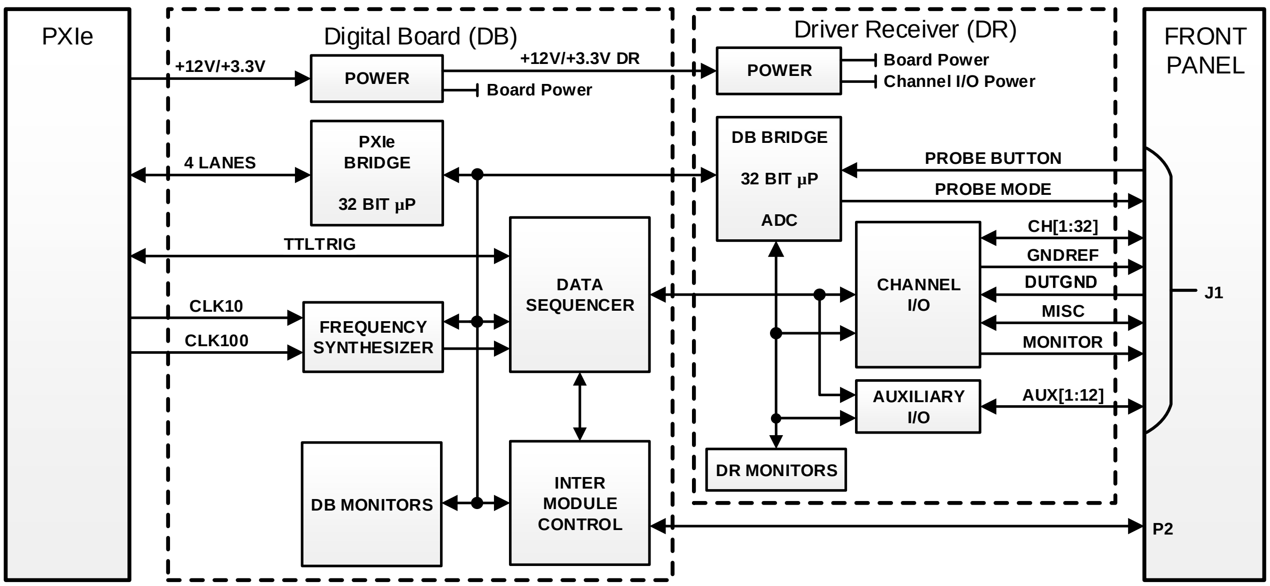


Figure 3-1 DTI Module Block Diagram

The following sections in this chapter describes the functional elements of each hardware block.

Front Panel

The front panel contains an external timing bus connector (P2) and a digital I/O connector (J1).

The P2 connector is used to Link multiple DTI modules using the ETB boards to create a DTS of up to 416 channels.

The J1 connector contains the digital I/O signal resources.

Table 3-1 lists the J1 part number. This same part must be used on the adapter board to route the DTI signals to the device under test.

Connector	Description	Part Number
J1	Digital I/O	SEAF-20-01-S-06-2-RA-LP-TR

Table 3-1 Front Panel J1 Connector Part Number

J1 Pinout and Signal Description

Figure 3-2 shows the J1 pin numbering.

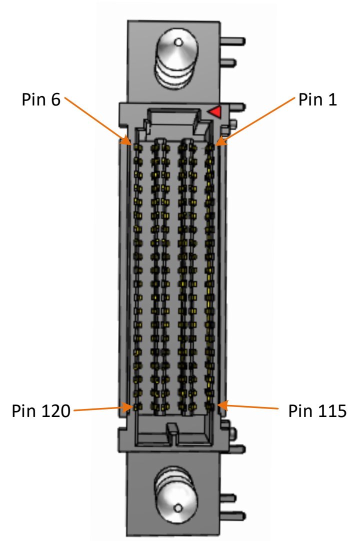


Figure 3-2 J1 Pin Numbering

Table 3-2 lists the J1 signal pinout at the front panel.

PIN-SIGNAL	PIN-SIGNAL	PIN-SIGNAL	PIN-SIGNAL	PIN-SIGNAL	PIN-SIGNAL
6-GND	5-GND	4-GND	3-GND	2-GND	1-GND
12-CH6	11-CH5	10-CH4	9-CH3	8-CH2	7-CH1
18-GND	17-GND	16-GND	15-GND	14-GND	13-GND
24-CH12	23-CH11	22-CH10	21-CH9	20-CH8	19-CH7
30-GND	29-GND	28-GND	27-GND	26-GND	25-GND
36-CH18	35-CH17	34-CH16	33-CH15	32-CH14	31-CH13
42-GND	41-GND	40-GND	39-GND	38-GND	37-GND
48-CH24	47-CH23	46-CH22	45-CH21	44-CH20	43-CH19
54-GND	53-GND	52-GND	51-GND	50-GND	49-GND
60-CH30	59-CH29	58-CH28	57-CH27	56-CH26	55-CH-25
66-GND	65-GND	64-GND	63-GND	62-GND	61-GND
72-AUX11P	71-AUX11N	70-AUX12P	69-AUX12N	68-CH32	67-CH31
78-GND	77-GND	76-GND	75-GND	74-GND	73-GND
84-AUX7	83-AUX8	82-AUX9P	81-AUX9N	80-AUX10P	79-AUX10N
90-GND	89-GND	88-GND	87-GND	86-GND	85-GND
96-AUX1	95-AUX2	94-AUX3	93-AUX4	92-AUX5	91-AUX6
102-GND	101-GND	100-GND	99-GND	98-GND	97-GND
108-BCLK	107-MFSIG	106-ARL1IN	105-IN3	104-IO2	103-IO1
114-GND	113-GND	112-GND	111-GND	110-GND	109-GND
120-DUTGND	119-MONITOR	118-+3.3V	117-PLED	116-PBUT	115-EXTFORCE

Table 3-2 J1 Pinout

Table 3-3 lists the signal descriptions.

Name	Pin No.	Description
CH1-CH32	Various	(Bi-directional) High speed variable voltage channels
GND	Various	Signal Ground reference
AUX12N	69	(Bi-directional) LVDS Negative I/O pin, 100 Ohm parallel 15K pullup to GND
AUX12P	70	(Bi-directional) LVDS Positive I/O pin, 100 Ohm parallel 15K pullup to +3.3V
AUX11N	71	(Bi-directional) LVDS Negative I/O pin, 100 Ohm parallel 15K pullup to GND
AUX11P	72	(Bi-directional) LVDS Positive I/O pin, 100 Ohm parallel 15K pullup to +3.3V
AUX10N	79	(Bi-directional) LVDS Negative I/O pin, 100 Ohm parallel 15K pullup to GND
AUX10P	80	(Bi-directional) LVDS Positive I/O pin, 100 Ohm parallel 15K pullup to +3.3V
AUX9N	81	(Bi-directional) LVDS Negative I/O pin, 100 Ohm parallel 15K pullup to GND
AUX9P	82	(Bi-directional) LVDS Positive I/O pin, 100 Ohm parallel 15K pullup to +3.3V
AUX8	83	(Bi-directional) LVTTL I/O pin, 50 Ohm series equivalent 10K pullup to +3.3V
AUX7	84	(Bi-directional) LVTTL I/O pin, 50 Ohm series equivalent 10K pullup to +3.3V
AUX6	91	(Bi-directional) LVTTL I/O pin, 50 Ohm series equivalent 10K pullup to +3.3V
AUX5	92	(Bi-directional) LVTTL I/O pin, 50 Ohm series equivalent 10K pullup to +3.3V
AUX4	93	(Bi-directional) LVTTL I/O pin, 50 Ohm series equivalent 10K pullup to +3.3V
AUX3	94	(Bi-directional) LVTTL I/O pin, 50 Ohm series equivalent 10K pullup to +3.3V
AUX2	95	(Bi-directional) LVTTL I/O pin, 50 Ohm series equivalent 10K pullup to +3.3V
AUX1	96	(Bi-directional) General Purpose LVTTL I/O pin, 50 Ohm series equivalent 10K pullup to +3.3V
IO1	103	(Bi-directional) Reserved
IO2	104	(Bi-directional) Reserved
IN3	105	(Input) Reserved
ARL1IN	106	(Input) Reserved
MFSIG	107	(Output) Multi-Function Signal
BCLK	108	(Output) Reserved
EXTFORCE	115	(Bi-directional) External Force routed to all of the Pin Electronics devices
PBUT	116	(Bi-directional) Reserved
PLED	117	(Output) Reserved
+3.3V	118	(Output) Reserved
MONITOR	119	(Output) Monitor signal from the Pin Electronics devices. Note: Only one channel can be selected at a time.
DUT_GND	120	(Input) DUT/UUT ground reference. All of the Pin Electronics devices have a UUT ground reference input that can be selected to be this signal or signal ground.

Table 3-3 J1 Signal Description

Mating Cable Pinout

The mating cable connects the DTI J1 signals to the user's adapter board that contains the same connector type as J1. Figure 3-3 below shows the mating cable (SEAC-020-06-XX.X-TU-TU).

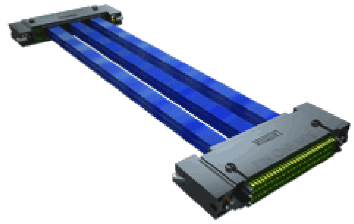


Figure 3-3 Mating Cable

Figure 3-4 illustrates the pin 1 locations on the J1 cable.

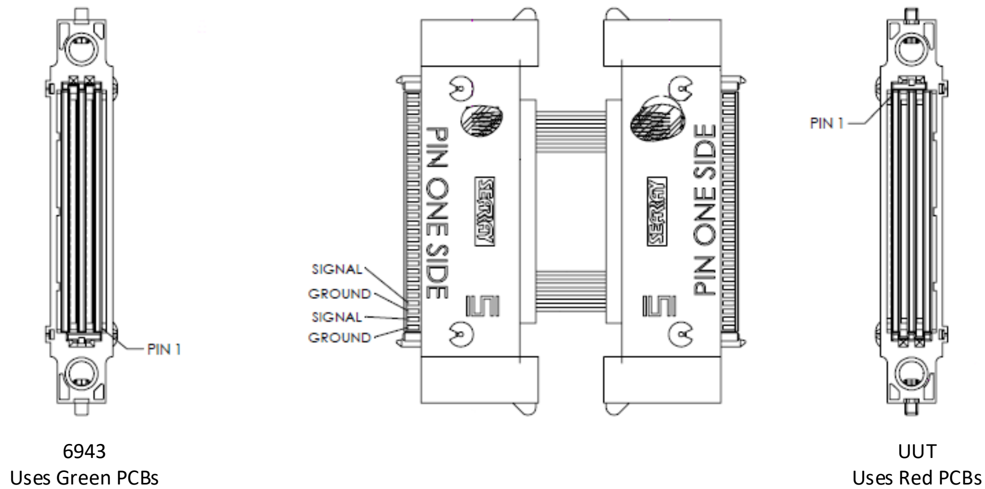


Figure 3-4 J1 Cable Pin 1

Table 3-4 lists the J1 cable pinout.

6943 Pin	1	2	3	4	5	6
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	115	116	117	118	119	120
6943 Pin	7	8	9	10	11	12
Signal	CH1	CH2	CH3	CH4	CH5	CH6
UUT Pin	109	110	111	112	113	114
6943 Pin	13	14	15	16	17	18
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	103	104	105	106	107	108
6943 Pin	19	20	21	22	23	24
Signal	CH7	CH8	CH9	CH10	CH11	CH12
UUT Pin	97	98	99	100	101	102
6943 Pin	25	26	27	28	29	30
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	91	92	93	94	95	96
6943 Pin	31	32	33	34	35	36
Signal	CH13	CH14	CH15	CH16	CH17	CH18
UUT Pin	85	86	87	88	89	90

6943 Pin	37	38	39	40	41	42
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	79	80	81	82	83	84
6943 Pin	43	44	45	46	47	48
Signal	CH19	CH20	CH21	CH22	CH23	CH24
UUT Pin	73	74	75	76	77	78
6943 Pin	49	50	51	52	53	54
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	67	68	69	70	71	72
6943 Pin	55	56	57	58	59	60
Signal	CH25	CH26	CH27	CH28	CH29	CH30
UUT Pin	61	62	63	64	65	66
6943 Pin	61	62	63	64	65	66
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	55	56	57	58	59	60
6943 Pin	67	68	69	70	71	72
Signal	CH31	CH32	AUX12N	AUX12P	AUX11N	AUX11P
UUT Pin	49	50	51	52	53	54
6943 Pin	73	74	75	76	77	78
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	43	44	45	46	47	48
6943 Pin	79	80	81	82	83	84
Signal	AUX10N	AUX10P	AUX9N	AUX9P	AUX8	AUX7
UUT Pin	37	38	39	40	41	42
6943 Pin	85	86	87	88	89	90
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	31	32	33	34	35	36
6943 Pin	91	92	93	94	95	96
Signal	AUX6	AUX5	AUX4	AUX3	AUX2	AUX1
UUT Pin	25	26	27	28	29	30
6943 Pin	97	98	99	100	101	102
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	19	20	21	22	23	24
6943 Pin	103	104	105	106	107	108
Signal	IO1	IO2	IN3	ARL1IN	MFSIG	BCLK
UUT Pin	13	14	15	16	17	18
6943 Pin	109	110	111	112	113	114
Signal	GND	GND	GND	GND	GND	GND
UUT Pin	7	8	9	10	11	12
6943 Pin	115	116	117	118	119	120
Signal	EXTFORCE	PBUT	PLED	+3.3V	MONITOR	DUTGND
UUT Pin	1	2	3	4	5	6

Table 3-4 J1 Cable Pinout

Digital Board (DB)

The DB controls 32 sets of digital I/O signals that are passed to the driver/receiver board for level translation. Each channel can be set to Dynamic HiZ, Dynamic

VTT or PMU modes.

Dynamic control is performed by data sequencer timing signals and can also be set to static. PMU control is performed by device settings on the driver/receiver board.

In addition to the 32 I/O channels, 12 general-purpose I/O signals can be used for dynamic pattern control or UUT signal emulation.

The following sections provide a description of the digital board hardware shown in Figure 3-1.

Power

This logic converts the PXI backplane +12V and +3.3V power to voltages required by the digital board as well as providing monitored +12V and +3.3V power to the driver board.

All voltages can be queried for voltage and current.

During power up the +12V and +3.3V signals from the PXIe connector are queried before and after the DR board is turned on to verify minimum voltage levels and maximum current draws are not exceeded before initializing the DR.

PXIe Bridge/32 Bit µP

This logic provides the interface to the PXIe bus that supports four lanes. It also includes the interfaces to communicate with the programmable logic on the digital board via standard (SPI, I2C) and custom (Data Sequencer CBUS) interfaces. The µP is used to reduce the PCI express traffic and allow for event monitoring for the module.

Frequency Synthesizer

This logic generates the fixed clocks for the digital and driver boards as well as the programmable frequency synthesizer (FS). The reference clock for the FS is selectable between the PXI CLK10, PXI CLK100 or any of the AUX inputs.

Data Sequencer

This logic provides the stimuli and control for the DR channel I/O and auxiliary I/O. The data sequencer block diagram is illustrated in Figure 3-5.

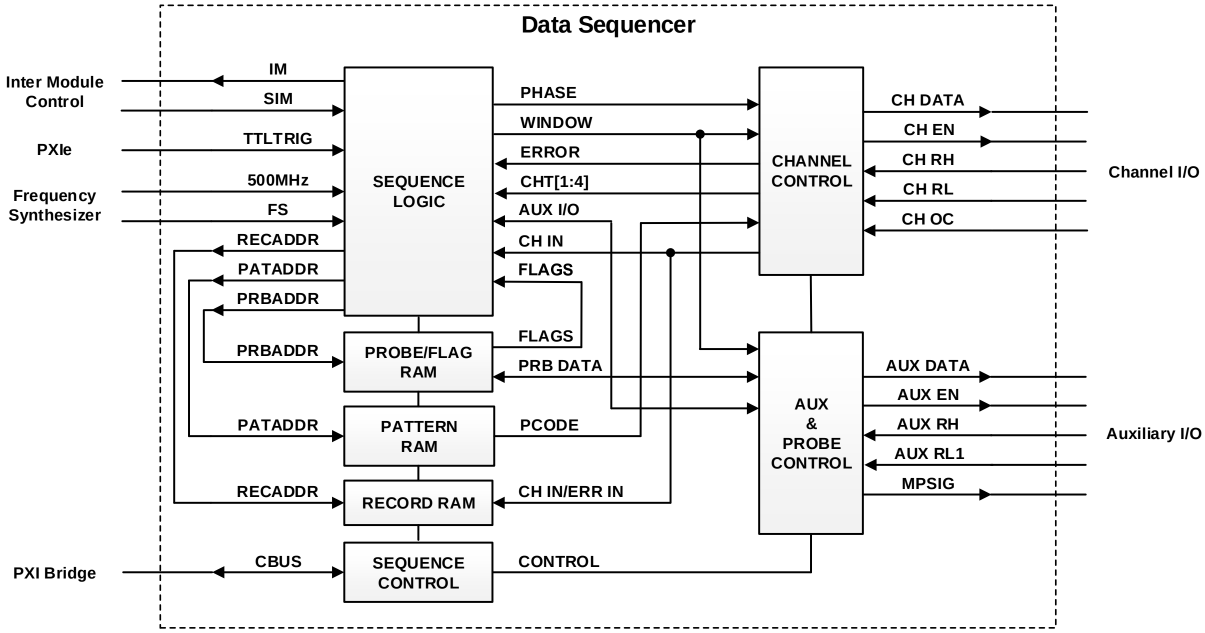


Figure 3-5 Data Sequencer Block Diagram

Sequence Logic

This logic generates the addresses for the external RAMs as well as performing the sequence flow. This logic also includes the counter timer and pulse generator resources. The sequence logic block diagram is illustrated in Figure 3-6.

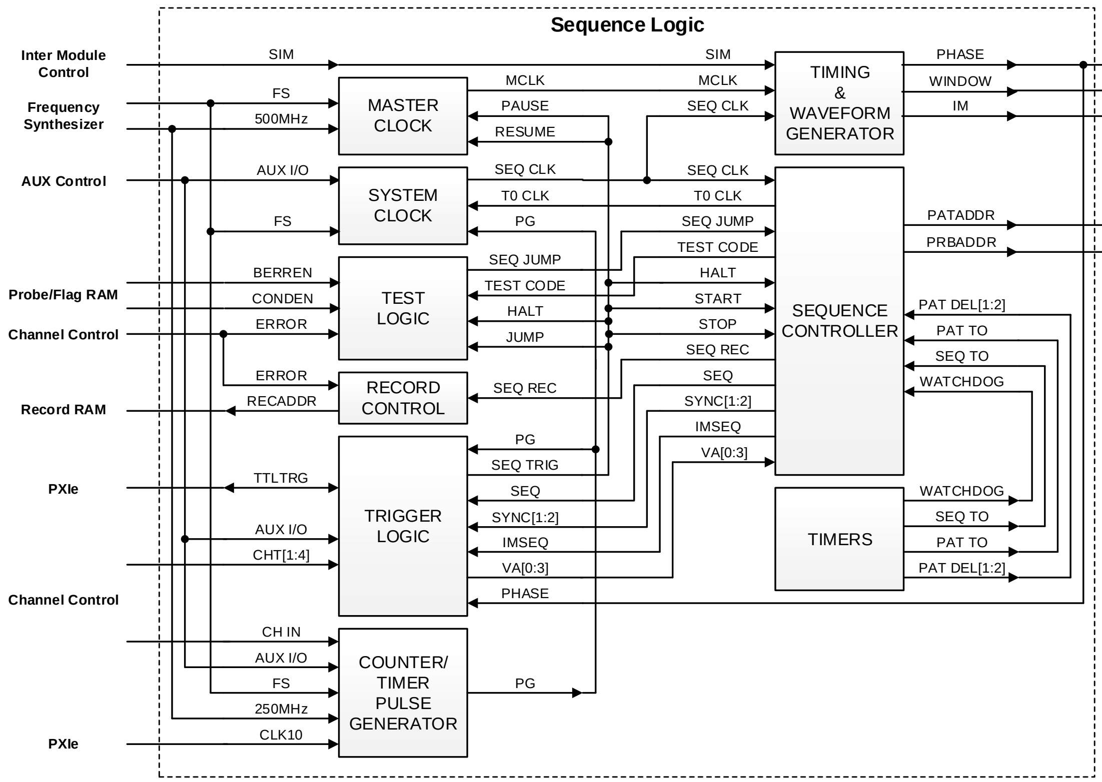


Figure 3-6 Sequence Logic Block Diagram

Master Clock

This block selects the master clock signal (MCLK) used by the timing and waveform generator.

System Clock

This block selects the sequence clock signal (SCLK) used by the sequence controller.

Test Logic

This block determines if a valid conditional jump is enabled or not.

Record Control

This block generates the address for the Record RAM based on the Recording Mode. This block also contains the error address memory, record index memory and burst error counter.

Trigger Logic

This block programs the various DTI triggers and assigns the source, test condition, inversion and edge detect clear. The DTI triggers are used to; enable sequence jumps; start and stop sequence execution; pause, halt and resume sequence execution.

Counter/Timer & Pulse Generator

The pulse generator can be used to generate triggers, system clock or as an AUX output signal.

The counter/timer can be used to measure frequency or time interval data several sources that includes any channel or AUX input.

Timing & Waveform Generator

This block contains the logic and memory for the phases, windows and waveforms.

Sequence Controller

This block contains the Sequence RAM which defines the order in which Patterns will be output/input. As such, this block provides the addressing to the Pattern RAM and the Record RAM. The Sequence RAM also contains the T0CLK period, Jump Type, Jump Addresses, looping controls/loop counts, Jump codes, CPP and other control bits for: Pause Code/Pause Resume Options, Record Capture type, Waveform control and Phase Trigger Type along with 2 Sequence Flags that can be output. The **Sequencer Operation** section of Chapter 6 provides detailed information on sequence operation.

Timers

This block contains the Watchdog, Sequence Timeout, Pattern Delay (2) and the Pattern Timeout Timers.

Probe/Flag RAM

The probe input code, probe results and CONDEN/BERREN data for each pattern is stored in this RAM.

Pattern RAM

The output code as well as the input code for every channel of each pattern is stored in the Pattern RAM.

Record RAM

This is where the individual channel results are stored. The channel results are either the pattern input compare result or raw response data based on RH or RL. The results can be stored in normal or indexed starting from address zero and expanded.

Sequence Control

This block contains the registers and logic used to program the data sequencer.

Channel Control

This block takes the output code from the pattern RAM, formats it and outputs it according to the phase timing (PHASE). The resultant drive (CH DATA) and enable (CH EN) signals go to the Driver/Receiver logic.

The response high (CH RH) and response low (CH RL) signals from the Receivers are examined, and then, based on the window timing (WINDOW), the response is analyzed with respect to the input code. The channel results are routed to the Record RAM. The cumulative Error signal goes to the Sequence Logic block so it can be used for Jumping, Halting and the Counting of Errors. Individual over-current (OC) signals from the Channel Drivers can also be processed by this block to disable the channel drivers if desired.

AUX & Probe Control

AUX control allows user and diagnostic signals to be input or output the AUX pins. The inputs go to the Sequence Logic block described above. There is also a Multi-purpose signal (MPSIG) which can be combined with other signals on the Driver/Receiver board and provided to the user on the J1 connector.

Probe expect data is received from the Probe/Flag RAM and result data is generated that is stored back into the Probe/Flag RAM.

DB Monitors

The digital board monitors consist of twelve voltage/current sensors that can be used to query the following voltages or currents:

- FPGA Vcc +1.0V
- FPGA AUX Vcc +1.8V
- FPGA I/O +3.3V
- GTP Regulator +1.4V
- FPGA GTP Vcc +1.0V
- FPGA GTP Vtt +1.2V
- Sequencer FPGA I/O +2.5V
- Sequencer Vcc +1.2V
- PXIe DB 3.3 +3.3V
- PXIe DB 12 +12.0V
- Front-end 3.3 +3.3V
- Front-end 12 +12.0V

Inter Module Control

The inter module control logic allows multiple modules to be linked so that the I/O channels and sequencers from up to 13 modules can be synchronized. One module, nomenclated the primary module, generates the timing bus signals used by the other modules that are linked to it. The timing bus signals broadcast to the linked modules via the external timing bus (ETB) on the front panel using ETB link connectors. Individual error and pass valid flags from each module are also transmitted via the ETB links back to the primary module. Figure 3-7 shows the ETB link connections to a multi-module system.

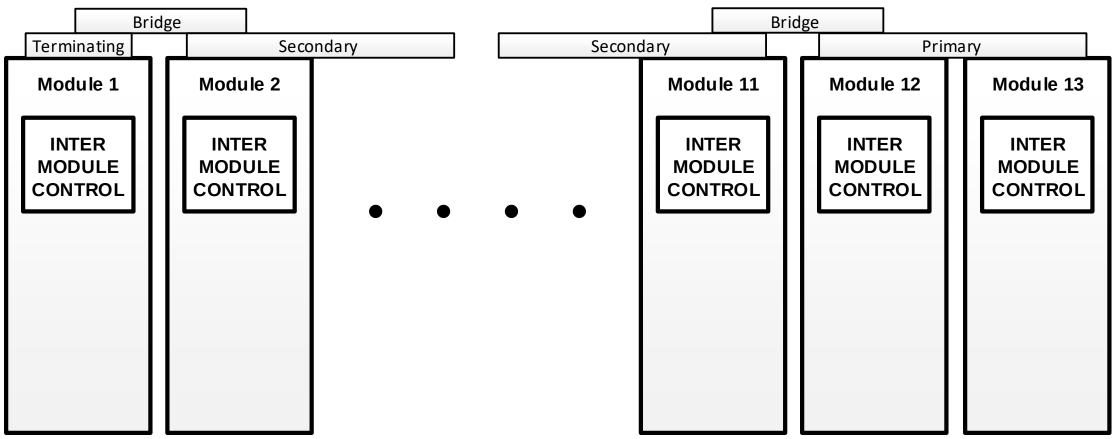


Figure 3-7 ETB Links Multi Module System

Figure 3-8 illustrates the inter module control logic.

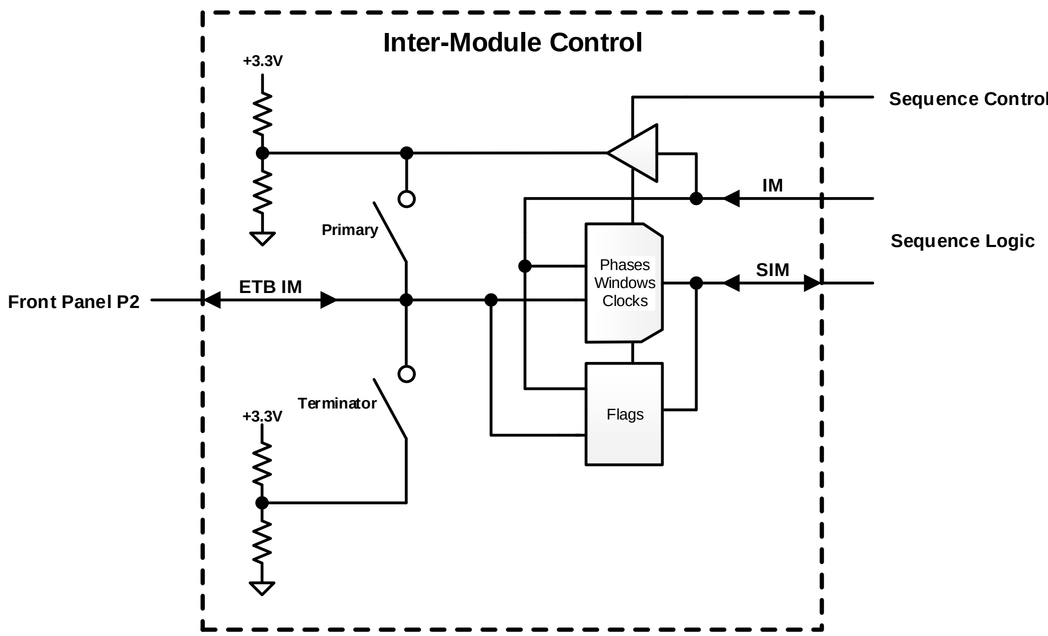


Figure 3-8 Inter Module Control Logic

The selected timing bus (SIM) is set to either the local timing bus (IM) or the ETB timing bus (ETB IM). Included in the logic are precise delay elements that align the inter module bus signals to reduce channel to channel skew.

Driver Receiver

The following sections provide a description of the driver receiver board hardware shown in Figure 3-1 at the beginning of this chapter.

Power

This logic converts the PXI backplane +12V and +3.3V power to voltages required by the driver board as well as providing the analog power (HV_VCC, VCC and VEE) to the I/O channels.

The analog power distribution is separated in two groups, the lower sixteen channels (HV_VCC1, VCC1 and VEE1) and the upper sixteen channels (HV_VCC2, VCC2 and VEE2). In order to reduce the power required, the HV_VCC is set based on user defined I/O max levels within each group.

DB Bridge/32 Bit µP ADC

This logic provides the interface to the digital board. It also includes the interfaces to communicate with the programmable logic on the driver board via standard (SPI, I2C) interfaces. The µP is used to reduce the PCI express traffic and allow for event monitoring for the module. The ADC is used for voltage monitoring, PMU readings and calibration.

Channel I/O

The channel I/O contains all the driver/receiver logic, relays, sensors and termination circuitry for channels 1 through 32.

Auxiliary I/O

The auxiliary I/O contains the logic and termination for twelve general purpose I/O signals. Table 3-5 lists the logic and termination of the twelve AUX signals.

Signal	Logic	Termination
AUX1 – AUX8	LVTTL	50Ω Series, 10K pullup to +3.3V
AUX9 – AUX12	LVDS	100Ω parallel, 15K pullup to +3.3 on pos., 15K pulldown to gnd on neg.

Table 3-5 AUX I/O Logic and Termination

DR Monitors

Voltage and current monitors are used to test for under voltage or over current conditions when the analog power is turned on. If thresholds listed below are exceeded, then analog power is turned off.

Signal	Voltage	Threshold
VEE1	-3.6	Voltage > -3.1
VEE2	-3.6	Voltage > -3.1
+3.3V	+3.3	Voltage < 3.2 or Current > 3.6
+5V	+5	Voltage < 4.7 or Current > 1.8
VCC1	+3.5	Voltage < 3.4 or Current > 0.7
VCC2	+3.5	Voltage < 3.4 or Current > 0.7
HV_VCC1	IO Max 1..16 + 1.5	Not Tested
HV_VCC2	IO Max 17..32 + 1.5	Not Tested

Table 3-6 Analog Voltage Power On Voltages and Thresholds

This page was left intentionally blank.

Chapter 4
Programming

This Chapter provides information for programming the PXIe-6943 module.

The Chapter includes:

- Soft Front Panel Description
- Programming Steps
- LabView vi's and API Library
- Examples of Use

Soft Front Panel

The Soft Front Panel (SFP) is a stand-alone executable.

The SFP panels and controls allow the user to query, program and execute DTI and DTS settings.

The SFP can be used as an aid to understanding the product as well as being useful in troubleshooting test development using configuration files for factory support.

Starting the Soft Front Panel

The SFP is installed with the API function driver and is located in the folder specified during installation.

The SFP file name is:

- ats6943e_front_panel_32.exe 32 Bit version
- ats6943e_front_panel_64.exe 64 Bit version

The SFP can be started from a Windows Explorer window or from the VXIPNP program group.

System Panel

When started, the SFP searches for all the installed DTI modules in the system and queries each module for installed ETB links. If one or more DTI modules are found, the Test System panel is displayed, (see Figure 4-1).

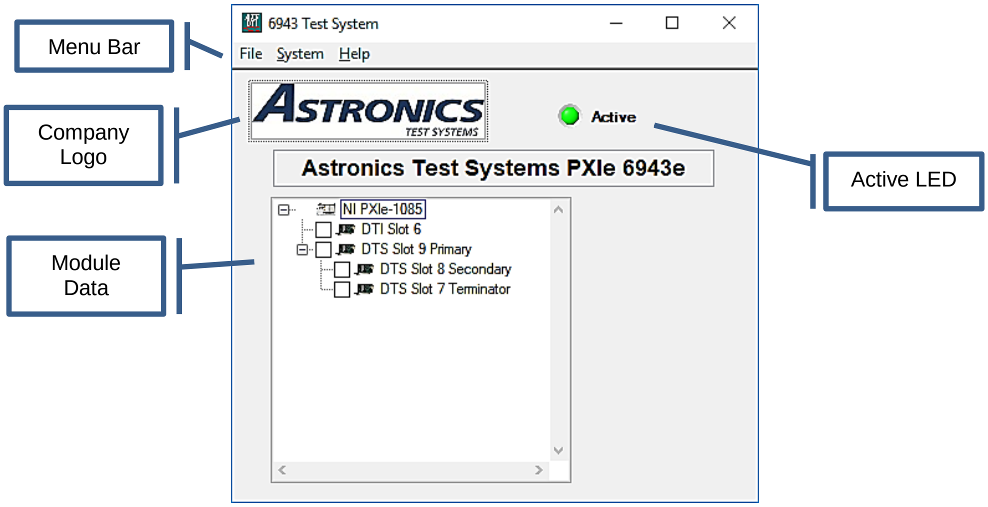


Figure 4-1 Test System Panel

Relevant vi(s):

Initialize

Auto Connect To DTS

System Panel Menu Bar

The system panel menu bar provides File, System and Help utilities.

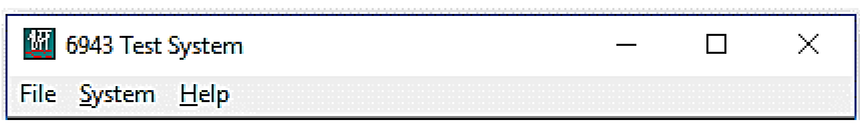


Figure 4-2 System Panel Menu Bar

File Menu

The File Menu is used to exit/close the SFP.

Figure 4-3 System Panel File Menu

File>Exit

Closes the DTI session(s) and exits the SFP.

When the **File>Exit** or close button is selected a prompt panel will display to confirm the close action.

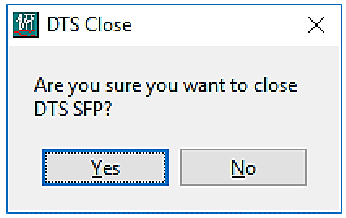


Figure 4-4 System Panel Close Prompt

If “Yes” is selected, a second prompt panel will display asking if the DTI modules should be reset. Selecting “Yes” will reset all DTIs before closing the SFP.

Figure 4-5 System Panel Reset Prompt

Relevant vi(s):

Reset

Close

System Menu

The System Menu is used to edit the DTI settings, configure and calibrate a DTS.

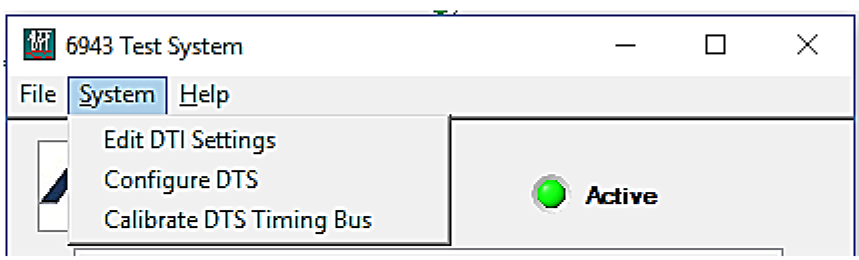


Figure 4-6 System Panel System Menu

System>Edit DTI Settings

Opens a DTI Main Panel for the selected DTI module in the Module Data control.

System>Configure DTS

This menu command performs the following:

1. Verifies the checked DTIs form a valid DTS. A valid DTS must include the “Primary” module and one or more “Secondary” or “Terminating” modules that are under the primary branch.
2. Programs the module interconnect for each DTI.
3. Tests the local and ETB signals of each DTI.

To select a module for the DTS chain click in the check box next to the DTI module.

The results of the DTS configuration will be displayed in a pop up panel.

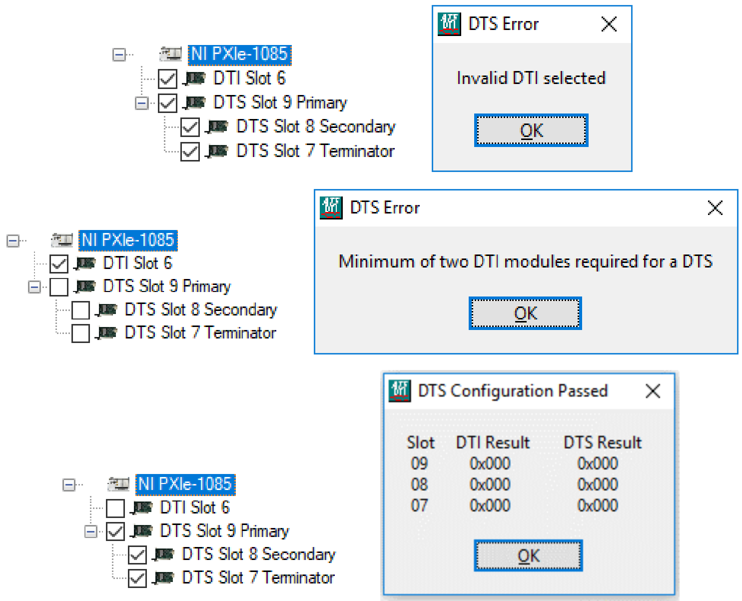


Figure 4-7 Configure DTS Error and Passed Pop Ups

The DTI Result column in the pop up displays the signal results for the local timing bus and the DTS Result column displays the signal results for the ETB timing bus. For both columns a non-zero in any bit position indicates an error of the following signals:

Bit Number	Signal
0	Phase 1
1	Phase 2
2	Phase 3
3	Phase 4
4	Window 1
5	Window 2
6	Window 3
7	Window 4
8	SEQ_CLK
9	SEQ_CLK_D
10	T0_CLK
11	Jump

Table 4-1 Configure DTS Error Bits

Relevant vi(s):

Set Module Interconnect

Test DTS Timing Bus

System>Calibrate DTS Timing Bus

This command calibrates the external timing bus signals of the DTS chain that were previously configured via the "System>Configure" DTS command.

DTS calibration results are displayed in a pop up panel.

The pop up panel messages are listed in Table 4-2.

Message	Description
DTS Calibration Passed	ETB calibration complete, no errors.
Invalid DTS configuration; Primary module slot n	The module in slot n is not configured as a Primary.
Invalid DTS configuration; Secondary module slot n	The module in slot n is not configured as a Secondary.
Invalid DTS configuration; Terminator module slot n	The module in slot n is not configured as a Terminator.
DRS calibration assert out of range; Step 1 terminating module	Unable to position Phase 1 of the terminating module with CLK10 using coarse increment of assert signal.
DRS calibration assert out of range; Step 2 terminating module	Unable to position Phase 1 of the terminating module with CLK10 using fine increment of assert signal.
DRS calibration assert out of range; Step 3 terminating module	Unable to position Phase 1 of the terminating module with CLK10 using decrement of assert signal.
DTS calibration delay out of range; Step 3 terminating module	Unable to position Phase 1 of the terminating module with CLK10 using the assert signal.
Delay value too low error	Unable to align Phase 1 of the terminating module with CLK10 using hardware delay.
Delay value too high error	Unable to align Phase 1 of the terminating module with CLK10 using hardware delay.
Delay value too [low or high] error; DTI Slot n:PH1 alignment	Unable to align Phase 1 of the DTI in slot n to CLK10 using hardware delay.
Delay value too [low or high] error; DTI Slot n:PH1 alignment after coarse delay increment.	Unable to align Phase 1 of the DTI in slot n to CLK10 using hardware delay after coarse delay increment.
Delay value too [low or high] error; DTI Slot n:Signal m.	Unable to align signal m of the DTI in slot n to Phase 1 using hardware delay.
Counter/Timer measurement not ready; Error pulse slot n	Error pulse missing in the DTI in slot n.
Error pulse delay calibration failure; Error pulse delay slot n	Error pulse delay offset out of range in the DTI in slot n.

Table 4-2 Calibrate DTS Pop Up Panel Messages

Signal m in Table 4-2 refers to the following:

- Phase 2, m = 1

- Phase 3, m = 2
- Phase 4, m = 3
- Window 1, m = 4
- Window 2, m = 5
- Window 3, m = 6
- Window 4, m = 7
- SEQ_CLK, m = 8
- T0CLK, m = 9
- SEQ_CLK_D, m = 10

NOTE

DTS calibration results will be saved in the configuration file. Loading a configuration will replace the calibration results.

Help Menu

The Help Menu is used to open the instrument driver help contents and display the SFP programming information panel.

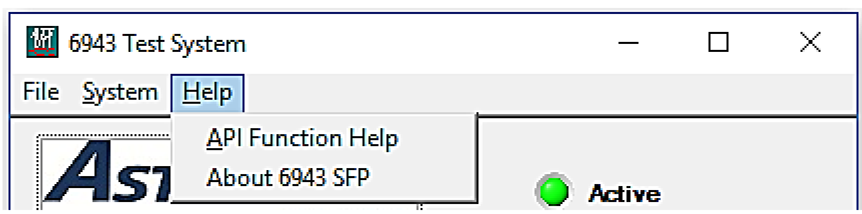


Figure 4-8 System Panel Help Menu

Help>API Function Help

Displays the C API help file table of contents.

Help>About 6943 SFP

Displays revision data for the Soft Front Panel executable.

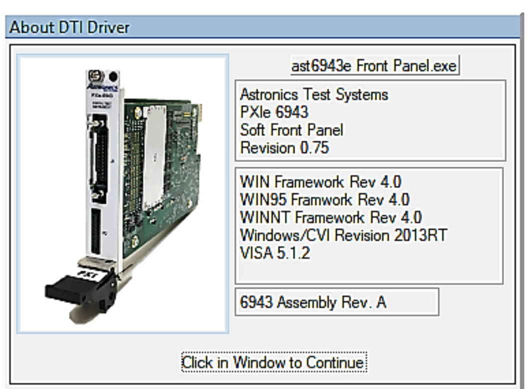


Figure 4-9 Soft Front Panel About Panel

Company Logo

Pressing this control displays the information panel.

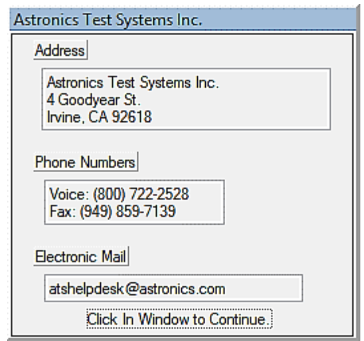


Figure 4-10 Company Information Panel

Active LED

The Active LED indicates whether a session has been established successfully with the DTI module(s).

Module Data

The module data control indicates the module hierarchy and DTS setting. If a module has a ETB link installed in the P2 connector, then the module is listed as a DTS and its inter-module setting is listed (primary, secondary, terminator). If a module does not have an ETB link installed, then the module is listed as a DTI.

Double-clicking on a module will open the DTI main panel for that module.

DTI Main Panel

The DTI main panel is displayed by selecting the “System>Edit DTI” or double-clicking on a DTI from the Chassis Data control.

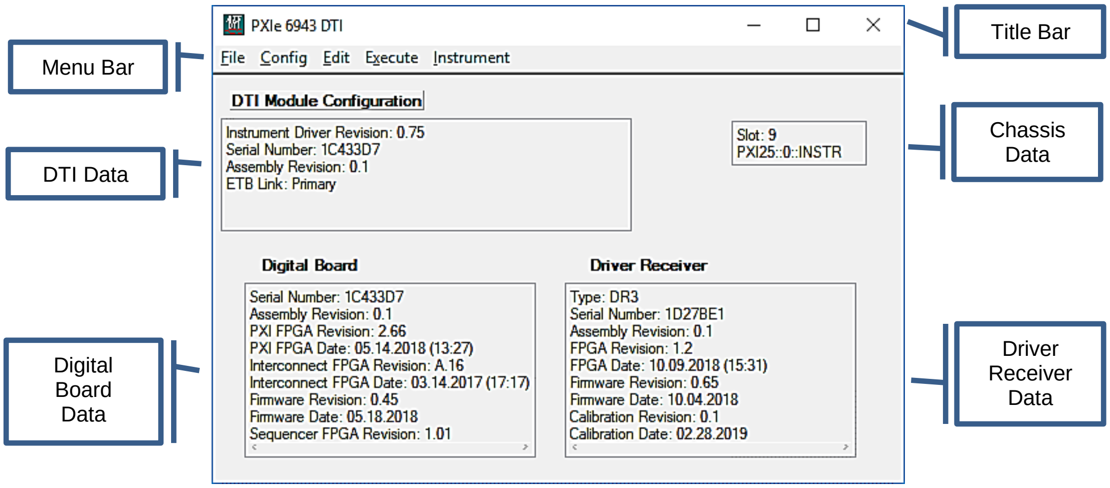


Figure 4-11 DTI Main Panel

This panel displays information that includes DTI data, digital board data, driver receiver data and chassis data and calibration data. The menu bar provides access to program or query the module settings. The title bar displays the configuration file if one was loaded. Panels opened from this panel (child panels) display the slot number of the DTI to avoid confusion since multiple DTI Main Panels can be opened at the same time.

Relevant vi(s):

- **Query Configuration**
- **Revision Query**
- **Query Digital Board**
- **Query Digital Resource Module**
- **Query Module Data**
- **Query Driver Receiver Board**
- **Query Module Firmware**
- **Query Module FPGA**

DTI Main Panel Menu Bar

The DTI main panel menu bar provides access to select, program and save the DTI hardware. Relevant VI's are included with the menu options.

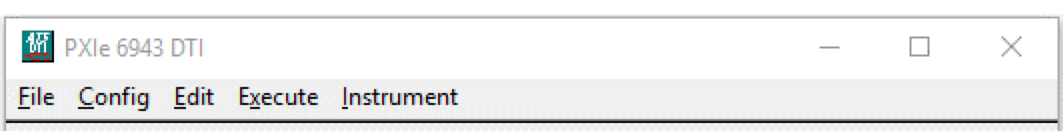


Figure 4-12 DTI Main Panel Menu Bar

File Menu

The File Menu manages the loading and saving of test files. With this menu, DTI SFP project files are created, loaded, saved and renamed. There are also diagnostic loads, register dumps and calibration data loads. A file history list permits quick reloading of recently accessed test files. The DTI main panel and all its children can also be closed from this menu.

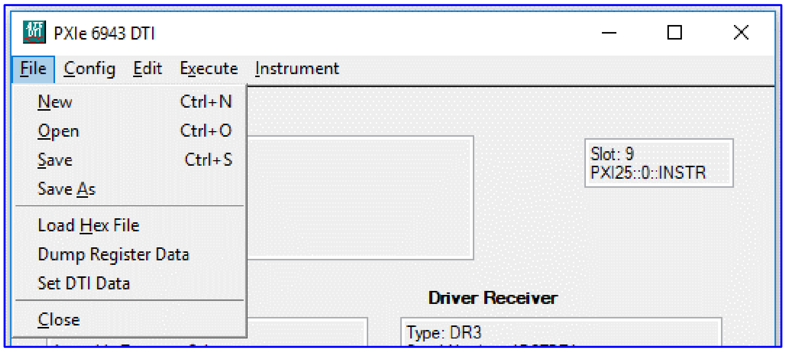


Figure 4-13 DTI Panel File Menu

File>New

Clears the DTI hardware to power up reset settings.

Relevant vi(s):

Reset

File>Open

Opens a file browser for choosing a configuration file.

Relevant vi(s):

Load Configuration

File>Save

Updates the configuration file with the latest editing changes.

Relevant vi(s):

Save Configuration

File>Save As

Creates a new configuration file with the latest editing changes. It then becomes the current configuration.

Relevant vi(s):

Save Configuration

File>Load Hex File

Low level utility routine for hardware checkout.

File>Dump Register Data

Low level utility routine for hardware checkout and test development diagnostics.

File>Set DTI Data

Opens up a panel that allows the operator to set serial number and assembly revision data into the non-volatile memory on the digital and driver receiver boards.

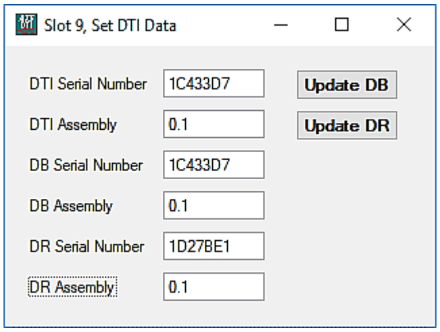


Figure 4-14 Set DTI Data Panel

The "**Update DB**" button saves the DTI and DB serial number and assembly revision data to non-volatile memory on the digital board.

The "**Update DR**" button saves the DR serial number and assembly revision data to non-volatile memory on the driver receiver board.

File>Close

Closes the DTI Main panel and any child panels associated with the specific DTI.

Config Menu

The Configuration (Config) Menu queries and configures the DTI hardware settings.

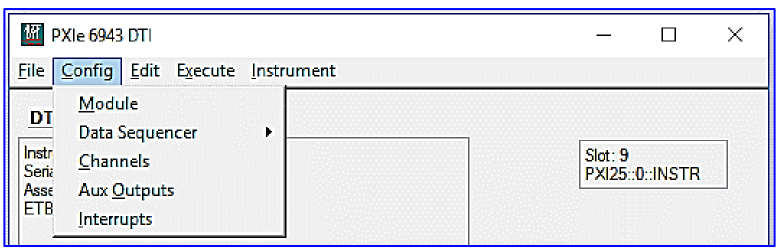


Figure 4-15 DTI Panel Config Menu

Config>Module

Config Module panel sets and queries module parameters.

Config>Data Sequencer

Config Data Sequencer sets and queries the following sequencer settings;

- Clocks
- Timers
- Triggers
- Pulse Generator
- Generic Settings

Config>Channels

Config Channels panel sets and queries the 32 I/O channel settings.

Config>Aux Outputs

Config Aux Outputs sets and queries the 12 AUX output settings.

Config>Interrupts

Config Interrupts sets and queries the interrupt registers.

Edit Menu

The Edit Menu creates, programs and queries data sequencer structures and settings.

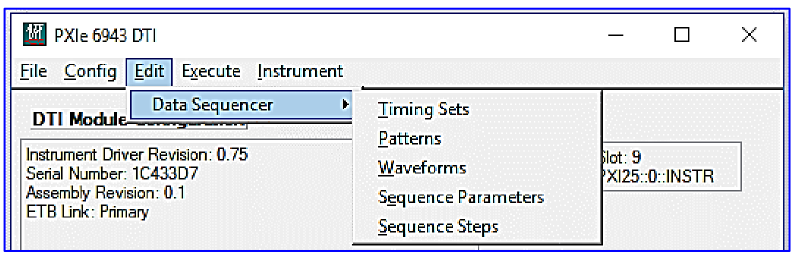


Figure 4-16 DTI Panel Edit Menu

Edit>Data Sequencer

Edit Data Sequencer is sets and queriesthe following sequencer settings:

- Timing Sets
- Patterns
- Waveforms
- Sequence Parameters
- Sequence Steps

Execute Menu

The Execute Menu programs the run options, runs the sequences, sets and executes timer/counter logic, sets PMU operation and views the execution results.

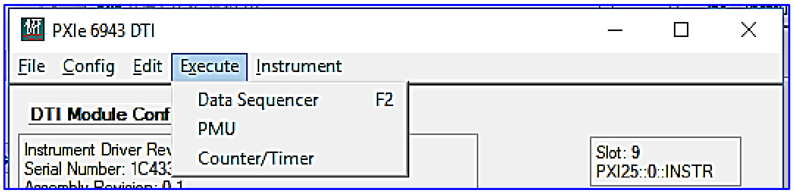


Figure 4-17 DTI Panel Execute Menu

Execute>Data Sequencer

Execute Data Sequencer sets and queries the run options, enable the driver/receiver power and view the results and events.

Execute>PMU

Execute PMU sets and queries the PMU options, and query the results.

Execute>Counter/Timer

Execute Counter/Timer sets and queries the counter options, triggers and queries the results.

Instrument Menu

The Instrument Menu runs self-test, calibration, updates firmware and monitors routines on the DTI hardware.

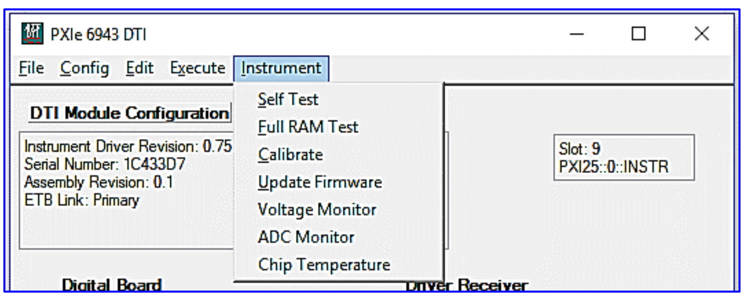


Figure 4-18 DTI Panel Instrument Menu

Instrument>Self Test

This executes the standard self-test and displays the results.

Relevant vi(s):

Self-Test

Instrument>Full RAM Test

This executes a memory test on all the read/write memory on the module and displays the results.

Relevant vi(s):

Ram Test

Instrument>Calibrate

Calibrates and stores the driver/receiver I/O hardware.

Instrument>Update Firmware

Updates the FPGA and µP firmware on the module.

Instrument>Voltage Monitor

Displays the voltage monitor data from the digital and the driver/receiver boards.

Instrument>ADC Monitor

Displays the ADC monitor data from the driver/receiver board.

Instrument>Chip Temperature

Displays the chip temperature data from the 32 I/O channels on the driver/receiver board.

Clicking the company logo in the top left corner of the main panel displays the information panel.

Programming Steps

The following list details the steps used to implement a test program:

1. Open a communication link to the DTI module(s) called a session.
2. Configure hardware settings.
3. Configure the I/O and AUX channels.
4. Edit the Data Sequencer.
    a. Program the timing sets to govern the I/O data transfers.
    b. Create the pattern sets and populate them as appropriate.
    c. Create the Waveforms and define
    d. Set sequence parameters
    e. Edit sequence steps
5. Execute the sequence.
6. PMU Operation.
7. Counter/Timer Operation.
8. Utilize status and post process functions to evaluate/analyze results.
9. Close the session(s).

The following sections describe the SFP operation as it pertains to the steps listed above. Additionally, sections are included covering the instrument functions, self-test, calibration, and utility functions.

The relevant instrument driver function(s) for each step are listed.

Opening DTI Session(s)

Starting the SFP initiates a search for all DTIs using the VISA library. Once all the DTIs have been identified, the system panel displays and the DTIs are inserted and ordered in to the **Module Data** control.

Relevant vi(s):

Initialize

Auto Connect To DTS

Configuring Hardware Settings

Configuring the hardware settings is done from two panels: Configure Module and Configure Data Sequencer.

Configure Module Panel

Access this panel from the menu bar: **Config>Module**.

The Configure Module panel is used to program the inter-module mode, signal delays, PXI TRG routing, driver/receiver properties, and record settings.

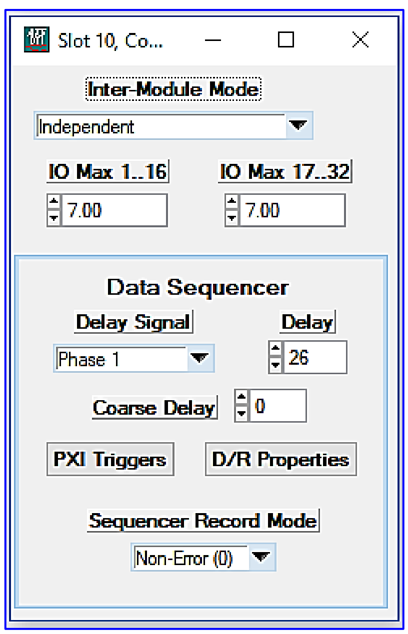


Figure 4-19 Configure Module Panel

The following sections describe the Configure Module panel controls.

Inter-Module Mode

This pull-down control programs the timing bus source for the sequencer. The default setting is the ETB chain are set via the ETB links on the front panel. If a link is not installed, then DTI can only be configured as Independent.

The table below lists the selections for the Inter-Module Mode pull-down control.

Inter Module Mode	Description
Independent	This setting causes the DTI to use local timing bus signals. This is the default setting. All DTI modules can operate as independent.
Primary	This setting is only valid on DTI modules that have a primary ETB link installed. The primary module is located in the right-most slot position in the PXI chassis relative to the DTI modules that are part of the ETB chain. This modules sequencer provides all the timing for the sequencers that are part of the ETB chain.
Secondary	This setting is only valid on DTI modules that have a secondary ETB link installed on the front panel. Secondary module(s) located between the Primary and Terminator modules.
Terminator	This setting is only valid on DTI modules that have a terminator ETB link installed. The Terminator module is located in the DRM leftmost slot of the ETB chain.
Test Mode 1	This setting is used for factory test.

Table 4-3 Inter-Module Mode Settings

Relevant vi(s):

Set Module Interconnect

IO Max 1..16

This numeric entry specifies the maximum drive/compare level that can be programmed for the lower 16 I/O channels. This level establishes the HV-VCC1 voltage level.

The valid range is 0 V to +7 V.

Relevant vi(s):

Set IO Max

IO Max 17..32

This numeric entry specifies the maximum drive/compare level that can be programmed for the upper 16 I/O channels. This level establishes the HV-VCC2 voltage level.

The valid range is 0 V to +7 V.

Relevant vi(s):

Set IO Max

Delay Signal

The DTI uses the front panel external timing bus to function in a multi-module mode. During the timing bus alignment process, these signals need to be delayed in order to align the timing between modules. This control is used to select the signal to query the delay time.

Setting	Description
Phase1-4	Phase timing signals
Window 1-4	Window timing signals
SEQ_CLK	Sequence Clock
SEQ_CLK_D	Delayed Sequence Clock
T0_CLK	Pattern Clock
Jump	Jump signal

Table 4-4 Delay Signal Settings

Delay

This control displays the delay value for the signal specified by the Delay Signal control. The valid delay range is from 0 to 63 and the delay is 0.15 ns/step.

Coarse Delay

This control displays the coarse delay value for the module. The valid delay range is from 0 to 3 and the delay is ~2 ns/step.

PXI Triggers

This command button displays the “Set PXI Triggers” panel so the signals can be programmed. The panel contains a pull-down control and Invert button for each trigger signal.

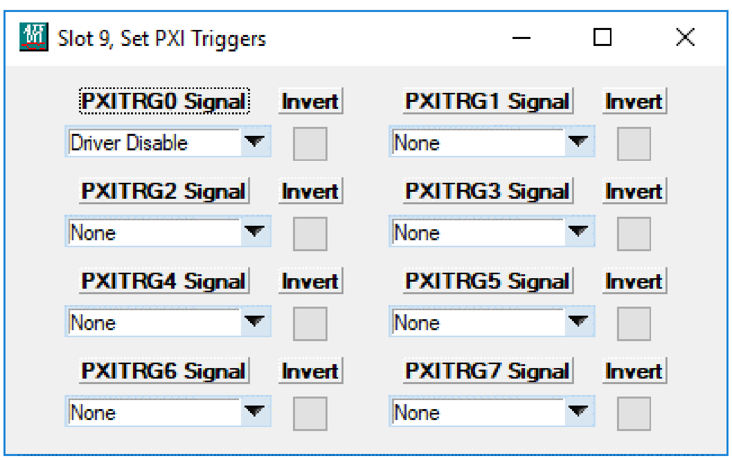


Figure 4-20 Set PXI Triggers Panel

The trigger lines are open collector on the PXI backplane. The chassis provides a split termination which provides a weak pull-up (thus there is a slow rising edge recovery time). Programming an active high signal on this panel will actually drive the backplane signal low. This allows multiple DTIs to actively drive the same trigger line and form a wired-OR condition. The DTI module receiving the signal knows to invert the incoming signal to re-create an active high. But non-DTI PXI modules which receive triggers from a DTI or send triggers to the DTI will need to account for this protocol.

Under certain circumstances, it may be desired to form a wired-AND or wired-OR on the backplane such as when doing channel tests. This is discussed further in the **PXI Backplane Trigger Bus** section of Chapter 6. There is substantially more information in this section regarding the use of the trigger bus.

PXITRGn Signal

This pull-down control programs the signal source for the specified PXI trigger.

Setting	Description of the PXI Trigger Source Signal
None	Disables the TTLTRG driver
AUX1-AUX12	Selects the specified AUX input signal from the front panel
Halted	Used for DTS halt operation between coupled sequencers
Probe Button	Selects the state of the probe button
Pulse Generator	Selects the pulse generator signal
Sequence Flag 1-2	Selects the specified sequence flag
Sync 1-2	Selects the specified sync signal
CHT1-4	Selects the specified channel test signal
Idle Active	Idle active flag
Sequence Active	Sequence active flag
Sequence Reset	DTS sequence reset command
DTS Sync	DTS Sync signal
Driver Disable	DTS driver disable command
Master Reset	DTS master reset

Table 4-5 PXITRG Signal Settings

All DTI modules in the ETB chain must select the same trigger for the last four listed signals, if used. These signals are used for DTS signaling.

Relevant vi(s):

Set TTL Triggers

Invert

This Invert button inverts the associated signal before it is driven onto the selected backplane trigger line.

Relevant vi(s):

Set TTL Triggers

D/R Properties

This command button displays the “**Configure D/R Properties**” panel so the configuration settings can be programmed for the Driver/Receiver board.

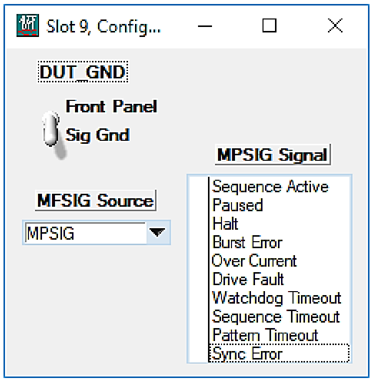


Figure 4-21 Configure D/R Properties Panel

DUT_GND

This control programs a relay that will connect the DUT_GND reference for the Pin Electronics to either a front panel DUT_GND or to signal ground. The former corrects ground reference offsets due to cabling.

Relevant vi(s):

Set Power Settings

MFSIG Source

This pull-down control programs the MFSIG signal function.

Setting	Description
Disabled	Signal is not driven
MPSIG	Signal is assigned to the sequencer MPSIG signal

Table 4-6 MFSIG Settings

Relevant vi(s):

Set Power Settings

MPSIG Signal

This control sets the source of the MPSIG. All checked signals are ORed together.

Setting	Description
Sequence Active	MPSIG goes high when sequence active is true.
Paused	MPSIG goes high when the sequencer is paused.
Halt	MPSIG goes high when the sequencer is halted.
Burst Error	MPSIG goes high when burst error is true.
Over Current	MPSIG goes high when over current is true.
Drive Fault	MPSIG goes high when drive fault is true.
Watchdog Timeout	MPSIG goes high when the watchdog timeout is true.
Sequence Timeout	MPSIG goes high when the sequence timeout is true.
Pattern Timeout	MPSIG goes high when the pattern timeout is true.

Setting	Description
Sync Error	MPSIG goes high when the sync error is true.

Table 4-7 MPSIG Source

Relevant vi(s):

Set MPSIG Source

Sequencer Record Mode

This pull-down control programs the sequencer record mode.

The sequencer record mode selects what is stored in the record memory when the sequence **Step Record Mode** is set to Record Count (see the section on **Step Record Mode** later in this chapter).

NOTE

If Step Record Mode is set to either “Record Error” or “Record Response”, then the Sequence Record Mode setting will be ignored.

Setting	Description	Typical Usage
Disabled	The contents of the record memory will not change during the next burst if Step Record Mode is set to either None or Record Count.	Setting to Disabled insures that the record memory will not be written to when Step Record Mode is set to either None or Record Count. This means that if errors were recorded in a previous burst, they will remain in memory throughout the current burst.
Non-Error(0)	The contents of the record memory will be set to 0 during the next burst if Step Record Mode is set to either None or Record Count.	Setting to Non-Error (0) when Step Record Mode is set to either None or Record Count clears the record memory during the next burst, ensuring that any previously recorded errors will not persist.

Table 4-8 Sequencer Record Mode Settings

Relevant vi(s):

Set Sequence Record Mode

Config Data Sequencer

The Configure Data Sequencer panel programs the clock settings, sequence control signals, timeout values, overcurrent, and record settings.

Access this panel from the menu bar: **Config>Data Sequencer**.

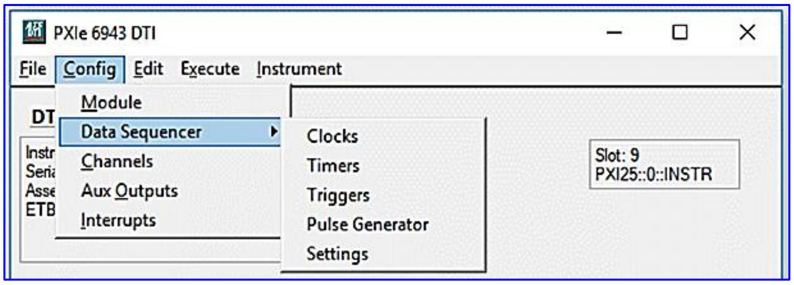


Figure 4-22 Configure Data Sequencer

Configure Clocks

Access this panel from the menu bar: **Config>Data Sequencer>Clocks**.

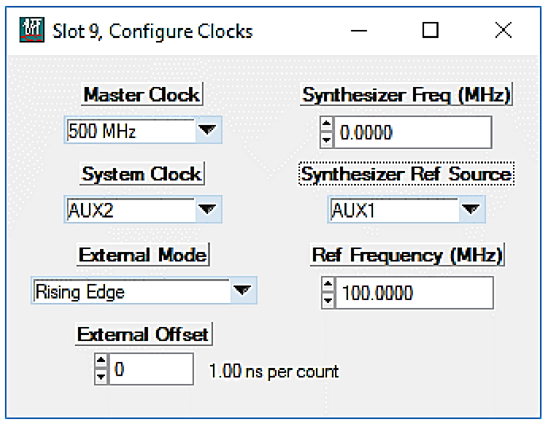


Figure 4-23 Configure Clocks

Master Clock

This pull-down control programs the sequencer master clock source.

The master clock defines the sequencer timing resolution. The resolution is half of the master clock period.

Setting	Description	Typical Usage
500 MHz	Sequencer timing resolution set to 1ns	Default case; 1 ns timing resolution is required; no frequency reference
Frequency Synthesizer	Sequencer timing resolution set to 1 / (2 * FS) For example; if FS = 100 MHz Resolution = 1 / (2 * 100,000,000) Resolution = 5ns	An external frequency reference will be used to train the master clock, or when a non-standard, exact data rate is required. For example, if a 48 MHz data rate is required, the synthesizer set to 480 MHz gives 1.04167 ns per count timing. 20 counts gives a 20.8333 ns period or 48 MHz. Using the 500 MHz clock with 21 counts yields a data rate of 47.619 MHz, the closest pattern rate achievable using the 500 MHz clock.

Table 4-9 Master Clock Source Settings

Relevant vi(s):

Set Master Clock Source

System Clock

This pull-down control programs the sequencer System Clock source.

The System Clock signal defines the pattern period.

Setting	Description	Typical Usage
Internal T0CLK	System Clock source set to the internal period defined by the sequencer step.	DTI or DTS where internal master clock timing is acceptable.
AUX1-AUX12	System Clock source set to the external front panel signal.	Auxiliary line is assigned the function of external clock where: AUX1-8: LVTTL source 1kHz-50 MHz AUX9-12: differential LVDS source from 1 kHz to 50 MHz
Pulse Generator	System Clock source set to the internal pulse generator signal.	For when pulse width control of the system clock is required.
Frequency Synthesizer	System Clock source set the to the internal frequency synthesizer signal.	For generating a pattern period not tied to the MCLK or synchronizing the pattern output to an external clock by using it as the FS reference clock.

Table 4-10 System Clock Source Settings

Relevant vi(s):

Set System Clock Source

External Mode

This pull-down control selects the clock edge mode when the System Clock source is set to any non T0CLK selection.

Setting	Description
Rising Edge	Use the rising edge of the external signal as the active edge
Falling Edge	Use the falling edge of the external signal as the active edge
Both Edges	Use the rising and falling edge of the external signal as the active edge
Divide by 2 Rising Edge	Divide the external signal by two and use the rising edge as the active edge
Divide by 2 Falling Edge	Divide the external signal by two and use the falling edge as the active edge

Table 4-11 External Mode Settings

Relevant vi(s):

Set System Clock Parameters

External Offset

This control specifies the external System Clock offset in order to align the clock/data relationship. The valid offset range is from 0 to 65534 (even numbers only) and the resolution is 1/2 the MCLK period. For example if the MCLK is set to 100 MHz then the resolution is 5 ns (1/2 of 10 ns).

Relevant vi(s):

Set System Clock Parameters

Synthesizer Freq (MHz)

This input control specifies the Frequency Synthesizer setting. The valid frequency range is from 40 kHz to 500 MHz. Setting the control to **0** turns off the frequency synthesizer.

Relevant vi(s):

Set Frequency Synthesizer

Synthesizer Ref Source

This pull-down control programs the frequency synthesizer reference source.

Setting	Description
Internal	Reference source set to PXIe backplane CLK100
AUX1-AUX12	Reference source set to front panel signal
CLK10	Reference source set to PXIe backplane CLK10
CLK50	Reference source set to PXIe backplane CLK100 / 2.

Table 4-12 Synthesizer Ref Source Settings

Relevant vi(s):

Set Frequency Synthesizer

Reference Freq (MHz)

This input control specifies the external reference frequency and only appears when an external synthesizer reference source is selected. In these cases, the frequency synthesizer needs to be scaled so that it can produce the desired output frequency given the nominal external reference frequency. The valid external reference frequency range is from 5 MHz to 100 MHz.

Relevant vi(s):

Set Frequency Synthesizer

Configure Timers

Access this panel from the menu bar: **Config>Data Sequencer>Timers**. There are five timers:

- Watchdog
- Sequence Timeout
- Pattern Timeout
- Pattern Delay 1
- Pattern Delay 2

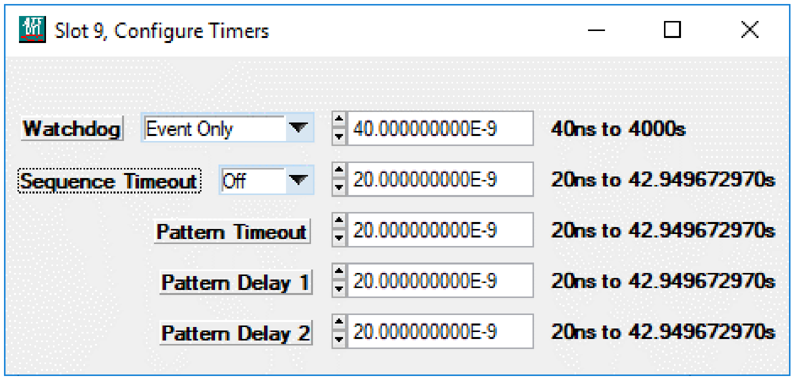


Figure 4-24 Configure Timers

Watchdog

The watchdog timer is used to indicate that the sequence execution did not finish within the specified time period:

- The Watchdog Timeout Timer starts when SEQACT begins. This timer does not stop during a Pause or Halt (including single stepping).
- Generates an event (WDTO) if the sequence active time exceeds the specified value.
- If the watchdog action is set to Disable Drivers, all 32 drivers will tri-state when a timeout occurs (but any active load or resistive loading remains).

Sequence Timeout

The sequence timeout timer can be used in a Sequence Step that has a conditional loop where one is waiting for a termination condition to proceed to the next Sequence Step:

- It starts when the first branch takes place.
- The timer is reset at the beginning of every step unless the sequence timeout continue flag is set in the Edit Sequence Step panel.
- Cannot be nested.
- Does not stop during a Pause or Halt (including single-stepping).
- A timeout will generate an event and the occurrence of this particular event can be enabled to generate an interrupt so the S/W can query the events to see which one occurred.
- The continuous conditional loop will continue to branch unless the termination condition is subsequently met, whereby execution will advance to the next Sequence Step as usual. If it doesn't, the user can manually halt or stop the Sequence.
- The sequence timeout can be used to generate an event to indicate that a sequence step (or steps) has taken too long to complete.

Pattern Timeout

The pattern timeout timer can be used in a sequence step that has a Pause:

- In the Sequence Step, the **Handshake Modifier** can be set to the Pattern Timeout.
- The Timer starts when the Pause begins.
- The Pattern Timeout Timer will generate an event when the timer times out. The Pause will continue unless the termination condition is subsequently met, whereby execution will resume. If it doesn't, the user can manually resume or stop the Sequence.

Pattern Delay

The two pattern delay timers are used in a sequence step that has a Pause:

- In the Sequence Step, the **Handshake Modifier** can be set to Pattern Delay 1 or 2.
- The Timer starts when the Pause begins.
- A Pattern Delay Timer timeout will cause a resume to be generated.

Watchdog Action

This toggle control enables/disables the watchdog timeout Event Only/Driver Disable feature.

Setting	Description
Event Only	Set bit in event register only when a watchdog timeout occurs.
Disable Drivers	Set bit in event register and disable the drivers when a watchdog timeout occurs.

Table 4-13 Watchdog Action

Relevant vi(s):

Set Watchdog Timer

Watchdog Time

This numeric control specifies the watchdog timeout count.

The timeout is programmed with a range of 40ns to 4000s.

The watchdog timer set resolution adjusts based on the timeout value:

Timer Setting	Resolution
Less than 10 ms	20 ns
From 10ms to < 10 s	100 ns
From 10 s to 4000 s	1 us

Table 4-14 Watchdog Timer Resolution Ranges

Relevant vi(s):

Set Watchdog Timer

Sequence Timeout State

This toggle control is used to enable/disable the sequence timeout feature.

Setting	Description
Off	Disable sequence timeout bit in event register.
On	Enable sequence timeout bit in event register.

Table 4-15 Sequence Timeout State Action

Relevant vi(s):

Set Sequence Timer

Sequence Timeout Time

This numeric control specifies the sequence timeout count.

The timeout is programmed in 10 ns steps with a range of 20ns to 42.94967297s.

Relevant vi(s):

Set Sequence Timer

Pattern Timeout

This numeric control specifies the pattern timeout count.

The timeout is programmed in 10ns steps with a range of 20ns to 42.94967297s.

Relevant vi(s):

Set Pattern Timer

Pattern Delay 1-2

This numeric control specifies the pattern delay.

The pattern delay is programmed in 10ns steps with a range of 20ns to 42.94967297s.

Relevant vi(s):

Set Pattern Delay Timer

Configure Triggers

Access this panel from the menu bar: **Config>Data Sequencer>Triggers**.

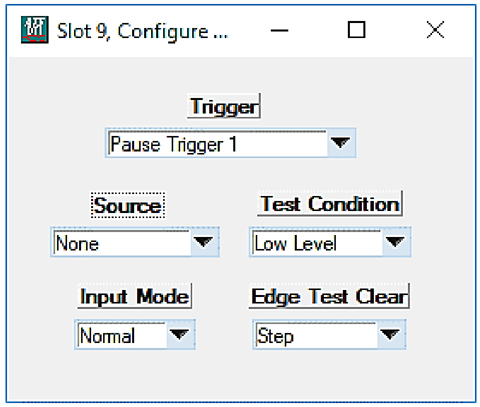


Figure 4-25 Configure Triggers Panel

Pause Trigger and Pause Resume Trigger

The pause triggers are used to stop the pattern timing during a burst. The corresponding resume trigger re-starts the pattern timing from where it was stopped.

A pause/resume can be based on the true/false state of any of the two pause triggers. For example; if Pause 1 Trigger was set to AUX1 'Low Level' and Pause 1 Resume was set to AUX1 'High Level', then the timing would stop when AUX1 is low and continue when AUX1 goes high.

Phase Resume Triggers

If the pattern timing is paused by either the assert or return edge of a phase, then this trigger is used to resume the timing.

Halt Trigger

The halt trigger causes the sequencer to halt based on the current halt mode.

Execute Start Trigger

The execute start trigger causes the selected sequence step to start. Selecting a sequence step consists of arming the sequence step. In DTS configuration, all of the coupled sequencers need to be armed first using the **Arm Sequence** vi.

Execute Stop Trigger

The execute stop trigger causes the sequencer to stop based on the current stop mode.

Jump Trigger

Four sequence jump triggers are available. The sequence jump triggers are used for conditional jumping/looping. A jump/loop can be based on the true/false state of any of the four sequence jump triggers. For example; if jump trigger 1 test mode is set to 'Low Level', then a jump if trigger 1 true would occur if the selected jump trigger 1 source is low.

Trigger

This pull-down control selects the trigger to program.

Setting	Description
Pause Trigger 1 - 2	Select Pause Trigger 1 or 2 to edit
Pause Trigger 1 - 2 Resume	Select Pause Trigger 1 or 2 Resume to edit
Phase 1 - 4 Resume	Select Phase 1 through 4 Resume to edit
Execute Start	Select Execute Start to edit
Execute Stop	Select Execute Stop to edit
Halt	Select Halt to edit
Jump 1 – 4	Select Jump 1 through 4 to edit

Table 4-16 Trigger Settings

Source

This pull-down control programs the trigger source.

Setting	Description
None	No trigger source selected
AUX1-AUX12	Trigger source set to front panel signal
CHT1	Trigger source set to channel test 1
TTLTRG0-7	Trigger source set to PXI TTL trigger

Table 4-17 Trigger Source Settings

Relevant vi(s):

Set Handshake Pause Trigger

Set Handshake Resume Trigger

Set Phase Resume Trigger

Set Jump Trigger

Set Halt Trigger

Set Execute Start Trigger

Set Execute Stop Trigger

Arm Idle Sequence

Arm Sequence

Test Condition

This pull-down control programs the trigger test condition.

Setting	Description
Low Level	Test for a low level
High Level	Test for a high level
Rising Edge	Test for a rising edge

Setting	Description
Falling Edge	Test for a falling edge

Table 4-18 Trigger Test Condition Settings

Relevant vi(s):

Set Handshake Pause Trigger

Set Handshake Resume Trigger

Set Phase Resume Trigger

Set Jump Trigger

Set Halt Trigger

Set Execute Start Trigger

Set Execute Stop Trigger

Input Mode

This pull-down control programs the trigger input mode.

Setting	Description
Normal	Do not modify input signal before testing.
Inverted	Invert input signal before testing.

Table 4-19 Trigger Input Mode Settings

Relevant vi(s):

Set Handshake Pause Trigger

Set Handshake Resume Trigger

Set Phase Resume Trigger

Set Jump Trigger

Set Halt Trigger

Set Execute Start Trigger

Set Execute Stop Trigger

Edge Test Clear

This pull-down control programs the trigger event clear.

The event clear allows the user to program when the rising/falling edge flip-flops are cleared during operation for the following triggers:

- Pause 1-2
- Halt
- Jump 1-4

Setting	Description
Start	Clear flip-flops at start of burst
Step	Clear flip-flops at start of ethinhvery sequence step
Event True	Clear flip-flops when trigger event tests true

Table 4-20 Trigger Event Clear Settings

Relevant vi(s):

Set Pause Trigger Reset

Set Halt Trigger Reset

Set Jump Trigger Reset

Configure Pulse Generator

Access this panel from the menu bar: **Config>Data Sequencer> Pulse Generator**.

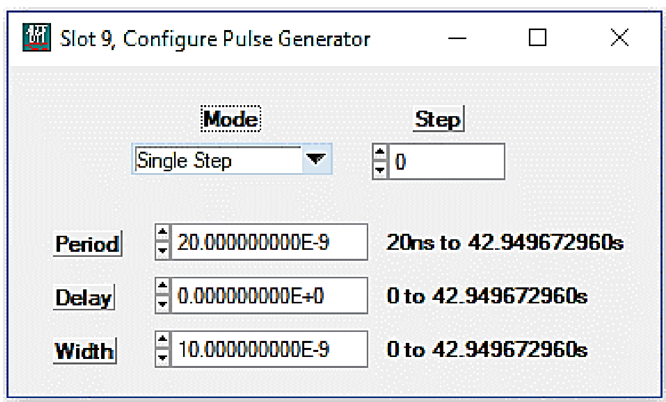


Figure 4-26 Configure Pulse Generator

Each data sequencer has a programmable pulse generator that can be routed to the following signals:

- Data sequencer System Clock
- PXI TTLTRG
- Front panel AUX

Mode

This pull-down control programs the pulse generator mode.

Setting	Description
Continuous	The pulse generator begins continuous output when armed
Continuous Start	The pulse generator begins continuous output from the start of the sequence when armed

Setting	Description
Single Start	The pulse generator outputs a single pulse from the start of the sequence when armed
Single Step	The pulse generator outputs a single pulse from the start of the specified step when armed. Note: if looping the sequence step or bursting the entire sequence, the pulse generator will re-trigger.

Table 4-21 Pulse Generator Mode Settings

Relevant vi(s):

Set Pulse Parameters

Step

This input control specifies the step number when the Mode is set to Single Step.

The Step is programmed with a range of 0 to 4095.

Relevant vi(s):

Set Pulse Parameters

Period

This input control specifies the pulse generator period.

The period is programmed in 10ns steps with a range of 20ns to 42.94967297s.

The pulse period is not required for Single Start and Single Step mode.

Relevant vi(s):

Set Pulse Period

Delay

This input control specifies the pulse generator delay from the start of the sequence or sequence step. Delay is not applicable when the Pulse Generator is in Continuous mode.

The delay is programmed in 10ns steps with a range of 20ns to 42.94967297s (with an uncertainty of ±5ns).

Relevant vi(s):

Set Pulse Delay

Width

This control specifies the pulse generator width.

The width is programmed in 10ns steps with a range of 0 to 42.94967295s.

If the width is equal to or greater than the period in Continuous and Continuous Start mode, then the result will be a continuously true pulse.

If the width plus the delay is greater than the period in Continuous and Continuous Start mode, then the pulse width will be reduced proportionately and vanish at some point.

Relevant vi(s):

Set Pulse Width

Configure Data Sequencer Settings

Access this panel from the menu bar: **Config>Data Sequencer>Settings**.

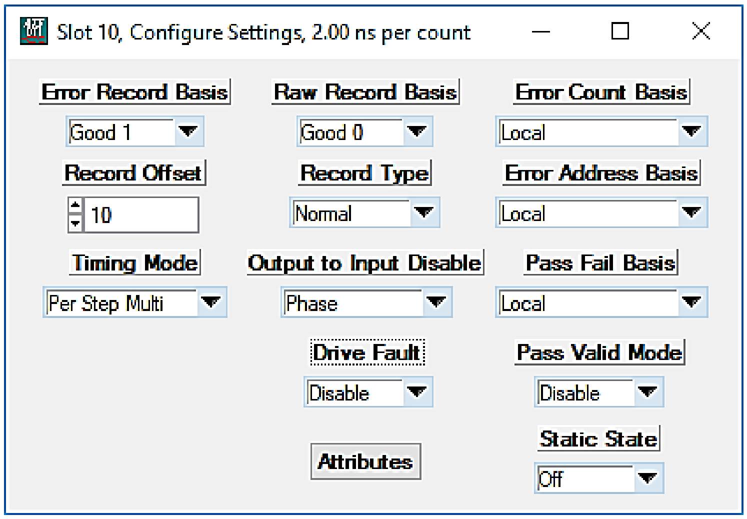


Figure 4-27 Data Sequencer Configure Settings Panel

Error Record Basis

This pull-down control programs the sequencer error record basis.

This control allows the user to select how the response data will be evaluated for errors.

Setting	Description
Dual	Use both good 1 and good 0 comparator levels
Good 1	Use only the good 1 comparator (Single threshold)

Table 4-22 Error Record Basis Settings

Relevant vi(s):

Set Record Parameters

Raw Record Basis

This pull-down control programs the sequencer raw record basis.

This control allows the user to select which comparator will be used to determine the data level.

Setting	Description
Good 0	Use good 0 comparator levels Note: The Good 0 is complemented when recorded

Setting	Description
Good 1	Use good 1 comparator levels

Table 4-23 Raw Record Basis Settings

Relevant vi(s):

Set Record Parameters

Record Offset

The record offset allows the user to shift the record signals (pattern code expect and mask, window strobes) to accommodate system and UUT delay.

The valid offset range is from 0 to 63 MCLKs.

The record offset is reduced as the Master Clock Frequency is reduced. For example, if the record offset was set to 20 with a 500MHz Master Clock, then the record offset should be set to 10 for a 250MHz Master Clock.

Performing end of cable deskew on the DTI or the DTS sets the record offset to the correct value and should not be modified.

When manually setting the Record Offset, the maximum value is limited by the pattern period using the following formula:

ROmax = (period * 4 – 4)/2

Notes:

1. Round the period down if it's odd.
2. The period is the value set as the T0Clk Period regardless of the Master Clock Frequency.

Example:

At a Data Rate of 50MHz (period = 20ns, using a 500MHz Master Clock), the maximum record offset is 38 (76ns). The intrinsic round-trip Driver/Receiver delay is ~20 ns. The delay left for cabling is ~56ns (76ns – 20ns). Using RG178/316 (1.46ns per foot), the max. cable length tolerable is ~38'.

Relevant vi(s):

Set Record Parameters

Record Type

This pull-down control programs the record type.

Settings	Description
Normal	Data stored in the record memory will be at the same offset as the pattern set memory.
Indexed	Data stored in the record memory will begin at offset 0. The record index memory contains the information needed to realign the record memory with the sequence step data.

Table 4-24 Record Type Settings

Relevant vi(s):

Set Record Parameters

Error Count Basis

This pull-down control programs the sequencer error count basis.

This control allows the user to select which error signal to use to determine the error count.

Setting	Description	Typical Usage
Local	Use local error	Error counting is globally enabled
Qualified Local	Use BERREN qualified local error	Error counting is enabled per pattern by the BERREN bit qualifier
DTS	Use DTS error	DTS error counting is globally enabled
Qualified DTS	Use BERREN qualified DTS error	DTS error counting is enabled per pattern by the BERREN bit qualifier

Table 4-25 Error Count Basis Settings

This is discussed in more detail in the **Recording Sequence Results** section in Chapter 6 including data rate limitations.

Relevant vi(s):

Set Error Parameters

Error Address Basis

This pull-down control programs the sequencer error address basis.

This control allows the user to select which error signal causes an error to be recorded in the Error Address Memory.

Setting	Description	Typical Usage
Local	Use local error	Error recording is globally enabled
Qualified Local	Use BERREN qualified local error	Error recording is enabled per pattern by the BERREN bit qualifier
DTS	Use DTS error	DTS error recording is globally enabled
Qualified DTS	Use BERREN qualified DTS error	DTS error recording is enabled per pattern by the BERREN bit qualifier

Table 4-26 Error Address Basis Settings

This is discussed in more detail in the **Recording Sequence Results** section in Chapter 6 including data rate limitations.

Relevant vi(s):

Set Error Parameters

Timing Mode

This pull-down control programs the timing mode, which selects one of three available timing set organization methods.

Setting	Description
Per Step Multi	1024 steps with four phase/window pairs per step.
Per Step Single	4096 steps with one phase/window pair per step.
Indexed	4096 sequence steps with 256 timing sets indexed. Four phase/window signals per timing set.

Table 4-27 Timing Mode Settings

Relevant vi(s):

Set Timing Mode

Output-to-Input Disable

This pull-down control programs the output-to-input disable setting.

When a channel transitions from an output pattern code to an input pattern code, this enable can be set to disable the output at the beginning of the pattern (System Clock) or on a phase assert.

Setting	Description
System Clock	Disable output on System Clock
Phase	Disable output on Phase Assert

Table 4-28 Output-to-Input Disable Settings

Relevant vi(s):

Set Driver Enable Control

Pass Fail Basis

This pull-down control programs the sequencer pass fail basis.

The control allows the user to select which error signal to use to determine the PASS/FAIL state for jumping.

Setting	Description
Local	Use local error
Qualified Local	Use CONDEN qualified local error
DTS	Use DTS error
Qualified DTS	Use CONDEN qualified DTS error

Table 4-29 Pass Fail Basis Settings

This is discussed in more detail in the **Pass/Fail Flag Operation** section in Chapter 6 including data rate limitations.

Relevant vi(s):

Set Pass Fail Parameters

Pass Valid Mode

This pull-down control programs the sequencer pass valid mode.

This control allows the user to define the Pass as a Valid Pass. A Valid Pass is one where no channel errors were detected but there must be at least one valid pattern expect code for each pattern in the sequence step.

This is discussed in more detail in the **Pass/Fail Flag Operation** section in Chapter 6 including data rate limitations.

Setting	Description
Disable	Do not use pass valid signal
Enable	Use pass valid signal

Table 4-30 Pass Valid Mode Settings

Relevant vi(s):

Set Pass Fail Parameters

Drive Fault

This pull-down control programs the sequencer Drive Fault mode.

If an output pin is enabled to also compare its state (Capture mode programmed and compare levels set), then a drive fault will be generated if the compare level does not match the output state. Drive faults are used with stimulus only pattern codes and can be used to detect dynamic over-current conditions.

If enabled a drive fault will disable all channels of the specified sequencer and a drive fault event will be generated.

Use **Query Sequencer Event** to query the drive fault event and **Query Sequencer Drive Fault** to query which channel caused the drive fault.

Setting	Description
Disable	Disable drive fault signal
Enable	Enable drive fault signal

Table 4-31 Drive Fault Settings

Relevant vi(s):

Set Driver Fault State

Attributes

This command button on the Configure>Data Sequencer>Settings panel displays the Attribute panel so that the sequencer attributes can be programmed.

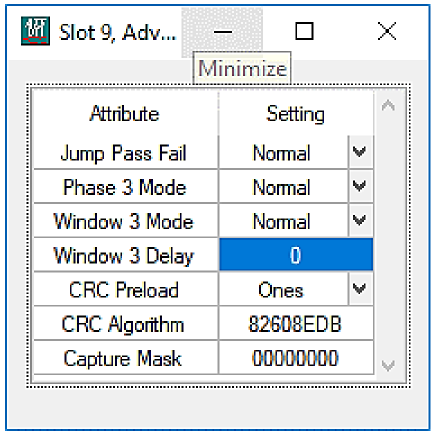


Figure 4-28 Attribute Panel

Jump Pass Fail

This control sets the sequencer step pass/fail accumulator mode.

Setting	Description
Normal	Enable the sequence step pass/fail accumulator (Default).
Legacy	Disable the sequence step pass/fail accumulator.

Table 4-32 Jump Pass Fail Settings

This is discussed in more detail in the **Pass/Fail Flag Operation** section in Chapter 6 including data rate limitations.

Relevant vi(s):

Set Sequencer Attribute

Phase 3 Mode

This control sets the phase 3 signal source.

Setting	Description	Typical Usage
Normal	Phase 3 is sourced from the internal phase generator. (Default)	Internally programmed timing for drive phases.
Jump 1	Phase 3 is sourced from the Jump 1 trigger signal.	Externally programmed timing controlled by an external stimulus clock tied to the Jump 1 Trigger source.

Table 4-33 Phase 3 Mode Settings

Normal mode sets the Phase 3 signal to be generated by the timing set phase generator.

Jump 1 sets the Phase 3 source to the Jump 1 Trigger. This allows external control of stimulus timing and/or pause and resume operation.

Relevant vi(s):

Set Sequencer Attribute

Window 3 Mode

This control sets the Window 3 signal source.

Setting	Description	Typical Usage
Normal	Window 3 is sourced from the internal window generator. (Default)	Internally programmed timing for response windows.
Jump 2	Window 3 is sourced from the Jump 2 trigger signal.	Externally programmed timing controlled by an external response clock tied to the Jump 2 Trigger source.

Table 4-34 Window 3 Mode Settings

Normal mode sets the Window 3 signal to be generated by the timing set window generator.

Jump 2 sets the Window 3 source to the Jump 2 Trigger. This allows external control of response timing and/or pause and resume operation.

Relevant vi(s):

Set Sequencer Attribute

Window 3 Delay

This control is used to delay the window 3 signal and is used when the "Window 3 Mode" attribute is set to Jump 2. Window 3 Delay is used to align an external response clock with the incoming response data.

The valid delay range is from 0 to 15 with 2ns resolution.

Relevant vi(s):

Set Sequencer Attribute

CRC Preload

This control sets the seed number for the CRC preload.

Setting	Description
Zeros	Preload 0’s
Ones	Preload 1’s
Masked	Mask Preload

Table 4-35 CRC Preload Settings

Relevant vi(s):

Set Sequencer Attribute

CRC Algorithm and Capture Mask

These numeric controls set the number for the CRC algorithm and for the CRC capture mask settings and are available in sequencer revision 0.23 and later.

Setting	Description
CRC Algorithm	A one in a bit position enables the corresponding CRC register bit feedback path.
Capture Mask	A one masks the corresponding channel's capture data (Error Signal, Pass Valid, Capture Fault, CRC and Drive Fault). Bit 0 corresponds to CH1 and bit 31 corresponds to CH32.

Table 4-36 CRC Algorithm and Mask Settings

Relevant vi(s):

Set Sequencer Attribute

Static State

This pull-down control programs the sequencer static state.

The static state is used to enable or disable the channel static mode setting.

Setting	Description
Off	Disable static operation
On	Enable static operation.

Table 4-37 Static State Settings

Refer to the **Static Data** section for details on executing I/O.

Relevant vi(s):

Set Static State

Configuring the I/O Channels

Configuring the channels is a three step process:

1. Select the channels.
2. Set the channel function.
3. Program channel parameters.
4. Configure channel properties.

Access this panel from the menu bar: **Config>Channels**.

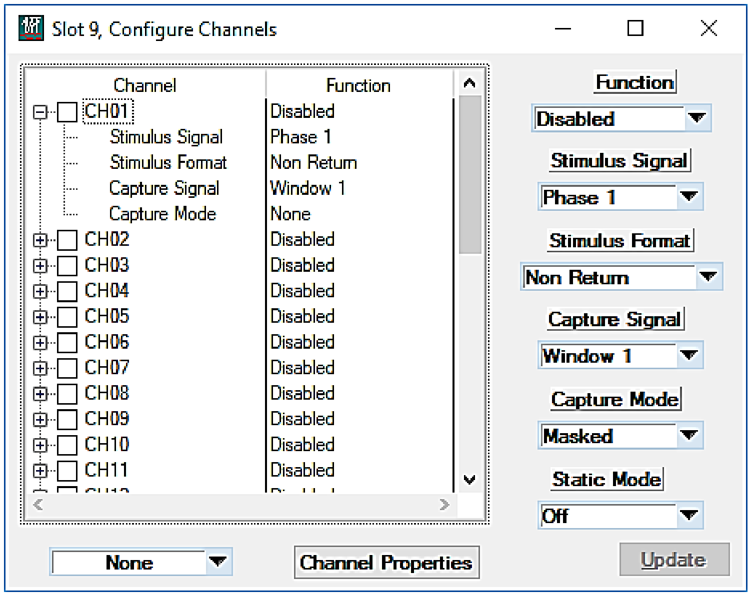


Figure 4-29 Configure Channels Panel

Selecting the Channels

Before the channel parameters or properties can be programmed, the channels must be selected. There are two methods for selecting the channels:

1. Left click on the desired channel in the channel list control. A check mark indicates the channel has been selected. Multiple channels can be selected.
2. Use the pull down list box to select the desired channels and press the **Select** command button. The choices include:

   - None – De-selects all channels.
   - All Channels – Selects CH1 through CH32.

CH1 - CH8 – Selects CH1 through CH8.

   - CH9 – CH16 – Selects CH9 through CH16.
   - CH17 – CH24 – Selects CH17 through CH24.
   - CH25 – CH32 – Selects CH25 through CH32.

Channel Parameters

The channel parameters consist of:

- Channel Function
- Stimulus Signal
- Stimulus Format
- Capture Signal

- Capture Mode
- Static Mode

After any of the channel parameters have been changed, the **Update** command button must be depressed in order for the new channel settings to be programmed.

Note

Stimulus Signal, Stimulus Format, Capture Signal and Capture Mode are only applicable for channels assigned as Dynamic HiZ or Dynamic VTT.

Channel Function

This pull-down control sets the channel function.

The channel function determines the pin functionality.

Setting	Description
Disabled	The channels pin electronics are set to a low power mode and channel is unusable.
Dynamic HiZ	Channel output controlled by the data sequencer two level programmable output with a high impedance disabled state.
Dynamic VTT	Channel output controlled by the data sequencer two level programmable output with a programmable third level disabled state.
PMU FV	Channel set to PMU force voltage mode.
PMU FI	Channel set to PMU force current mode.

Table 4-38 Channel Function Settings

Relevant vi(s):

Set Channel Function

Stimulus Signal

This pull-down control programs the drive phase timing for the selected channel(s) stimulus signal.

Setting	Description
Phase 1	Use phase 1 timing signal to control output driver timing.
Phase 2	Use phase 2 timing signal to control output driver timing.
Phase 3	Use phase 3 timing signal to control output driver timing.
Phase 4	Use phase 4 timing signal to control output driver timing.

Table 4-39 Stimulus Signal Settings

Relevant vi(s):

Set Channel Parameters

Stimulus Format

This pull-down control programs the stimulus data formatting for the selected channel(s).

Setting	Stimulus Format Description
Non Return	Phase Assert – Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return – No action.
Return Off	Phase Assert – Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return – Output driver disables.
Return Zero	Phase Assert – Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return – Output driver goes to low level.
Return One	Phase Assert – Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return – Output driver goes to high level.
Return Comp	Phase Assert – Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return – Output driver goes to complemented level determined by the Pattern Code instruction in Pattern Memory
Comp Surround	Start of Pattern – Output driver goes to complemented level determined by the Pattern Code instruction in Pattern Memory Phase Assert – Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return –Output driver goes to complemented level determined by the Pattern Code instruction in Pattern Memory **Note:** For this format to work effectively, the assert must be at least 15 ns (depends on the swing and slew-rate programmed).
Force Low	Output driver goes to low level immediately after an update.
Force High	Output driver goes to high level immediately after an update.
Force Off	Output driver goes disables immediately after an update.
Force /Phase	Phase Assert – Output driver goes from high to low level. Phase Return – Output driver goes from low to high level. Output driver coincides with the complement of the phase immediately after an update.
Force Phase	Phase Assert – Output driver goes from low to high level. Phase Return – Output driver goes from high to low level. Output driver coincides with the phase immediately after an update.

Table 4-40 Stimulus Format Settings

The last five settings, above, will only go to the new output state if the Channels drivers are enabled and power is applied. See **Channel Driver** and **V+/V-** in the **Execute Panel Modes and Settings** section of this chapter.

Relevant vi(s):

Set Channel Parameters

Capture Signal

This pull-down control programs the selected channel(s) capture signal.

Setting	Description
Window 1	Use Window 1 timing signal to control input comparator timing.
Window 2	Use Window 2 timing signal to control input comparator timing.
Window 3	Use Window 3 timing signal to control input comparator timing.
Window 4	Use Window 4 timing signal to control input comparator timing.

Table 4-41 Capture Signal Settings

Relevant vi(s):

Set Channel Parameters

Capture Mode

This pull-down control programs the selected channel(s) capture mode.

Setting	Description
Masked	Disables the channel error test
Open Edge	Channel error test and data capture performed on the Open edge of the window
Close Edge	Channel error test and data capture performed on the Close edge of the window
Window	Channel error test and data capture performed between the Open edge and the Close edge of the window

Table 4-42 Capture Mode Settings

Window mode requires that the channel must match the “expect” for the duration of the window.

Relevant vi(s):

Set Channel Parameters

Static Mode

This pull-down control programs static mode for the selected channel(s).

When the Static Mode Enable is set to on, the designated channel is put into the Static Mode and whatever is currently in the Static Broadside Stimulus Register will be applied to the output. Channels not in Static Mode will operate in the normal dynamic mode. When the channel is returned from Static to Dynamic Mode, dynamic operation will resume as though it had never been put into the Static Mode.

Setting	Description
Off	Static Mode enabled for selected channel(s).
On	Static mode disabled for selected channel(s.

Table 4-43 Static Mode Settings

Note: *The static state must be enabled before setting the static mode.*

Relevant vi(s):

Set Static Mode

Channel Properties

This command button allows the user to configure the driver/receiver properties.

The channel properties consist of the following seven elements:

1. Driver Levels
2. Comparator Levels
3. Driver Slew Rate
4. Output Impedance
5. Programmable Load
6. Channel Connect
7. Channel Mode

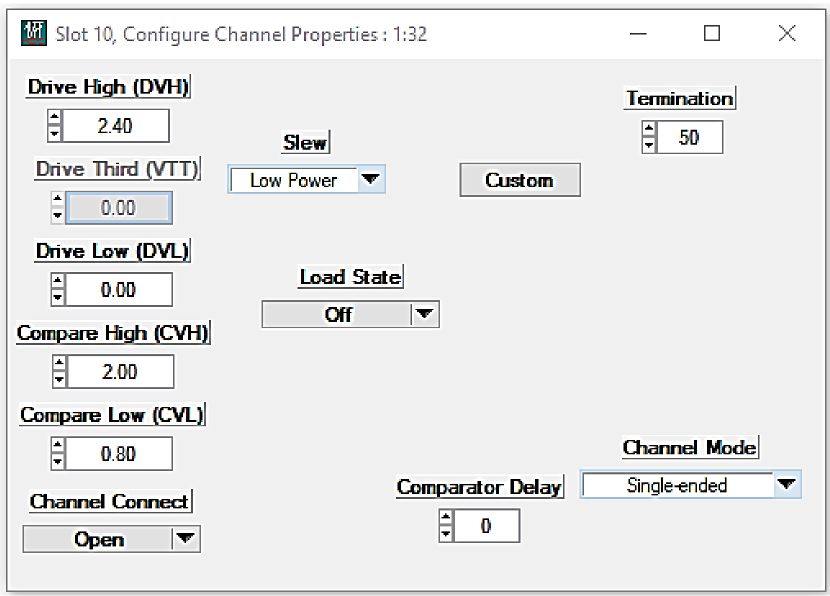


Figure 4-30 Configure Channel Properties Panel

The title bar displays the channels assigned to this panel. The property values displayed are from the first channel. If a subsequent channels properties are not the same, the control(s) text will be red and an **Update** button will be visible.

Depressing the Update button will program all the channels properties the same.

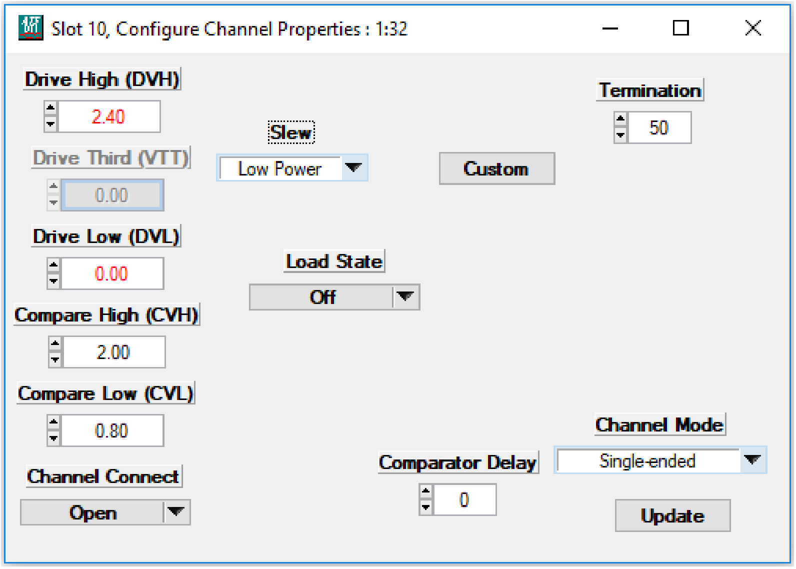


Figure 4-31 Configure Channel Properties Update

Driver Levels

The driver levels allow the user to set the **Drive High (DVH), Drive Low (DVL)** and **Drive Third (VTT)** voltage level for channels assigned as Dynamic HiZ or Dynamic VTT.

The absolute min/max levels are listed below:

- Drive High -1.5 to +7.0
- Drive Low -2.0 to +6.0
- Drive Third -2.0 to +7.0

The IO Max settings determine the max limit, page 4-15.

Drive High must be at least 0.5V greater that Drive Low.

Drive Third is only used for channels that are set to Dynamic VTT.

Relevant vi(s):

Set Channel Source Levels

Set Channel Source VTT

Comparator Levels

The comparator levels allow the user to set the **Compare High (CVH)** and **Compare Low (CVL)** voltage level for channels assigned as Dynamic HiZ or Dynamic VTT.

The absolute min/max levels are listed below:

- Compare High -2.0 to +7.0
- Compare Low -2.0 to +7.0

The IO Max settings determine the max limit, page 4-15.

Relevant vi(s):

Set Channel Sense Levels

Driver Slew

The driver slew allows the user to set the output **Slew Rate** for channels assigned as Dynamic HiZ or Dynamic VTT.

Setting	Description
Fast	Sets the slew rate to ~1.3 V/ns
Medium	Sets the slew rate to ~1.0 V/ns
Default	Sets the slew rate to ~0.7 V/ns
Slow	Sets the slew rate to ~0.25 V/ns
Low Power	Sets the slew rate to <0.1 V/ns

Table 4-44 Slew Settings

Depressing the **Custom** command button allows the user to specify the **+ Slew Rate**, **- Slew Rate** and **Fine**.

The range for the **+ Slew Rate** and **– Slew Rate** is from 0 (slowest) to 31 (fastest).

The range for the **Fine** is 0 (slowest) to 7 (fastest). The fastest slew rate would be with a value of 31 and a fine of 7.

Higher slew rates require more power and generate more heat.

Relevant vi(s):

Set Channel Slew Rate

Set Channel Source Parameters

Termination

The driver has a nominal output impedance of 50Ω for channels assigned as Dynamic HiZ or Dynamic VTT. The termination can be adjusted from 35Ω to 66Ω.

Relevant vi(s):

Set Channel Source Parameters

Programmable Load Settings

The programmable load settings allows the user to specify load state, active load type, commutating voltage (VCOM) level and source/sink current load.

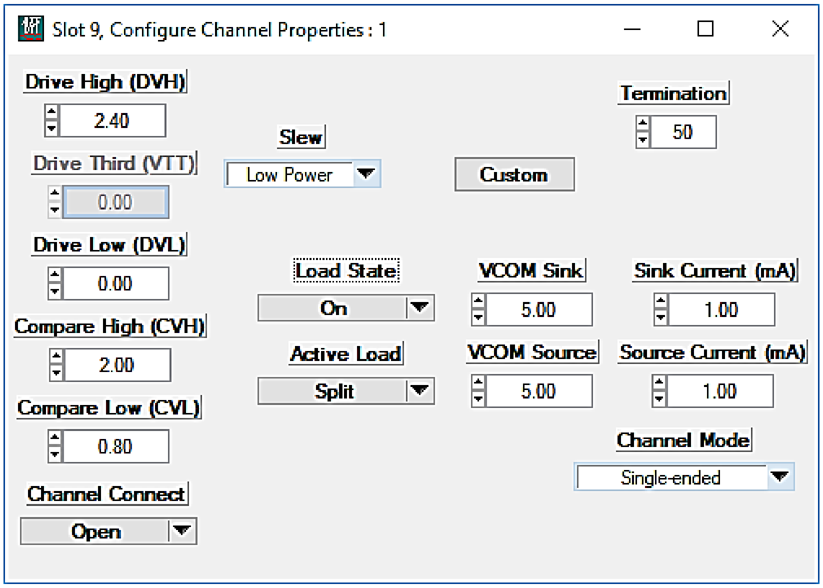


Figure 4-32 Channel Properties Programmable Load Settings

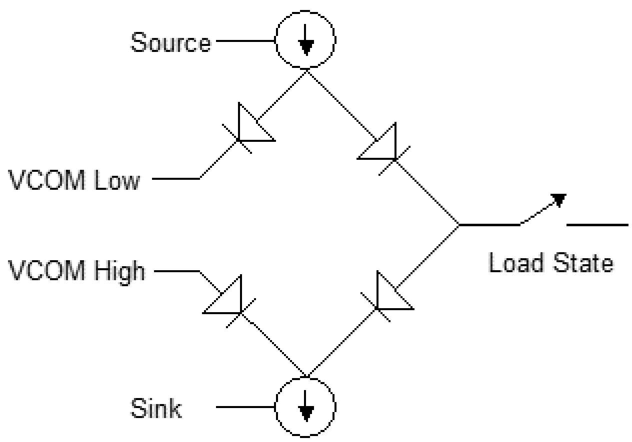


Figure 4-33 Current Load

When the channel voltage is greater than the VCOM Sink level, the Sink current becomes active. When the channel voltage is less than the VCOM Source level, the Source current becomes active.

Load State

This control programs the channel(s) load state. Only channels assigned as Dynamic HiZ can enable the active load.

Setting	Description
Off	No active load
On	Programmable current load on always.
HiZ	Programmable current load when driver not enabled.

Table 4-45 Load State Settings

Relevant vi(s):

Set Channel Sense Parameters

Set Channel Load State

Active Load

This control specifies the active load configuration.

Setting	Description
Split	Separate VCOM for source and sink.
Single	Single VCOM for source and sink.

Table 4-46 Active Load Settings

Relevant vi(s):

Set Channel Source Parameters

VCOM Sink

This specifies the VCOM Sink level. The VCOM Sink can be set from 0V to 5V but must be greater than or equal to VCOM Source.

Relevant vi(s):

Set Channel Source Parameters

VCOM Source

This specifies the VCOM Source level. The VCOM Source can be set from 0V to 5V but must be less than or equal to VCOM Sink.

Relevant vi(s):

Set Channel Source Parameters

Sink Current (mA)

This specifies the current level that will be applied if the channel voltage is greater the VCOM Sink. The current can be set from 0mA to 24mA.

Relevant vi(s):

Set Channel Source Parameters

Source Current (mA)

This specifies the current level that will be applied if the channel voltage is greater the VCOM Source. The current can be set from 0mA to 24mA.

Relevant vi(s):

Set Channel Source Parameters

Channel Connect

This control allows the user to control the isolation relays.

Setting	Description
Open	Isolation Relay Open
Closed	Isolation Relay Closed

Table 4-47 Channel Connect Settings

Relevant vi(s):

Set Channel Connect

Comparator Delay

This allows the user to add delay to the comparator inputs.

The range for **Comparator Delay** is from 0 to 1023 (~10.23ns) 10ps per count.

Relevant vi(s):

Set Comparator Delay

Channel Mode

This control programs the channel mode, which sets the comparator path that determines the good 1 and good 0 levels.

The comparator path can be set to:

- Single-ended
- Differential no termination

The differential path selects the comparator that uses adjacent odd and even channels. The single ended path uses dual comparators with the CVL and CVH thresholds. Figure 4-34 illustrates the channel mode logic.

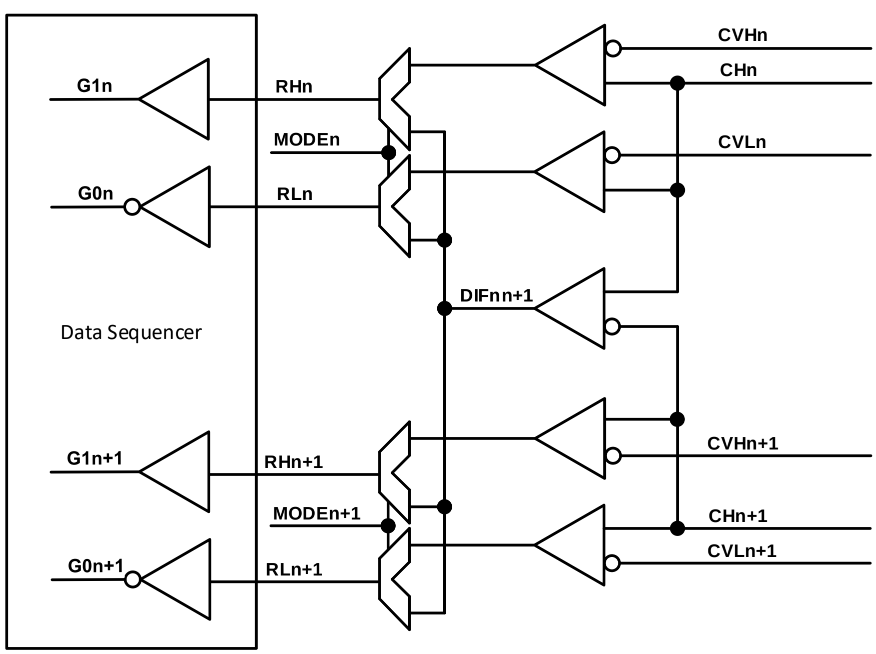


Figure 4-34 Channel Mode Logic

The following tables list the good 0 (G0) and good 1 (G1) logic states for both single ended and differential operation.

Single Ended	RH	RL	G1	G0
CH > CVH	1	X	1	X
CH < CVH	0	X	0	X
CH > CVL	X	1	X	0
CH < CVL	X	0	X	1

Table 4-48 Single Ended Comparator Logic States

Differential	RH	RL	G1	G0
CHn > CHn+1	1	1	1	0
CHn < CHn+1	0	0	0	1

Table 4-49 Differential Comparator Logic States

Adjacent channels can select different comparator paths.

Relevant vi(s):

Set Channel Mode

Configuring the AUX Channels

Access this panel from the menu bar: **Config>AUX Outputs**.

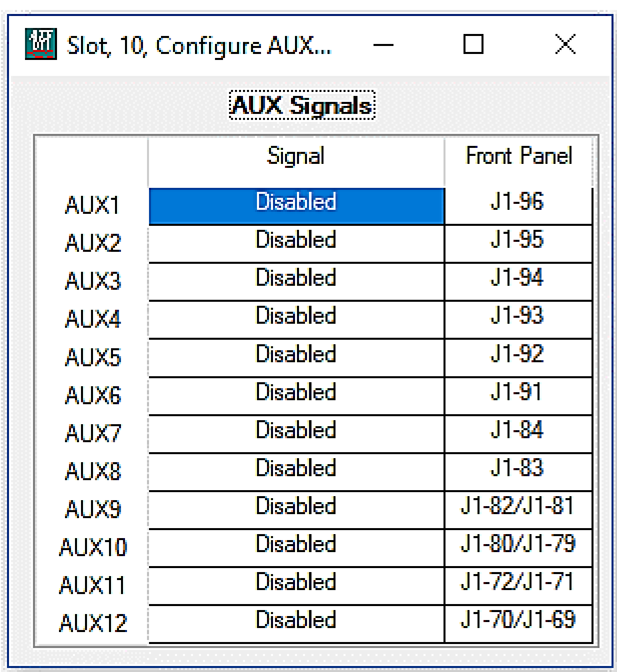


Figure 4-35 Configure AUX Channels Panel

The AUX channels are a set of 12 multi-purpose signals that can be used for any of the following I/O resources:

1. Trigger Source Input
2. Frequency Synthesizer Reference Clock Input
3. System Clock Input
4. Vector Jump Address Input
5. Waveform Output
6. Pulse Generator Output
7. Sync Output
8. Frequency Synthesizer Output
9. Timing Set Output Signals
   a. Phase
   b. Window
   c. T0_CLK
   d. Pattern Clock
10. Sequencer Status Outputs
    a. Idle Active
    b. Sequence Active
    c. Sequence Flag
    d. Pass/Fail
    e. Error
11. Numerous Factory Test Outputs

Configuring the AUX Signals

Configuring the AUX signals is done by double clicking the left mouse button on the signal name corresponding to the desired AUX number.

All AUX signals share the controls listed in the following figure:

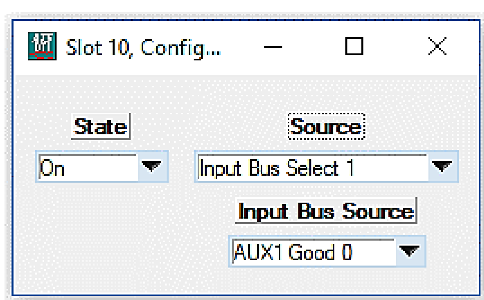


Figure 4-36 AUX Controls

State

This control allows the user to set the output state for the selected AUX signal.

Setting	Description
Off	Disable the AUX output.
On	Enable the AUX output.
Inv	Enable and invert the AUX output.

Table 4-50 AUX Output State Settings

Relevant vi(s):

Set Aux Output Signal

Source

This control is visible when the state is set to **On** or **Inv** and allows the user to set the output source for the selected AUX signal.

Setting	Description
Phase 1-4	Phase timing signal.
Window 1-4	Window timing signal.
Waveform 1-4	Waveform signal.
Sync 1,2	Sync signal.
Idle Active	1 = Active, 0 = Not Active.
Sequence Active	1 = Active, 0 = Not Active.
Channel good 1*	Channel good 1 comparator signal
Channel good 0*	Channel good 0 comparator signal
Waveform 5	Waveform 5 signal
Waveform 6	Waveform 6 signal
Input Bus Select 1-4**	Input Bus Select Signal
Seq. Flag 1,2	Sequence flag signal.
T0CLK_In	Test signal
Pattern Clock	Test signal
SEQ_CLK In	Test signal
Jump In	Test signal
Raw Error	Test signal
SEQ_CLK_D_In	Test signal
T0CLK Out	Test signal
SEQ_CLK Out	Test signal
Jump Out	Test signal
SEQ_CLK_D_Out	Test signal
Pulse Generator	Pulse generator signal
Record Active	1 = Active, 0 = Not Active.
FS Reference	Frequency synthesizer reference signal
Frequency Synthesizer	Frequency synthesizer signal
Jump Strobe	Test signal
Int Error	Test signal

Setting	Description
Ext Error	Test signal
HIGH	Drive high
PASS	PASS flag
FAIL	FAIL flag
CONDEN	Condition enable flag
BERREN	Burst error enable flag
LSR	Load Sequence Register
LLC	Load Loop Count
CA	Counter Active
CPPD	Clocks per Pattern Done
BCD	Burst Count Done
LCD	Loop Count Done
IN_SUB	Gosub Active
C_LOOP	Counted Loop
SUBRT	Subroutine Return
RTN	Return Flag
LSTSEQ	Last Sequence
Jump Test 1-4	Test signal

Table 4-51 AUX Source Settings

*The Channel Good 1/Channel Good 0 selections can select any of the front-end channels using **Set Aux Channel Select** vi.

The Input Bus Select selections can select any of the AUX, TTL trigger, or Channel Test 1 using **Set Aux Input Bus Select vi.

Relevant vi(s):

Set Aux Output Signal

Set Aux Channel Select

Set Aux Input Bus Select

Input Bus Source

This control is visible when the Source control is set to one of the four **Input Bus Select** signals. It selects the source for the selected input bus select.

Setting	Description
AUX1 Good 0	Source set to AUX1 Good zero signal.
AUX1-12 Good 1	Source set to AUXn Good one signal.
CHT1	Source set to channel test 1
TTLTRG0-7	Source set to PXI TTL trigger

Table 4-52 Input Bus Select Source Settings

Editing the Data Sequencer

Editing the data sequencers consists of programming the following:

1. Timing Sets
2. Patterns
3. Waveforms
4. Sequence Parameters
5. Sequence Steps

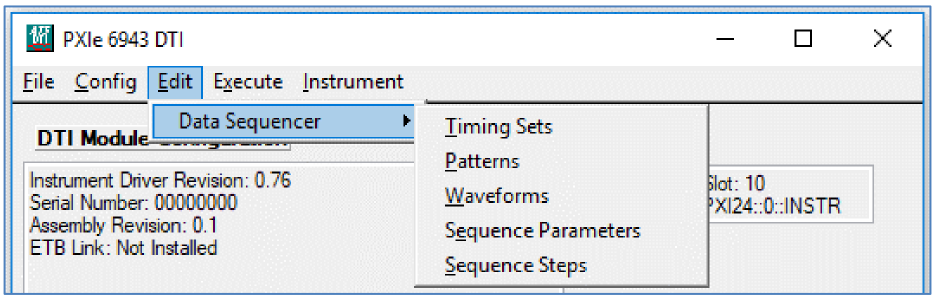


Figure 4-37 Editing the Data Sequencer

Editing the Timing Sets

The timing sets are used to control the channel drivers and receivers. Each timing set has either one or four phase/window groups based on the programmed timing mode, refer to **Set Error Parameters**

Timing Mode section.

Phases control the driver timing and consist of an **Assert** and a **Return.** The **Assert** signal loads the next pattern code in to the output driver. Pattern codes are discussed in the next section. The **Return** signal is used to load the format code in to the output driver. The **Return** signal is not used for the Non Return format code. See **Set Channel Parameters**

Stimulus Format earlier in this chapter.

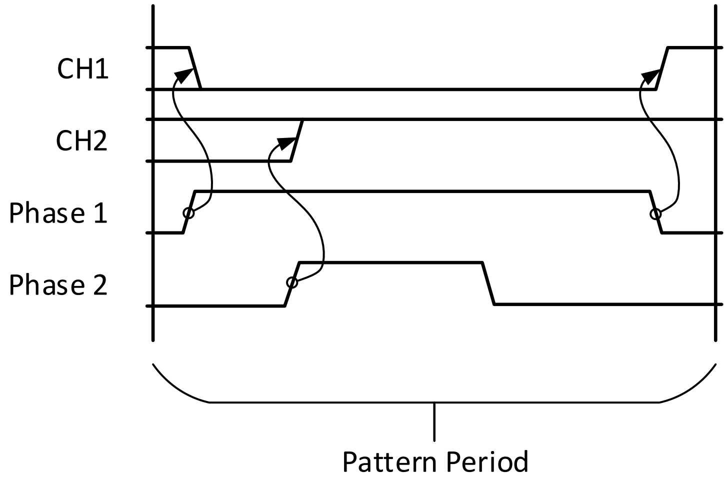


Figure 4-38 Phase Timing

The figure above represents two channels with the following configuration:

- CH1 Output Signal = Phase 1
  Stimulus Format = Return to One
  Pattern Code = Drive Low
- CH2 Output Signal = Phase 2
  Stimulus Format = Non Return
  Pattern Code = Drive High

The **Assert** signal (rising edge) causes the pattern code to be loaded in to the output register. The **Return** signal (falling edge) causes the Stimulus Format to output. Since CH2 is set to Non Return, the Return signal did not affect the output level.

Windows control the response timing and consist of an **Open** and a **Close** edge. The capture mode (see **Set Channel Parameters**

Capture Mode section) determines what edge (**Open**, **Close** or **Both**) samples the data from the input receiver for each channel and compares the results with the expect code.

There are two panels that display the timing set data, **Edit Timing Sets** and **Edit Sequence Step Timing** command button.

Access the **Edit Timing Sets** panel from the menu bar: **Edit>Data Sequencer>Timing Sets**.

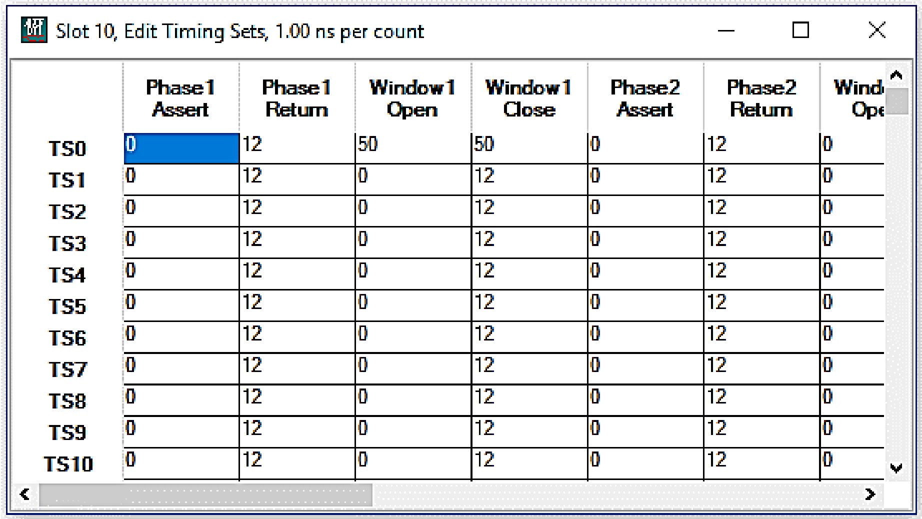


Figure 4-39 Edit Data Sequencer Timing Sets Panel

To program a timing set, scroll down the list until the desired timing set number is visible. Timing set numbers are assigned based on the current timing mode:

- Per Step Multi – 1024 timing sets with four phase/window groups per timing set. TS0 is the timing for sequence step 1, TS1 is the timing for sequence step 1, … , timing set 1023 is the timing for sequence step 1023.
- Per Step Single - 4096 timing sets with one phase/window group per timing set. TS0 is the timing for sequence step 1, TS1 is the timing for sequence step 1, … , timing set 4095 is the timing for sequence step 4095.
- Indexed – 256 timing sets with four phase/window groups per timing set and 4096 sequence steps where each sequence step points to one of the 256 timing sets.

Double-click on one of the available Assert/Return/Open/Close cells. Enter the desired value using the numeric keys or the up/down arrows followed by the Enter key.

Access the **Edit Sesuence Step** panel from the menu bar: **Edit>Data Sequencer>Sequence Steps** and select the sequence step to edit by double clicking on the row. From the **Edit Sequence** Step panel depress the **Timing** command button.

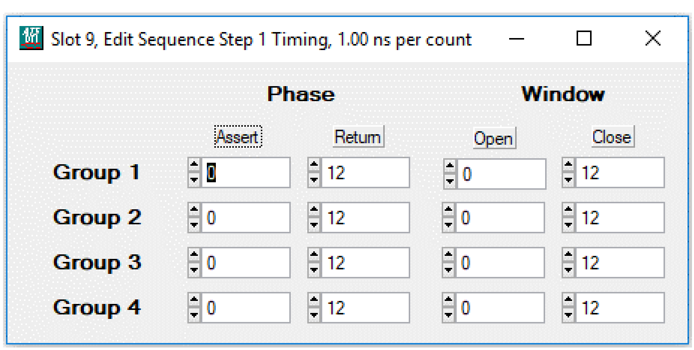


Figure 4-40 Edit Sequence Step Timing Set Panel

On both panels, the timing resolution is displayed in the title bar area of the panel and is the Master Clock period divided by two.

The user can disable the timing set phases/windows by setting Assert/Return and Open/Close values to zero. For example, Phase 1 and Window 1 are disabled during TS2 in the configuration shown below.

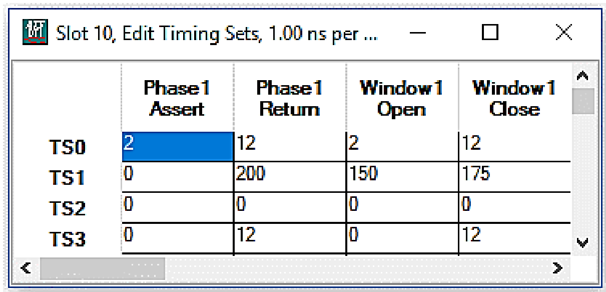


Figure 4-41 Disable Phase/Window

Relevant vi(s):

Set Timing Set Data

Timing Set Value Rules

For valid timing signal operation, the following rules must be followed:

- Phase pulse width must be greater than seven, i.e., the Return value must be at least eight more than the Assert value.
- Window pulse width must be greater than seven, i.e., the Close value must be at least eight more than the Open value.
- End of pattern dead time. Phase Return and Window Close values must occur eight counts or more before the end of the pattern. Additionally a Window Close must occur 13 ns prior to the end of the pattern period.
- Phases and Windows are allowed to extend past the initial pattern period if multiple clocks per pattern (CPP > 1) are programmed. (See **Clocks per Pattern** later in this chapter.)
- See **CPP Phase and Window Triggering** for details on triggering options.

Advanced Timing Set Features

Two advanced timing set features are available:

1. Phase/Window Spanning
2. Idle/Standby Timing

Phase/Window Spanning

Phase/Window spanning Asserts/Opens the timing signal in one pattern and Return/Close the signal in a different pattern. The following steps describe how to span timing signals across multiple patterns:

1. Disable the Return signal in the first pattern's timing set by setting the Return value equal to the pattern period.
2. Disable the Assert and Return signal in any patterns between the first and the last pattern being spanned by setting the Assert Value to zero and the Return value equal to the pattern period.
3. Disable the Assert signal in the last pattern by setting the Assert Value to zero.

For example, let's assume we have three patterns and each pattern has a period of 100. We want the Phase 1 Assert at 50 of the first pattern and Return at 75 of the third pattern.

Pattern 1, TS1 = Assert 50, Return 100

Pattern 2, TS2 = Assert 0, Return 100

Pattern 3, TS3 = Assert 0, Return 75

Idle/Standby Timing

One of the unique features of the DRM is the Idle/Standby state. After the execution of a sequence burst, the sequencer will enter the Idle/Standby state. The user can define the Idle/Standby state timing and pattern such that UUT stimulus can be maintained between pattern bursts. A single pattern can be specified so that the pattern memory can be updated (Standby) or a group of patterns can be specified (Idle) during this state.

The user can disable the timing set phases/windows during the Idle/Standby state by setting Assert/Return and Open/Close values to zero.

Editing the Patterns

Patterns are the memory element that contains the instructions for each channel during a sequence burst. These instructions, called pattern codes, define whether a channel will drive high, drive low, test high, etc.

Once a sequence step has been initialized, a pattern set is assigned to the step. A **Pattern Set** is one or more patterns. A **Pattern** is the pattern codes for all the channels that will be applied at the same time. (See **Patterns** in this chapter.)

Access this panel from the menu bar: **Edit>Data Sequencer>Patterns**.

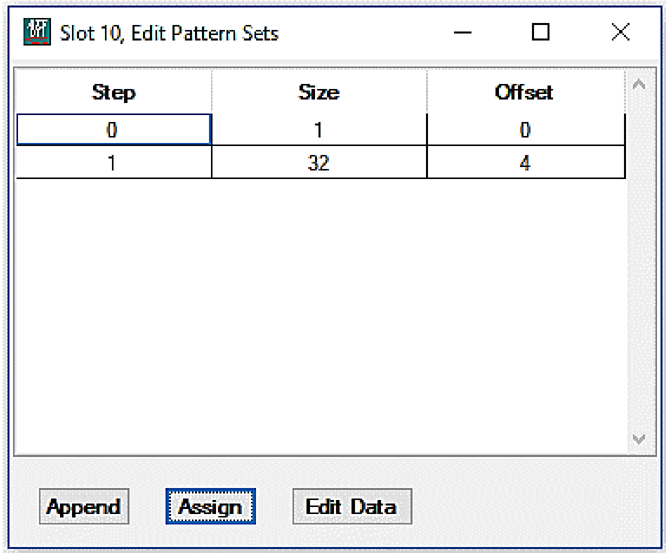


Figure 4-42 Edit Patterns Panel

This panel lists all the defined pattern sets. The associated step number, size and offset are displayed.

The size of a pattern set can be from 1 to 262144.

The offset can be from 0 to 262140 and must be a multiple of four.

Relevant vi(s):

Query Pattern Set

Query Pattern Set List

Append

This control appends more patterns to the selected pattern set.

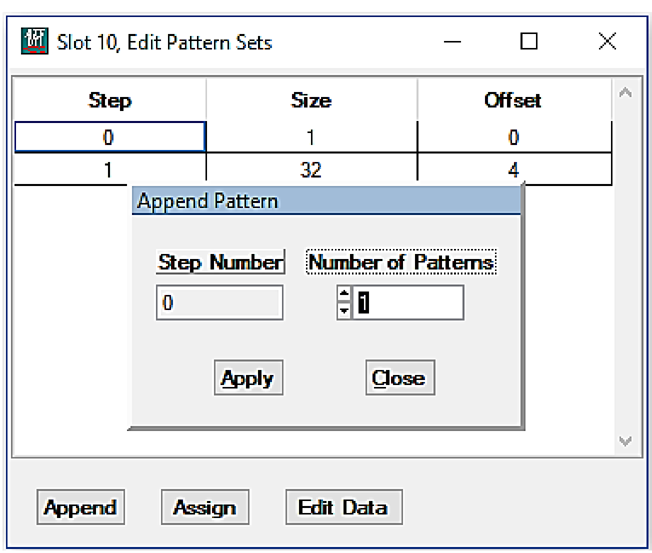


Figure 4-43 Append Data Sequencer Pattern Sets Panel

Enter the **Number of Patterns** to append and press the **Apply** command button. Append pattern memory will be initialized to Pattern Code “R”, which repeats the previous code.

The driver allows pattern set overlaps when appending patterns. If you don’t want pattern sets to overlap, make sure there’s enough space for the appended patterns. This can be facilitated by assigning the pattern offset initially (see **Assign** function next).

Press the **Close** command button to exit the panel without any changes.

Relevant vi(s):

Append Pattern

Assign

This control assigns a new size and/or offsets the selected pattern.

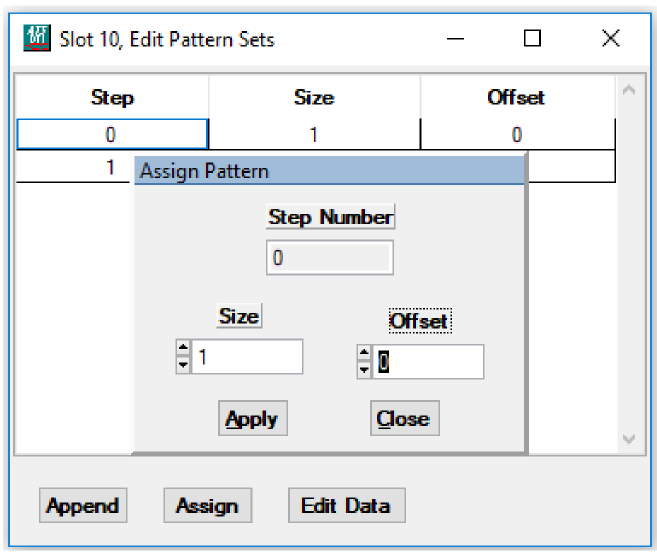


Figure 4-44 Assign Data Sequencer Pattern Sets Panel

Enter the new **Size** and/or **Offset** and press the **Apply** command button. Assigned pattern memory will not be initialized.

Press the **Close** command button to exit the panel without any changes.

Relevant vi(s):

Assign Pattern Set

Edit Data

This control displays the view/edit pattern set panel. This panel allows the user to view/edit the contents of the pattern set memory. Double-clicking on the desired pattern set can also open this panel.

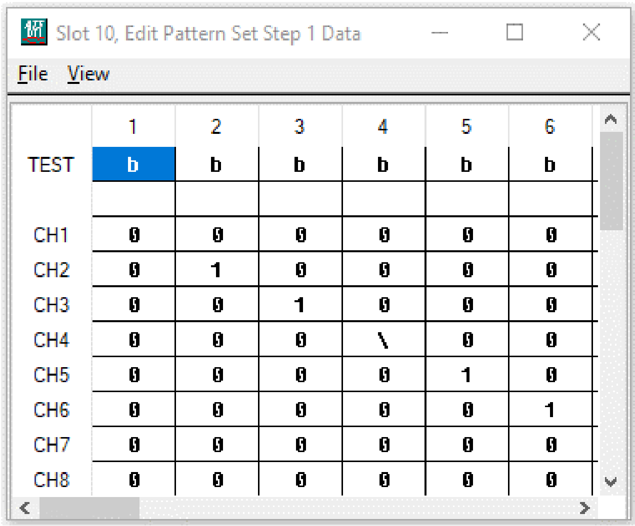


Figure 4-45 Pattern Set Sequencer Data Panel

Each column contains the TEST code and the pattern codes for all the channels. The pattern codes are described in Figure 4-48 and Table 4-53.

The pattern set is displayed in pages of 32 patterns. The View menu bar lists the page control shortcuts listed below:

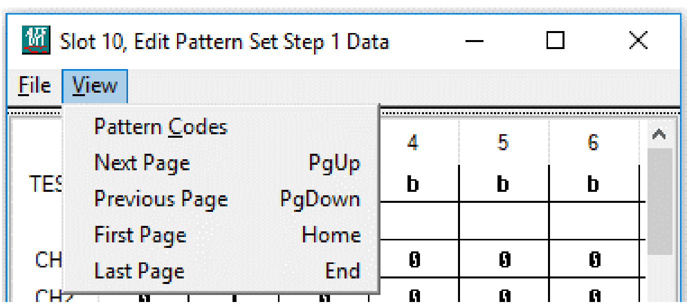


Figure 4-46 Pattern Set Data – View Menu

To jump to a specific pattern number, right click in any of the cells to display the **Goto Pattern** panel

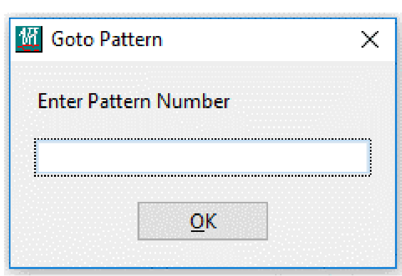


Figure 4-47 Goto Pattern Panel

The menu bar: **View > Pattern Codes** displays a legend of all the available TEST and CH entries.

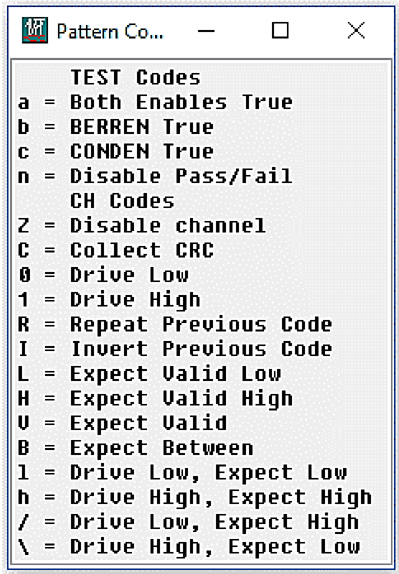


Figure 4-48 Pattern Codes

The row labeled “TEST” displays the test code for each pattern. There are two test flags per pattern:

1. BERREN – Burst Error Enable. This flag allows the user to designate which patterns will be examined for Burst Error, Burst Error counting and the logging of errors in the Error Address Memory.
2. CONDEN – Condition Enable. This flag allows the user to designate which patterns will be considered for PASS/FAIL jump tests.

The rows labeled CH1 through CH*n* contain the pattern codes for the specified channels. There are fourteen pattern codes. The following table lists how each pattern code affects the driver/comparator.

Pattern Code	Driver		Comparator Expect	Invert Code
	Mode	Level		
Disable Channel ‘Z’	Off	X	None	Disable Channel ‘Z’
Collect CRC ‘C’	Off	X	Enable CRC	Collect CRC ‘C’
Drive High ‘1’	On	DVH	None	Drive Low ‘0’
Drive Low ‘0’	On	DVL	None	Drive High ‘1’
Repeat Previous Code ‘R’	Repeats the last non repeat/invert code.			
Invert Previous Code ‘I’	Inverts the last non repeat/invert code. Refer to Invert Code column of this table.			
Expect Valid Low ‘L’	Off	X	< CVL	Expect Valid High ‘H’
Expect Valid High ‘H’	Off	X	> CVH	Expect Valid Low ‘L’
Expect Valid ‘V’	Off	X	< CVL or > CVH	Expect Between ‘B’
Expect Between ‘B’	Off	X	> CVL and < CVH	Expect Valid ‘V’
Drive Low, Expect Low ‘l’	On	DVL	< CVL	Drive High, Expect High ‘h’
Drive High, Expect High ‘h’	On	DVH	> CVH	Drive Low, Expect Low ‘l’

Drive Low, Expect High '/'	On	DVL	> CVH	Drive High, Expect Low '\'
Drive High, Expect Low '\'	On	DVH	< CVL	Drive Low, Expect High '/'

Table 4-53 Pattern Codes

Relevant vi(s):

Set Pattern Data

Set Pattern Test Enable

Import/Export Data

The pattern data can be imported/exported using the **File** menu bar selection.

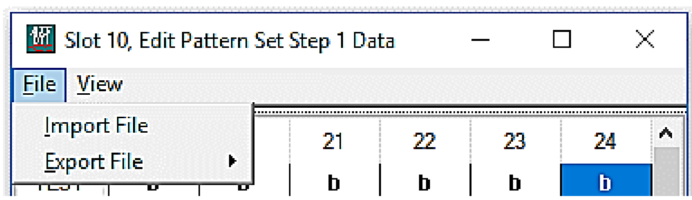


Figure 4-49 Pattern Set Data – File Menu

The import/export formats include:

- Pattern data as ASCII Hex
- Pattern data as ASCII String
- Pattern data as Binary
- Pattern data and flags as ASCII Hex
- Pattern data and flags as ASCII String
- Pattern data and flags as Binary

Relevant vi(s):

Save Pattern Memory

Load Pattern Memory

Import/Export File Format

The import/export file format consists of a header followed by the data.

The header identifies the number of patterns and the format, and must be the first line of the file.

Header Format

The format of the header is:

[ATS6943 PAT DUMP <dd> <nnnnnn>]

where:

<dd> is the format;

00 = Pattern Data ASCII Hex.

01 = Pattern Data Binary

02 = Pattern Data ASCII String

03 = Pattern Data and Flags ASCII Hex.

04 = Pattern Data and Flags Binary

05 = Pattern Data and Flags ASCII String

<nnnnnn> is the number of patterns.

Data Format

The data format consists of three types, ASCII hex, Binary and ASCII string. In addition, each of the three data formats can include or exclude the pattern flags.

ASCII Hex

The ASCII hex format represents pattern data as viewable ASCII hex characters, one character per channel. The following table lists the pattern code to ASCII Hex value translation.

Pattern Code	ASCII Hex Value
‘Z’	0
‘C’	1
‘0’	2
‘1’	3
‘R’	6
‘I’	7
‘L’	8
‘H’	C
‘V’	D
‘B’	9
‘l’	A
‘h’	F
‘/’	E
‘\’	B

Table 4-54 ASCII Hex File Data Format

Flag Code	Bit15, Bit 14 Code
‘a’	3
‘b’	2
‘c’	1
‘n’	0

Table 4-55 ASCII Hex File Flag Format

The ASCII characters are in four groups of eight characters and one line per pattern. A fifth column of four characters is present if flags are included.

00000002 00000000 00000000 00000000 80000000
00000000 30000000 00000000 00000000 80000000

The first column contains the data for channels 8 through 1.

The second column contains the data for channels 16 through 9.

The third column contains the data for channels 24 through 31.

The fourth column contains the data for channels 32 through 25.

The fifth column contains the flag data.

Each column contains the pattern code for eight channels; the least significant channel data is the right most hex character in each column. In the example above, all channels are set to 'Z' except channel 1 is set to '0' in pattern one. In pattern two all channels are set to 'Z' except channel 16 is set to '1'.

Binary

The binary format represents the pattern data as raw binary data. The pattern data is stored in four sequential 32 bit blocks, five if flags are included. The block order is listed below.

Block Number	Contents
1	Channel 8 through 1
2	Channel 16 through 9
3	Channel 24 through 17
4	Channel 32 through 25
5	Flags

Table 4-56 Binary Block Format

In blocks one through four, each 32 bit value contains eight pattern codes. The pattern code for each channel requires four bits. The channel mapping for each block is from the lowest channel to the highest channel, i.e., bits 0-3 are channel 1 in block 1, bits 4-7 are channel 2 in block 1, etc.

In block five, each 32 bit value contains the flag codes. The flag code for each pattern requires two bits. Bits 15 and 14 contain the flag code.

Pattern Code	Binary Value
'Z'	0000
'C'	0001
'0'	0010
'1'	0011
'R'	0110
'I'	0111
'L'	1000
'H'	1100
'V'	1101
'B'	1001
'I'	1010

Pattern Code	Binary Value
‘h’	1111
‘/’	1110
‘\’	1011

Table 4-57 Binary File Data Format

Flag Code	Bit15, Bit 14 Code
‘a’	11
‘b’	10
‘c’	01
‘n’	00

Table 4-58 Binary File Flag Format

ASCII String

The ASCII string format represents pattern data as viewable ASCII strings, one character per channel, 32 characters per line (34 if flag data is included). Each character is one of the pattern codes listed in Table 5-44. The following example lists two patterns.

aZ000RRRRRRRRRRRRRRRRRRRRRRRRRRRRm

bCRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRn

The flag code will be the first character followed by channel 1 through channel 32 and ending with probe expect.

In this example, pattern one has:

- Both flags set (‘a’)
- Channel 1 disabled (‘Z’)
- Channel 2 through channel 4 driven low (‘0’)
- Channel 5 through channel 32 repeating the previous state (‘R’)
- Last character ‘m’ is don’t care.

Pattern two has:

- BERREN flag set (‘b’)
- Channel 1 enabling the CRC (‘C’)
- Channel 2 through channel 32 repeating the previous state (‘R’)
- Last character ‘n’ is don’t care.

Editing Waveforms

Up to six waveforms can be defined and output during a pattern for generating UUT handshake or clock stimulus. The first four waveforms are enabled per sequence step and they replace certain Phase/Window signals as mapped below:

- Waveform 1 – Mapped to Phase 4
- Waveform 2 – Mapped to Window 4
- Waveform 3 – Mapped to Phase 3
- Waveform 4 – Mapped to Window 3

Waveforms 1-4 can be programmed to generate complex waveforms with as many transitions that can fit in the pattern period.

The last two waveforms (Waveform 5 and Waveform 6) are not mapped to any of the phase or window signals but are limited to one or two pulses per pattern.

The waveform output repeats for every pattern in the sequence step.

All waveforms can be output on any AUX I/O Channel. Waveform 1 and Waveform 3 can also be output on any channel.

Access this panel from the menu bar: **Edit>Data Sequencer>Waveforms.**

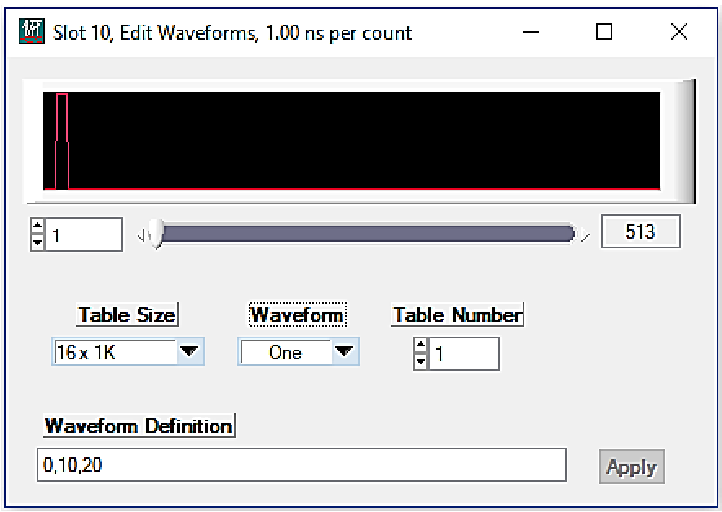


Figure 4-50 Edit Waveforms Panel Waveform 1

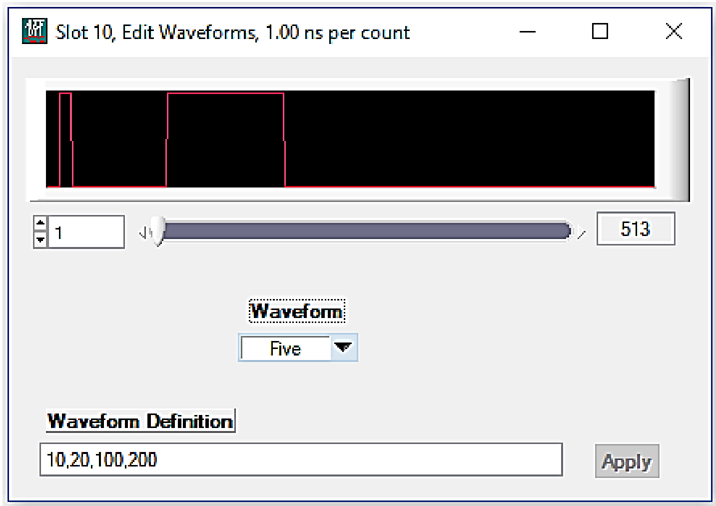


Figure 4-51 Edit Waveforms Panel Waveform 5

Table Size

This pull-down control programs the waveform table size for waveforms 1-4. Waveforms 5 and 6 are fixed at 65536.

Setting	Description
16 x 1K	16 tables each with 1024 bits
8 x 2K	8 tables each with 2048 bits
4 x 4K	4 tables each with 4096 bits
2 x 8K	2 tables each with 8192 bits
1 x 16K	1 table with 16384 bits

Table 4-59 Waveform Table Size Settings

Relevant vi(s):

Set Waveform Table Size

Waveform

This pull-down control selects the waveform to view/edit from one to six.

Table Number

This control selects the table number to view/edit. Waveforms five and six only have one table.

Waveform Definition

This control allows the user to define the waveform.

Specifying the beginning level and the bit number of subsequent transitions defines the waveform.

Example 1:

0,5,10,15

Beginning Level = 0;

3 Transitions at 5, 10, 15;

Would generate the following waveform;

"00000111110000011111111..."

Bits 1-5 low

Bits 6-10 high

Bits 11-15 low

Bits 16 through the size of the table high.

Example 2:

1

Beginning Level = 1;

No transitions;

Would generate the following waveform;

"111..."

Bits 1 through the size of the table high.

Waveform five and six have a maximum of two transitions.

Relevant vi(s):

Set Waveform Data

Editing Sequence Parameters

The sequence parameters consist of the following entries:

1. Loop Counter Mode
2. Pipeline Mask
3. Strobe/Vector Bit/Table Selection
4. Channel Test

Access this panel from the menu bar: **Edit>Data Sequencer>Sequence Parameters**.

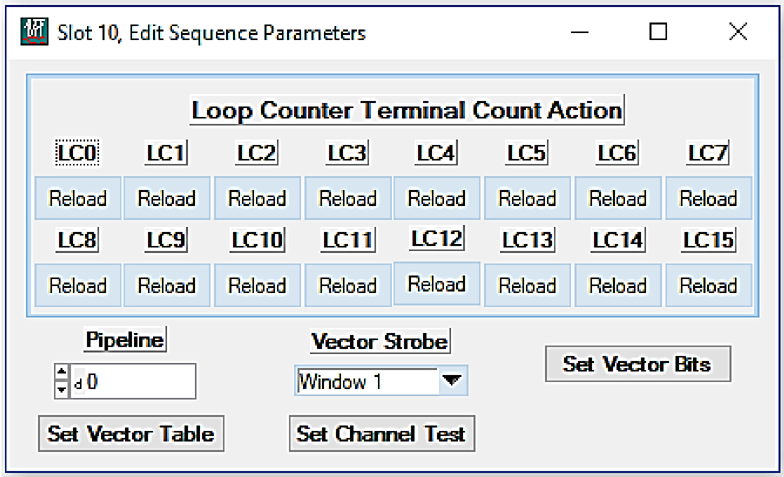


Figure 4-52 Data Sequencer Parameters Panel

LC0 – LC15

These controls program the loop counter mode.

There are sixteen 16-bit loop counters. Each of the sixteen loop counters can be programmed to either reload its count or disable when the terminal count is reached.

Given the following sample loop sequence:

Step 1 jump step 1 using LC0 count 2

Step 2 jump step 1 using LC1 count 3

Example 1:

If both loop counters reload on terminal count, then the step order will be:

1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2

Example 2:

If loop counter 0 is set to disable, then the step order will be:

1, 1, 1, 2, 1, 2, 1, 2, 1, 2

Relevant vi(s):

Set Sequence Loop Mode

Pipeline

This control programs the pipeline depth.

The pipeline may be from 0-16 Patterns deep. The “0” pipeline depth will hereafter be called a “zero pipeline depth”. A pipeline depth of “1-16” will hereafter be called a “non-zero pipeline depth”.

A non-zero pipeline depth offsets the PASS/FAIL result by the corresponding depth of the pipeline in patterns.

See the **Pipelining** section in Chapter 6 for a more in-depth explanation of how pipelining affects jumping, counting burst errors and the logging of errors in the error Address Memory.

Relevant vi(s):

Set Condition Pipeline

Vector Strobe

This control allows the user to set the vector strobe signal.

The closing edge of the selected window will sample the four vector bits VA0 (LSB) to VA3 (MSB). The vector bits are only used if the vector jump bit is set during a sequence jump step. The vector bits form an address into the vector table to determine the jump step and timing set (if timing mode set to indexed).

Setting	Description
Window 1	Sets the closing edge of window 1 as the vector strobe.
Window 2	Sets the closing edge of window 2 as the vector strobe.
Window 3	Sets the closing edge of window 3 as the vector strobe.
Window 4	Sets the closing edge of window 4 as the vector strobe.

Table 4-60: Vector Strobe Settings

Relevant vi(s):

Set Vector Jump Strobe

Set Vector Bits

This command button displays the **Edit Vector Bits** panel so the vector bit signal selection can be programmed.

The four vector signals comprise an index into a vector jump table that specifies the jump address as well as the timing set (indexed timing mode only). The vector table/signals are only used if the vector jump bit is set during a sequence jump step.

Configuring the vector signals consists of the following:

1. Select the Source.
2. Program the Input Mode.

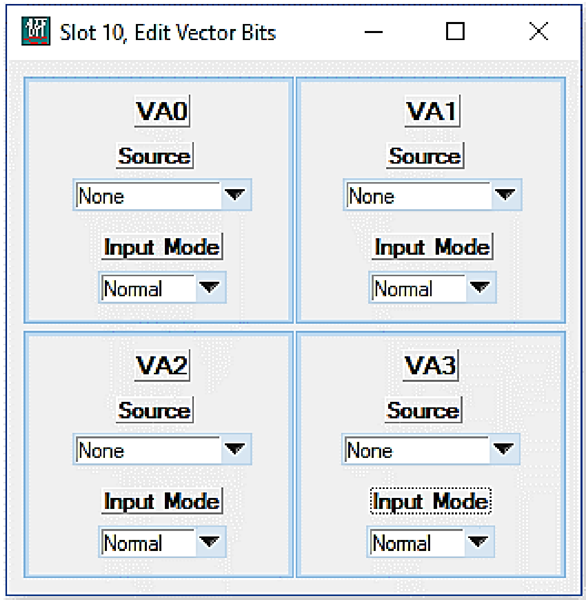


Figure 4-53 Edit Vector Bits Panel

Source

This pull-down control programs the vector bit source.

Setting	Description
None	No trigger source selected
AUX1-AUX12	Trigger source set to front panel signal
CHT1	Trigger source set to channel test 1
TTLTRG0-7	Trigger source set to PXI TTL trigger

Table 4-61 Vector Bit Source Settings

Relevant vi(s):

Set Vector Jump Signal

Input Mode

This pull-down control programs the trigger input mode for vector jumps.

Setting	Description
Normal	Do not modify input signal before testing.
Inverted	Invert input signal before testing.

Table 4-62 Vector Bit Input Mode Settings

Relevant vi(s):

Set Vector Jump Signal

Set Vector Table

This command button displays the Edit Vector Table panel so the vector table settings can be programmed.

The vector table is indexed by the four vector signals VA0 (LSB) to VA3 (MSB). Each vector table entry supplies the jump address as well as the timing set (indexed timing mode only). The vector table/signals are only used if the vector jump bit is set true in a sequence step.

Configuring the vector table signal consists of the following:

1. Select the Vector Bit Index
2. Select the Vector Jump Step
3. Program the Timing Set (only used in the indexed timing mode).

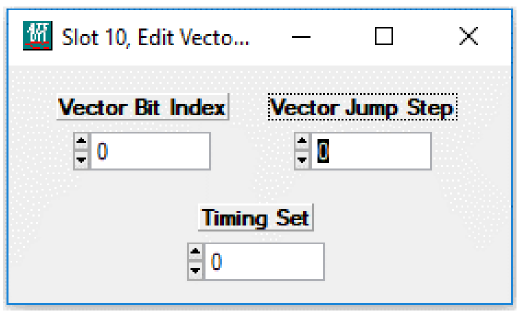


Figure 4-54 Edit Vector Table Panel

Vector Bit Index

This allows the user to enter the index to program. There are 16 indexes that can be set (0 to 15). The index is the binary value of the vector bits (VA0 through VA3).

Relevant vi(s):

Set Vector Jump Table

Vector Jump Step

This allows the user to enter the jump step number for the current vector jump index.

Relevant vi(s):

Set Vector Jump Table

Timing Set

When the timing mode is set to indexed, this control allows the user to specify the timing set for the current vector jump index.

Relevant vi(s):

Set Vector Jump Table

Set Channel Test

This command button displays the Edit Channel Test panel so the channel test settings can be programmed.

Configuring the sequence channel test registers consists of the following:

1. Program the expect value
2. Program the mask value

The expect value is compared to the response high (Good 1) of the input channel. A high in the mask, disables the comparison.

The result of all four channel test registers can be routed to the PXI TTL trigger bus. In addition channel test 1 result can also be routed to any of the sequence triggers.

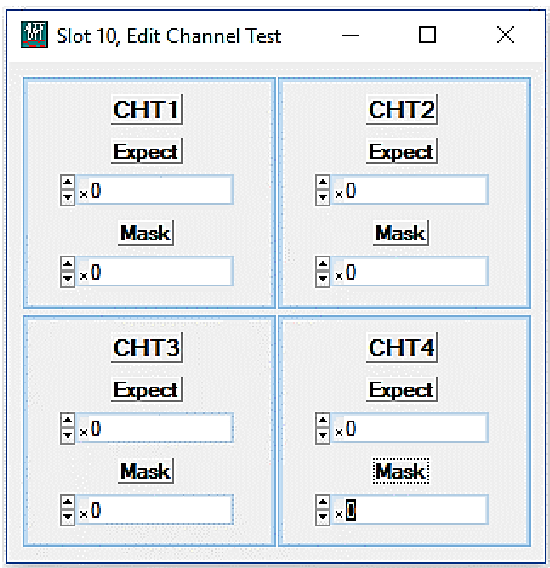


Figure 4-55 Sequencer Channel Test Panel

Expect

This command allows the user to enter the expect value for the channel test signal. Bit 0 of the expect value maps to the lowest channel and Bit 31 maps to the highest channel. A one represents a valid high test and a zero represents a valid low test.

Relevant vi(s):

Set Sequence Channel Test

Mask

This command allows the user to enter the mask value for the channel test signal. Bit 0 of the mask value maps to the lowest channel and Bit 31 maps to the highest channel. A one disables the comparison to the expect value and a zero enables the comparison.

Relevant vi(s):

Set Sequence Channel Test

PXI Backplane Trigger Bus section of Chapter 6 describes how to use Channel Tests to perform a logical OR and logical AND of two or more channels.

Editing Sequence Steps

The sequence steps are used to control the flow of the patterns and assign timing.

Access this panel from the menu bar: **Edit>Data Sequencer>Sequence Steps**.

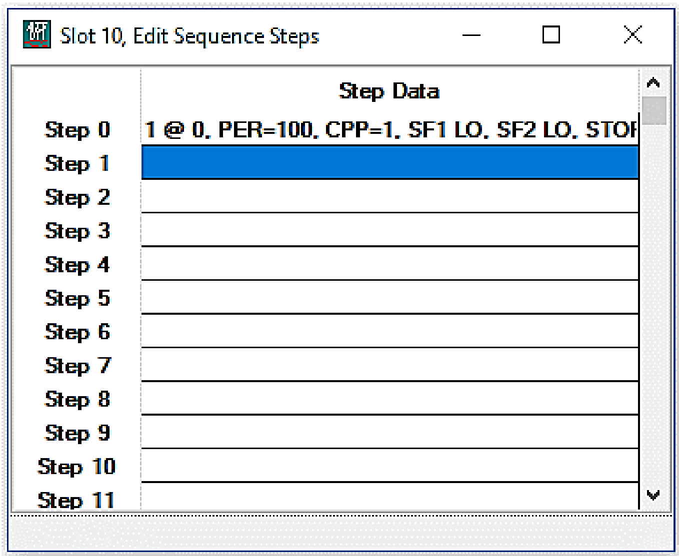


Figure 4-56 Edit Sequence Step Panel

Up to 4096 sequence steps are available for Indexed and Per Step Single timing modes. Up to 1024 sequence steps are available for “Per Step Multi” timing mode.

The Delete key will clear the step data contents, de-allocate any assigned pattern data and initialize the step settings.

A double-click on any of the step number cells opens a Sequence Step Data panel for that cell.

The **Sequencer Operation** section in Chapter 6 provides detailed information on sequencer operation.

Relevant vi(s):

Select Sequence Step

Initialize Sequence Steps

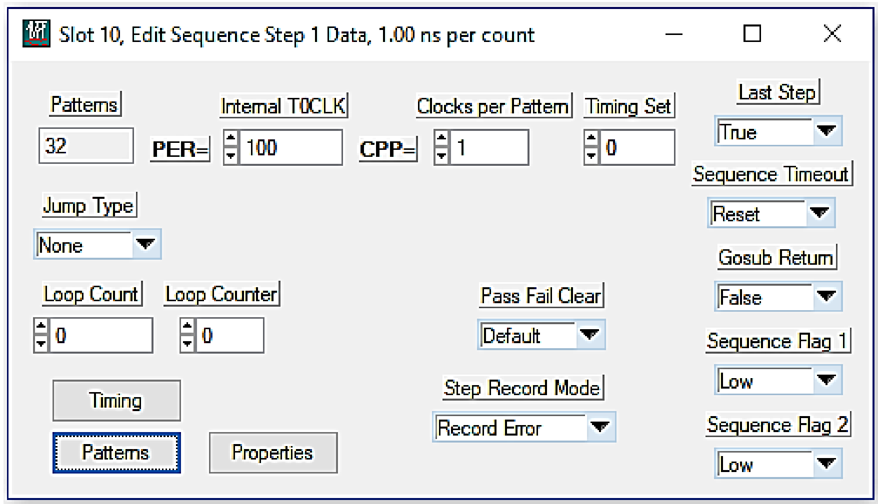


Figure 4-57 Sequence Step Data Panel

Internal T0CLK

This control allows the user to specify the Internal T0CLK period. The T0CLK period determines the pattern output period.

When the system clock source is set to internal T0CLK, this control specifies the system clock period. The period is programmed in master clock edges (rising and falling), i.e., 1/2 the master clock period.

For example, if the master clock is set to 500 MHz, then a setting of 20 would result in a system clock period of 20 ns.

20 * (1/2 (2 ns)) = 20 ns.

With a master clock of 100 MHz the system clock period would be 100ns.

20 * (1/2 (10 ns)) = 100 ns.

The valid values for T0CLK are from 20 to 65550.

Relevant vi(s):

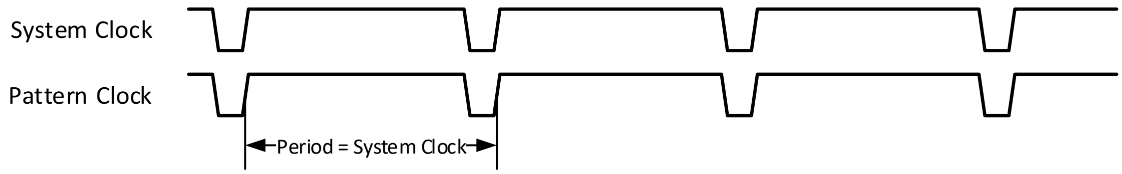
Set Sequence Clock

Clocks per Pattern

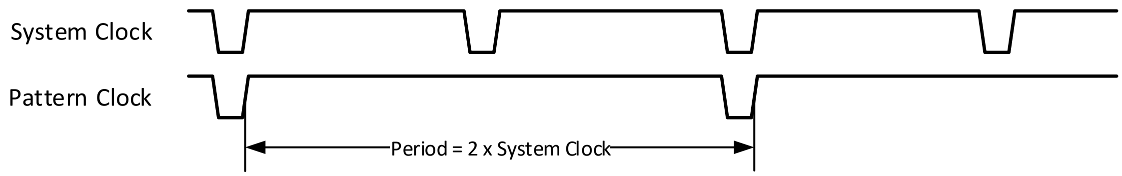
This numeric control defines the Clocks per Pattern (CPP) for each sequence step.

The CPP value determines the number of System Clocks that will be generated for each Pattern Clock. When CPP = 1, then Pattern Clock is equal to System Clock. When CPP = 2, then Pattern Clock is two times the System Clock.

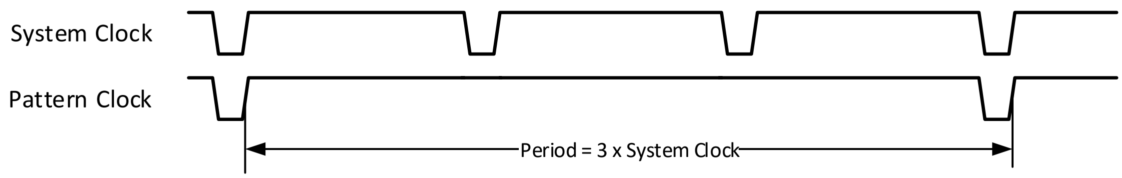
Example 1: CPP = 1



Example 2: CPP = 2



Example 3: CPP = 3



The valid values for CPP are from 1 to 256.

Relevant vi(s):

Set Sequence Clock

CPP Phase and Window Triggering

Two clocks are available for triggering the timing phases to begin their programmed definition; System Clock and Pattern Clock (see **Set Sequence Waveform**

Phase Trigger Properties in this chapter.)

If a Phase is defined to trigger on the System Clock then its span cannot exceed the System Clock period. If a Phase is triggered by the Pattern Clock, and the CPP >1, then that Phase can span the Pattern Clock period.

Windows are only triggered on the Pattern Clock and can span the Pattern Clock period while still observing the Timing Set Value Rules.

Timing Set

This numeric control sets the timing set number for the sequence step.

This control is only visible when the sequencer timing mode is set to indexed (see **Set Error Parameters**

Timing Mode in this chapter).

The valid values for control are from 0 to 255.

Relevant vi(s):

Set Sequence Timing Set

Last Step

This control specifies the Last Step flag. This flag indicates whether the current step is the last step of the sequence burst (True) or a sub-step of a multi-step burst (False).

Relevant vi(s):

Set Sequence Last Step

Sequence Timeout

This control specifies the Sequence Timeout mode. Every step in a multi-step burst can be timed using the sequence timeout timer. When the flag is set to **Reset**, the timer will re-start at the beginning of this step. If this flag is set to **Continue**, then the timer will not reset.

Relevant vi(s):

Set Sequence Timeout Continue

Gosub Return

This control specifies the Gosub Return flag. The Gosub Return flag is used to signal the last step of a subroutine.

Relevant vi(s):

Set Sequence Gosub Return

Sequence Flag 1 and Sequence Flag 2

This control specifies the level of Sequence Flag 1 and Sequence Flag 2 during this step. These general purpose outputs can be routed any of the AUX outputs as well as the PXI TTLTRG and ECLTRG outputs.

Relevant vi(s):

Set Sequence Flags

Jump Type

This pull-down control programs the Jump Type Mode.

Normal sequence step execution proceeds sequentially until the step with the "Last Step" flag is set true. Conditional and unconditional jumps and Gosubs can be added to allow the user to modify sequence step execution order.

Two jump types can be set, Normal and Gosub.

- Normal jumps force the next sequence step number to be replaced by the specified jump step number.

- Gosub jumps save the current step number and forces the next sequence step number to be replaced by the specified step number. The **Gosub Return** flag set true will force the sequence step number to be one more than the saved step number. For example, if step number 5 and 7 had a Gosub to step 10 and step 13 has the Gosub Return flag set, then the step number sequence starting from 1 would be,

  1, 2, 3, 4, 5, 10, 11, 12, 13, 6, 7, 10, 11, 12, 13, 8, 9 …

Setting	Description
None	Disable the jump logic for this step.
Normal	After executing this step's patterns, perform a normal jump if jump condition is true.
Gosub	After executing this step's patterns, perform a Gosub jump if the jump condition is true.

Table 5-81 Jump Type Settings

Relevant vi(s):

Set Sequence Jump

Jump Step

This numeric control programs the Jump Step number.

This control is only visible if the jump type is set to Normal or Gosub.

If the jump condition is true, then the next step number will be the value specified by the Jump Step instead of the next sequential step number.

The jump action takes precedence over the Last Step flag.

Relevant vi(s):

Set Sequence Jump

Jump Condition

This pull-down control programs the Jump Type Mode.

This control is only visible if the jump type is set to Normal or Gosub.

Jumps can be conditional or unconditional. Conditional jumps require a specified condition to be true in order for the jump to be enabled. Unconditional jumps are always enabled.

Setting	Description
Always	Jump always (Unconditional)
Step Not PASS	Jump if the PASS/FAIL flag is NOT a PASS (i.e. FAIL or Indeterminate)
Step Not FAIL	Jump if the PASS/FAIL flag is NOT a FAIL (i.e. PASS or Indeterminate)
Step FAIL	Jump if the PASS/FAIL flag is equal to FAIL
Step PASS	Jump if the PASS/FAIL flag is equal to PASS

Setting	Description
Sequence FAIL	Jump if Burst Error Count is not equal to zero.
Sequence PASS	Jump if Burst Error Count is equal to zero.
Jump Trigger 1 True	Jump if “Jump Trigger 1” true.
Jump Trigger 1 not True	Jump if “Jump Trigger 1” not true.
Jump Trigger 2 True	Jump if “Jump Trigger 2” true.
Jump Trigger 2 not True	Jump if “Jump Trigger 2” not true.
Jump Trigger 3 True	Jump if “Jump Trigger 3” true.
Jump Trigger 3 not True	Jump if “Jump Trigger 3” not true.
Jump Trigger 4 True	Jump if “Jump Trigger 4” true.
Jump Trigger 4 not True	Jump if “Jump Trigger 4” not true.

Table 4-63 Jump Condition Settings

The true/false state of the jump triggers is based on the jump trigger test condition. If the jump trigger test condition is set to “Low Level”, then “True” would indicate the jump trigger signal is low and “not True” would indicate the jump trigger signal is high.

Note: *Any CONDEN enabled FAIL during the Sequence Step will prevent a PASS.*

See the **Jumping** section of Chapter 6 for a detailed explanation of Jumping based on Errors.

Relevant vi(s):

Set Sequence Jump

Loop Count

This numeric control programs the Loop Count number.

Jumps can be qualified by a loop counter. The loop count can be set from 0 (no qualification) to 65536. A count qualified jump only allows the jump to occur a maximum of “count” times. This allows single or multiple steps to be looped.

Relevant vi(s):

Set Sequence Jump

Loop Counter

This numeric control programs the Loop Counter number.

Jumps can be qualified by a loop counter. Sixteen loop counters are available. Nested loops are supported including up to all 16 counters.

Relevant vi(s):

Set Sequence Jump

Vector Jump

This control specifies the Vector Jump flag. This flag indicates whether the vector jump mode is enabled (true) or disabled (false).

If the vector jump mode is enabled, then the sequence step number to jump to is specified in the vector jump table which is addressed by the Vector Bits which form the Vector Bit Index. If the vector jump mode is disabled, then the sequence step number to jump to is specified by the **Jump Step** control.

This control is only visible if the jump type is set to Normal or Gosub.

Relevant vi(s):

Set Sequence Jump

Pass Fail Clear

This control programs the Pass Fail Clear Mode during this step.

The pass fail flag is used for conditional jumping and indicates the results of a channel compare pattern code. The pass fail flag can be set to clear at the beginning of each sequence step (default) or to hold the previous state (mask).

Setting	Description
Default	Clear Pass Fail
Mask	Hold Previous Pass Fail

Table 4-64 Step Record Mode Settings

See the **Pass/Fail Flag Operation** section of Chapter 6 for a more in-depth explanation.

Relevant vi(s):

Set Sequence Pass Fail Clear

Step Record Mode

This control programs what data is recorded/logged during this step.

There are three memories that store data from a sequence burst:

1. Error Address Memory
2. Record Index Memory
3. Record Memory

There is also the Error Counter which counts the number of pattern errors that occurred during the previous sequence burst. The Error Count can be queried using the **Query Error Flags** vi.

The Error Address Memory stores the sequence step, address and index of each pattern that generated an error during the previous sequence burst. The Error Address Memory can be queried using the **Query Error Address** vi.

NOTE

If the error count basis is qualified, the Error Counter and the Error Address Memory only count/log errors that are enabled with BERREN.

The Record Index Memory contains the data required to align the record memory contents when data is stored sequentially (Record Type = Indexed) for the previous sequence burst.

The Record Memory contains either the error flag or response data for the previous sequence burst.

Setting	Description
None	Error counting and all three record memories are disabled.
Record Count	Error Counting enabled.
Record Error	Error counting and all three memories are enabled and the Record Memory is set to record error data.
Record Response	Error counting and all three memories are enabled and the Record Memory is set to record response data.

Table 4-65 Step Record Mode Settings

For the **Record Count** settings, the record memory can either be set to record all zeros (No Error) or disabled (see **Sequencer Record Mode** in this chapter).

See the **Recording Sequence Results** section of Chapter 6 for more details regarding the counting and recording of errors.

Relevant vi(s):

Set Sequence Record Mode

Timing

This command button displays the Edit Timing Set panel so the phase and window settings can be programmed for the selected sequencer step (see **Editing the Timing Sets** in this chapter).

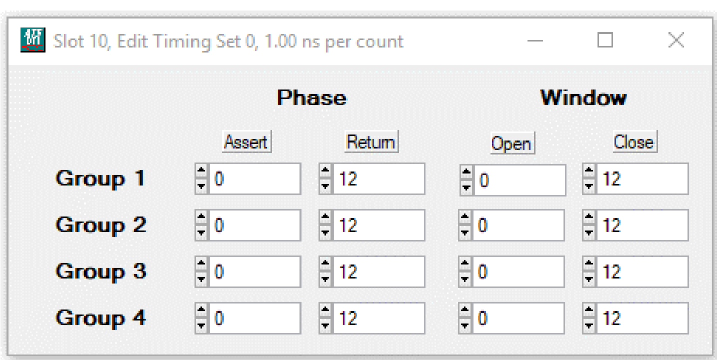


Figure 4-58 Edit Timing Set Panel

Relevant vi(s):

Set Sequence Timing Data

Patterns

If the **Patterns** control reads 0, then this command button displays the **Initialize Step Pattern Set** panel.

This panel allows the user to assign a block of pattern memory to the current sequence step.

The **Number of Patterns** control specifies how many patterns will be assigned and initialized to the current sequence step.

The **Memory Offset** control specifies the location of the first pattern. If the offset is set to -1, the driver automatically increments the offset to the next higher multiple of 4 from the previous offset. Any other number betw32een 0 and 262140, in multiples of 4, sets the offset.

Click **Apply** to initialize the patterns or **Close** to cancel.

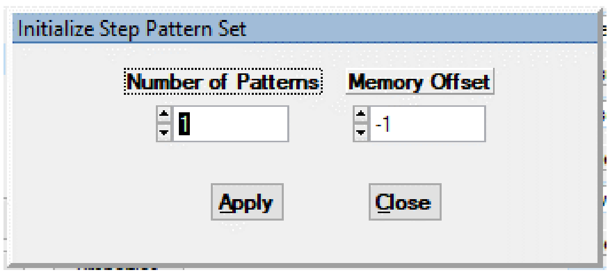


Figure 4-59 Initialize Step Pattern Set Panel

Relevant vi(s):

Initialize Pattern Set

If the **Patterns** control reads a number greater than zero, then this command button displays the Edit Pattern Data panel (see **Editing the Patterns** in this chapter).

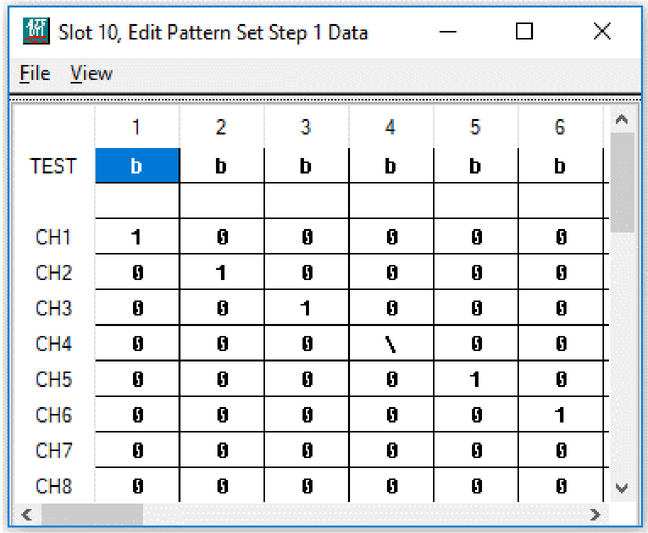


Figure 4-60 Edit Pattern Set Panel

Properties

This command button displays the Sequence Step Properties panel.

The sequence step properties consist of the following hardware settings:

1. Handshake Control (Pause/Resume)
2. Waveform
3. Phase Trigger

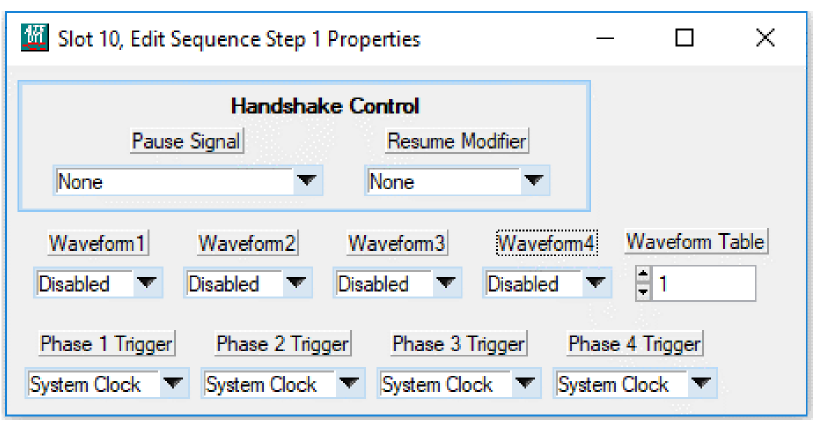


Figure 4-61 Sequence Step Properties Panel

Handshake Control

The handshake control assigns a signal (Pause) that can be either internal or external, which will pause the sequencer. When paused, the following will stop:

- Phases
- Windows

- Waveforms

For each pause signal selection, there is a corresponding signal that will continue (Resume) sequence operation. See the **Pause and Halt** section of Chapter 6 for additional details about the use of pause.

Pause Signal

This pull-down control programs the Handshake Pause signal.

Setting	Pause Signal	Resume Signal
None	Handshake mode disabled	NA
Pause Trigger 1 True	Pause Trigger 1 signal true	Pause Trigger 1 Resume
Pause Trigger 1 Not True	Pause Trigger 1 signal not true	Pause Trigger 1 Resume
Pause Trigger 2 True	Pause Trigger 2 signal true	Pause Trigger 2 Resume
Pause Trigger 2 Not True	Pause Trigger 2 signal not true	Pause Trigger 2 Resume
Phase 1 Assert	Phase 1 Assert edge occurs	Phase 1 Resume Trigger
Phase 1 Return	Phase 1 Return edge occurs	Phase 1 Resume Trigger
Phase 2 Assert	Phase 2 Assert edge occurs	Phase 2 Resume Trigger
Phase 2 Return	Phase 2 Return edge occurs	Phase 2 Resume Trigger
Phase 3 Assert	Phase 3 Assert edge occurs	Phase 3 Resume Trigger
Phase 3 Return	Phase 3 Return edge occurs	Phase 3 Resume Trigger
Phase 4 Assert	Phase 4 Assert edge occurs	Phase 4 Resume Trigger
Phase 4 Return	Phase 4 Return edge occurs	Phase 4 Resume Trigger

Table 4-66 Handshake Pause Signal

The true/false state of the pause triggers is based on the pause trigger test condition. If the pause trigger test condition is set to “Low Level”, then true would indicate the pause trigger signal is low and false would indicate the pause trigger signal is high.

The Resume Signal selection is covered in the **Configure Triggers** section in this chapter.

Relevant vi(s):

Set Sequence Handshake

Resume Modifier

This pull-down control programs the Handshake Resume Modifier.

The resume modifier allows the handshake to resume normally (None) or allows for the following modifications:

- Pattern Delay 1 or 2: Continue either on the presence of the specified resume signal or at the exhaustion of Pattern Delay timer 1 or 2 (the Delay Timer started when the Pause signal was received).
- Pattern Timeout: Set the pattern timeout (PTO) flag if the specified resume signal is not received by the time the Pattern Timeout timer has exhausted (the Pattern Timeout timer starts when the Pause is initiated).

Setting	Resume Modifier
None	No modifier, resume on ‘Resume Signal” only
Pattern Delay 1	Pattern Delay 1 timer
Pattern Delay 2	Pattern Delay 2 timer
Pattern Timeout	Pattern Timeout timer (PTO also set)

Table 4-67 Handshake Modifier Settings

Relevant vi(s):

Set Sequence Handshake

Waveform Properties

The waveform logic allows the user to enable up to six waveforms per sequence step (see **Editing Waveforms** in this chapter). Waveform 1 through Waveform 4 have to be enabled per sequence step to replace the timing signals they are paired with. Waveforms 5 and 6 are dedicated and do not need to be enabled.

Waveform1 – Waveform4

This control allows the user to enable/disable the specific waveform number.

Relevant vi(s):

Set Sequence Waveform

Waveform Table

This numeric control allows the user to program the waveform table for the sequence step. Numeric values can range from Waveform Tables 1 through 16.

Relevant vi(s):

Set Sequence Waveform

Phase Trigger Properties

The phase trigger logic allows the user to select the phase trigger signal source for the four phases between the “System Clock” and the “Pattern Clock (PCLK)”. In “System Clock” mode, another Phase is output for each System Clock. In “Pattern Clock” mode, another Phase is output for each Pattern Clock, which results in the Phase output rate being at a multiple of the System Clock period if CPP>1 ($PER_{PCLK} = PER_{SCLK} * CPP$).

Relevant vi(s):

Set Sequence Phase Trigger

Execute the Sequence

Sequence execution and control is performed from the Execute panel.

Access this panel from the menu bar: **Execute>Data Sequencer**.

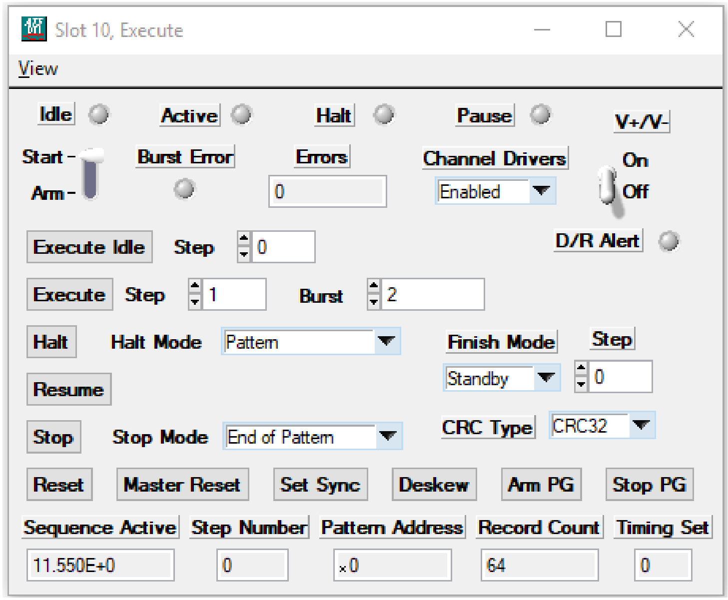


Figure 4-62 Executing a Sequence Panel

The following sections describe the execution overview as well as the indicators and controls of the execute panel.

Execution Overview

The sequencer execution state diagram is illustrated in the following figure.

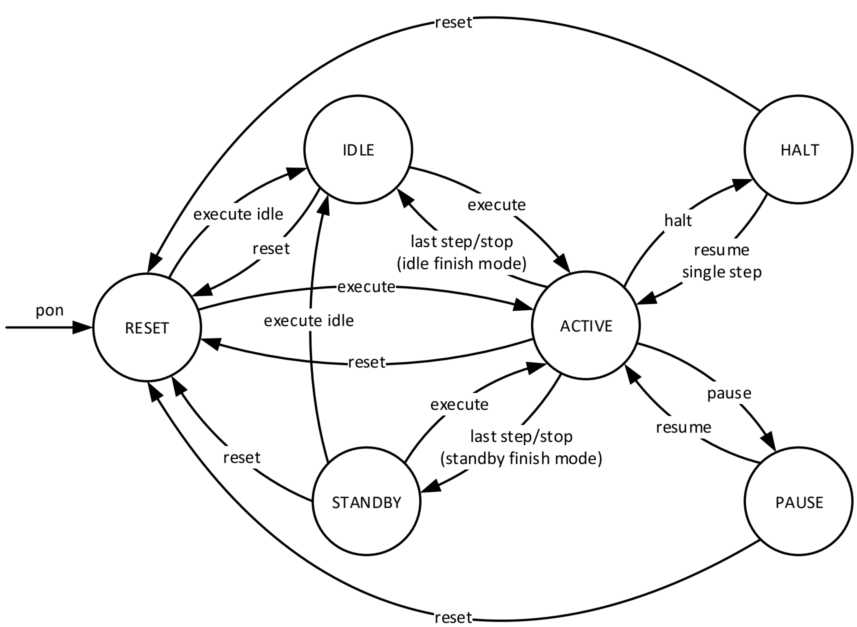


Figure 4-63 Execute State Diagram

The following table describes the six execute states of the DRM and how the state is entered.

Setting	Description	Entry Condition
RESET	Idle Active: false Sequence Active: false Halt flag: false Paused flag: false Active step: 0 Pattern Memory: Free	"pon", "reset"
STANDBY	Idle Active: false Sequence Active: false Halt flag: false Paused flag: false Active step: User Pattern Memory: Free	"last step/stop (standby finish mode)"
IDLE	Idle Active: true Sequence Active: false Halt flag: false Paused flag: false Active step: User Pattern Memory: Busy	"execute idle", "last step/stop idle finish mode"
ACTIVE	Idle Active: false Sequence Active: true Halt flag: false Paused flag: false Active step: User Pattern Memory: Busy	"execute", "resume"
HALT	Idle Active: false Sequence Active: true Halt flag: true Paused flag: false Active step: User Pattern Memory: Free	"halt"
PAUSE	Idle Active: false Sequence Active: true Halt flag: false Paused flag: true Active step: User Pattern Memory: Busy	"pause"

Table 4-68 Execute State Description

The following table describes the state transitions and the execute panel control to perform it.

Transition	Description	Soft Front Panel Control
pon	Power on	NA
reset	Sequencer reset	Depress **Reset** command button. Depress **Master Reset** command button (also disables output drivers).

Transition	Description	Soft Front Panel Control
execute idle	Execute idle sequence	Enter step number and depress **Execute Idle** command button.
execute	Execute sequence	Enter step number and depress **Execute** command button.
last step/stop (idle finish mode)	Sequence completes step with last step flag true or stop command. Finish Mode set to Idle	Set **Finish Mode** to "Idle" Enter step number and depress **Execute** command button. If sequence is still active, depress the **Stop** command button.
last step/stop (standby finish mode)	Sequence completes step with last step flag true or stop command. Finish Mode set to Standby	Set **Finish Mode** to "Standby" Enter step number and depress **Execute** command button. If sequence is still active, depress the **Stop** command button
halt	Halt the active sequence.	Make sure the **Halt Mode** is not set to "Disabled" Depress the **Halt** command button. If sequence was active, **Halt LED** should be red (halted). If sequence was not running, **Halt LED** should be green (armed).
resume/single step	Halt resume or single step	While in HALT state: Depress **Resume** command button to resume. Depress **Halt** command button to single step.
pause	Pause the primary sequence	No control to manually pause the primary sequence.
resume	Pause resume	While in PAUSE state: Depress **Resume** command button to resume.

Table 4-69 Execute State Transition Description

Execute Panel Indicators

There are twelve indicators that display the current sequencer status. These indicators are updated every 50 ms.

Idle LED

When green, indicates that the sequencer is in the IDLE state.

Relevant vi(s):

Query Sequencer Status

Active LED

When green, indicates that the sequencer is in the ACTIVE state.

Relevant vi(s):

Set Sequence Waveform

Halt LED

When green, indicates that the halt mode has been armed. When red, indicates that the sequencer is in the HALT state.

Relevant vi(s):

Set Sequence Waveform

Pause LED

When green, indicates that the sequencer is in the PAUSE state.

Relevant vi(s):

Set Sequence Waveform

Burst Error LED

When red, indicates that one or more burst errors have occurred in the previous sequence run.

Relevant vi(s):

Query Error Flags

Errors

This numeric indicator displays the number of pattern errors from the previous sequence burst.

Relevant vi(s):

Query Error Flags

D/R Alert

Illuminated red indicates that one or more bits are set in the Driver/Receiver event register.

Relevant vi(s):

Query FE Condition

Sequence Active

This numeric indicator displays the execution time of the previous sequence burst (10 ns resolution ± 10 ns with an accuracy of 500 ppm up to~43 sec).

Relevant vi(s):

Query Sequence Active

Step Number

This numeric indicator displays the current sequence step address.

Relevant vi(s):

Query Sequencer Status

Pattern Address

This numeric indicator displays the current pattern address.

Relevant vi(s):

Query Sequencer Status

Record Count

This numeric indicator displays the current record count.

Relevant vi(s):

Query Record Count

Timing Set

This numeric indicator displays the current timing set index (only visible in indexed timing mode).

Relevant vi(s):

Query Sequencer Timing Set

Execute Panel Modes and Settings

There are eleven controls that set the execution mode settings.

Start/Arm Selector

This slide selects whether the **Execute Idle** or **Execute** command buttons arm or start the specified action (See **Execute Idle** and **Execute** command button descriptions).

Channel Drivers

This pull-down control programs the channel drivers.

Setting	Description
Disabled	All the channel drivers are forced off (disabled).
Enabled	Level and state are determined by pattern code and channel parameters and properties.

Table 4-70 Channel Drivers Settings

Note: *The following events can cause force the drivers to be disabled:*

- A Watch Dog Timeout, if enabled to do so

- A local or DTS global drive fault event, if enabled to do so
- A channel over-voltage event

Relevant vi(s):

Set Driver Enable

V+/ V-

This control allows power to be applied to the driver/receiver circuits. It also enables the isolation relays to be closed.

Note: *The following Driver/Receiver Event will automatically force the V+ and V- power switch off (or not allow it to be turned on) protecting the module pin drivers.*

- Temperature Fault detected
- OVP detect (DR3e, DR9 and UR14)

Relevant vi(s):

Set Power Connect

Execute Idle Step

This control sets the idle step number for the **Idle** command button operation.

Relevant vi(s):

Set Idle Sequence

Execute Step

This control sets the step number for the **Execute** command button operation.

Relevant vi(s):

Execute Sequence

Arm Sequence

Burst

This control sets the burst count for the **Execute** command button operation. The burst count determines how many times the sequence will be looped. A count of 0 causes continuous looping. Maximum burst count is 1048576.

Relevant vi(s):

Set Burst Count

Halt Mode

This pull-down control programs the halt mode. The halt mode determines where execution will halt following either a manual halt (**Halt** command button) or an external halt trigger.

See the **Halt Operation** section in Chapter 6 for additional details about the use of halt.

Setting	Description
Disable	Halt signal ignored.
Pattern	Halt the current sequence at the end of the next pattern.
Step	Halt the current sequence at the end of the next step.
Sequence	Halt the current sequence at the end of the next sequence loop.
Sync 1	Halt the current sequence at the end of the next pattern according to where the Sync Pulse 1 is positioned.
Sync 2	Halt the current sequence at the end of the next pattern according to where the Sync Pulse 2 is positioned.
Pattern Fail	Halt the current sequence at the end of the next pattern if the pass/fail flag is set to fail.
Step Fail	Halt the current sequence at the end of the next sequence step if the pass/fail flag is set to fail.
Sequence Fail	Halt the current sequence at the end of the next sequence if the pass/fail flag is set to fail.
Pattern Pass	Halt the current sequence at the end of the next pattern if the pass/fail flag is set to pass.
Step Pass	Halt the current sequence at the end of the next sequence step if the pass/fail flag is set to pass.
Sequence Pass	Halt the current sequence at the end of the next sequence if the pass/fail flag is set to pass.

Table 4-71 Halt Mode Settings

Relevant vi(s):

Set Halt Mode

Finish Mode

This pull-down control programs the finish mode. When a sequence execution completes, the sequencer will enter either the **Standby** or **Idle** state. The **Standby** state outputs the first pattern of the specified step and pattern memory can be accessed by the user while the sequencer is in Standby. The **Idle** state outputs the entire pattern set of the specified step and pattern memory cannot be accessed while the sequencer is idling.

Setting	Description
Standby	Go to **Standby** after sequence completes.
Idle	Go to **Idle** after sequence completes.

Table 4-72 Finish Mode Settings

Relevant vi(s):

Set Finish Sequence

Finish Mode Step

This control sets the finish mode step number.

Relevant vi(s):

Set Finish Sequence

Stop Mode

This pull-down control programs the stop mode. The stop mode controls what action a CPU generated stop or a triggered stop will perform if received.

Setting	Description
Disable	The stop signal will be ignored.
End of Pattern	The stop signal causes the current sequence burst to terminate at the end of the next pattern.
Looping	The stop signal causes the next jump to be ignored. Sequence execution resumes at the step sequentially following the step with the ignored jump.
End of Sequence	The stop signal causes the current sequence burst to terminate at the end of the sequence of a continuous or looped burst.

Table 4-73 Stop Mode Settings

Relevant vi(s):

Set Stop Mode

CRC Type

This pull-down control programs the CRC type for the next burst.

Setting	Description
CRC16	Set the CRC algorithm to hex 8940. CRC results will be stored in the 16 bit CRC memory.
CRC32	Set the CRC algorithm to hex 82608EDB. CRC results will be stored in the 32 bit CRC memory.
Custom	CRC algorithm set by user. CRC results will be stored in 32 bit CRC memory.

Table 4-74 CRC Type Settings

Relevant vi(s):

Set CRC Type

Set Sync

This command button displays the **Set Sync** panel so that Sync 1 and Sync 2 signals can be programmed to generate a pulse.

These two sync outputs can be routed to any of the AUX or TTLTRG outputs. The sync parameters consist of an offset and a length. Once the programmed sync event occurs, the sync pulse will begin after the "offset" and last for "length". Both "offset" and "length" are specified in pattern clocks. The sync pulse will not extend past the end of the sequence. In the "Step" event, the sync pulse will not extend beyond the specified step.

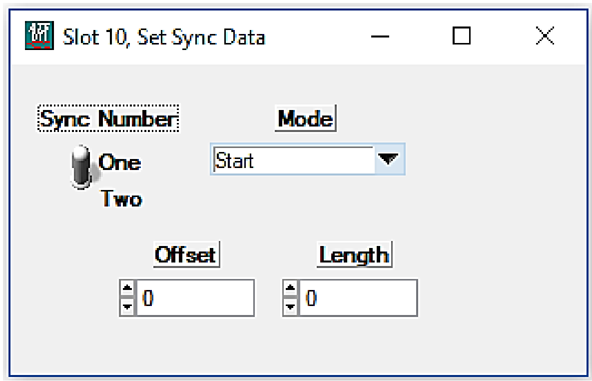


Figure 4-64 Set Sync Panel

Sync Number

This control selects which sync pulse signal to program, either Sync 1 or Sync 2.

Mode

This pull-down control programs the sync mode. The sync pulse event can be set to either the start of a sequence or a specific step.

Setting	Description
Start	The sync pulse begins from the start of the sequence.
Single Step	A sync pulse will be generated the first time the specified sequence step is executed.
Continuous Step	A sync pulse will be generated every time the specified sequence step is executed.

Table 4-75 Sync Mode Settings

Relevant vi(s):

Set Sync Event

Step

This control (only visible if the mode is set to single or continuous step) specifies the step number for the sync pulse and can be set from 0 to 4095.

Relevant vi(s):

Set Sync Event

Offset

This control sets the offset from the sync event before the sync pulse starts. The offset can be set from 0 to 1048575 patterns.

Relevant vi(s):

Set Sync Parameters

Length

This control sets the length for the sync pulse from 0 (no pulse) to 4095 patterns.

Relevant vi(s):

Set Sync Parameters

Execute Panel Command Buttons

There are ten command buttons that control DTI sequence, deskew and pulse generator execution.

Execute Idle

If the **Start/Arm** control is set to Start, this command button starts the Idle sequence at the sequence step specified in the **Execute Idle Step** control. If the **Start/Arm** control is set to Arm, this command button arms the Idle sequence at the sequence step specified in the **Execute Idle Step** control. Arming the idle sequence would be used in conjunction with an external start trigger. It is also used if this is not the Primary Sequencer in a DTS.

Relevant vi(s):

Execute Idle Sequence

Arm Idle Sequence

Execute

If the **Start/Arm** control is set to Start, this command button starts the sequence at the sequence step specified in the **Execute Step** control. If the **Start/Arm** control is set to Arm, this command button arms the sequence at the step specified in the **Execute Step** control. Arming the sequence would be used in conjunction with an external start trigger. It is also used if this is not the Primary Sequencer in a DTS.

Relevant vi(s):

Execute Sequence

Arm Sequence

Halt

The **Halt** command button halts the sequence based on the **Halt Mode** selection.

Once halted (indicated by a red **Halt LED**), another push of the **Halt** command button resumes the sequence and then halts it again (single step).

See the **Halt Operation** section in Chapter 6 for additional details about the use of halt.

Relevant vi(s):

Halt Sequence

Resume

The **Resume** command button terminates a pause or halt state and sequence execution continues.

See the **Pause and Halt** section in Chapter 6 for additional details about resuming a pause or halt.

Relevant vi(s):

Resume Sequence

Stop

The **Stop** command button stops the sequence based on the **Stop Mode** selection. The standby or idle state will become active based on the **Finish Mode** setting. Pressing the **Stop** command button when the sequence is not active latches the stop command until the sequence is active.

Relevant vi(s):

Stop Sequence

Reset

The **Reset** command button forces the sequence to the reset state (Sequence Step 0) with the **Channel Drivers** setting unchanged.

Relevant vi(s):

Reset Sequence

Master Reset

The **Master Reset** command button forces the sequence to the reset state (Sequence Step 0) and also sets the **Channel Drivers** to Disabled.

Relevant vi(s):

Master Reset Sequence

Deskew

The **Deskew** command button activates the end-of-cable deskew procedure. Only closed channels will be deskewed.

Relevant vi(s):

Deskew DTI Channels

Arm PG

The **Arm PG** command button arms the pulse generator.

Note: *The Pulse Generator will not work in any of its modes until armed.*

Relevant vi(s):

Arm Pulse Generator

Stop PG

The **Stop PG** command button stops the pulse generator.

Relevant vi(s):

Stop Pulse Generator

PMU Operation

The PMU display is accessed from the **Execute>PMU** menu bar selection.

The PMU panel controls available are based on the selected channel function.

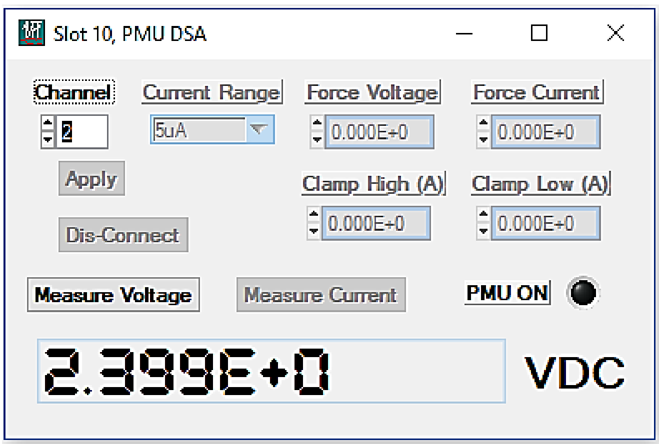


Figure 4-65 PMU Panel Dynamic Channel

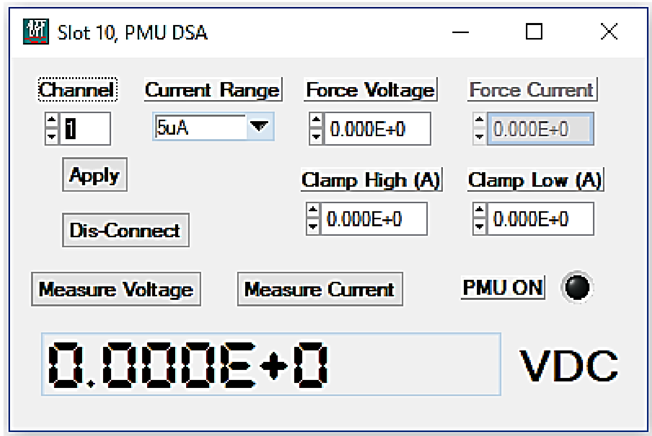


Figure 4-66 PMU Panel PMU FV Channel

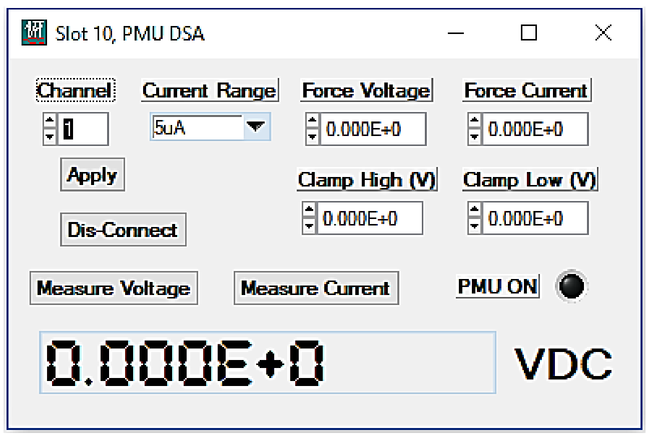


Figure 4-67 PMU Panel PMU FI Channel

Channel

This control sets the channel number to program or query.

Current Range

This pull-down control sets the specified channels current range. The control is only valid for channels whose function has been set to PMU FV or PMU FI.

Setting	Description
5uA	The current range set to +/- 5µA.
50uA	The current range set to +/- 50µA.
500uA	The current range set to +/- 500µA.
5mA	The current range set to +/- 5mA.
50mA	The current range set to +/- 50mA.

Table 4-76 Current Range Settings

If the channel function is set to PMU FV, then the current range sets the resolution and min/max levels for the **Clamp High (A)** and **Clamp Low (A)** controls.

If the channel function is set to PMU FI, then the current range sets the resolution and min/max levels for the **Force Current** control.

If the PMU is on, then changing the current range will disconnect the PMU, apply the settings and reconnect the PMU.

The Relevant vi(s):

Set Force Voltage

Set Force Current

Force Voltage

This control specifies the force voltage level for the specified channel. The control is only valid for channels whose function has been set to PMU FV or PMU FI.

The force voltage range is from -2.0 to +7.0.

If the channel is set to PMU FV, then the output will be programmed.

If the channel is set to PMU FI, then the voltage clamps will be set to this voltage plus and minus 100mV when the PMU is connected before setting the actual voltage clamp levels.

The Relevant vi(s):

Set Force Voltage

Force Current

This control specifies the force current level for the specified channel. The control is only valid for channels whose function has been set to PMU FI.

The recommended range is from ± Imax where Imax is the programmed current range. The usable range is ± (2 * Imax).

The Relevant vi(s):

Apply Force Current

Set Force Current

Clamp High/Clamp Low

These controls specify the voltage or current clamp levels. These controls are only valid for channels whose function has been set to PMU FV or PMU FI.

Current Clamps

The PMU will force a voltage as long as the current flow at the channel pin is between the high and low current clamp values.

The recommended range is from ± Imax where Imax is the programmed current range. The usable range is ± (2 * Imax).

The Relevant vi(s):

Apply Force Voltage

Set Current Clamps

Voltage Clamps

The PMU will force a current as long as the voltage at the channel pin is between the high and low voltage clamp values.

The recommended range is from -2V to +7V.

The Relevant vi(s):

Apply Force Current

Set Voltage Clamps

Apply

This control applies the force voltage/force current parameters and then connects the PMU of the specified channel. The control is only valid for channels whose function has been set to PMU FV or PMU FI.

The Relevant vi(s):

Apply Force Voltage

Apply Force Current

Dis-connect

This control dis-connects the PMU of the specified channel. The control is only valid for channels whose function has been set to PMU FV or PMU FI.

The Relevant vi(s):

Disconnect PMU

PMU ON

This status LED indicates the status of the PMU for the specified channel.

- On – PMU is on and connected
- Off – PMU is off and dis-connected

The Relevant vi(s):

Query PMU Connect

Measure Voltage

This control initiates a voltage measurement and displays the result.

Relevant vi(s):

Measure Voltage

Measure Current

This control initiates a current measurement and displays the result.

Relevant vi(s):

Measure Current

Counter/Timer Operation

The / Counter/Timer Panel is accessed from the **Execute>Counter/Timer** menu bar selection.

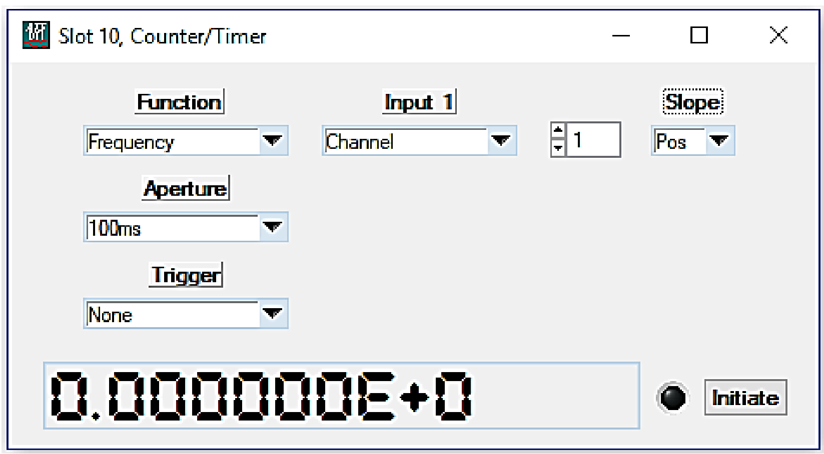


Figure 4-68 Timer/Counter Panel

Function

This pull-down control programs the counter/timer function.

Setting	Description
Frequency	**Initiate** command button performs a frequency measurement on input 1.
Period	**Initiate** command button performs a period measurement on input 1.
Time Interval	**Initiate** command button performs a time interval measurement from input 1 to input 2.
Totalize	**Initiate** command button counts the input 1 transitions during input 3.
Timed Totalize	**Initiate** command button counts input 1 transitions during the specified aperture time.
Positive Pulse	**Initiate** command button performs a time interval measurement from the rising edge input 1 to the falling edge of input 1.

Setting	Description
Negative Pulse	**Initiate** command button performs a time interval measurement from the falling edge input 1 to the rising edge of input 1.

Table 4-77 Counter/Timer Function Settings

Relevant vi(s):

Set Counter Function

Input <1-3> Source

These controls select the counter input source.

Source	Description
Channel	Channel 1 through 32
AUX	AUX 1 through 12
Freq. Synth.	Frequency Synthesizer
CLK10	10 MHz backplane clock.
250 MHz	500 MHz clock divided by 2.
Pulse Generator	Pulse Generator.

Table 4-78 Counter/Timer Input <1-3> Source

Relevant vi(s):

Set Counter Input

Input <1-3> Slope

These controls select the counter input slope.

Source	Description
Pos	Select rising edge.
Neg	Select falling edge.

Table 4-79 Counter/Timer Input <1-3> Slope

Relevant vi(s):

Set Counter Input

Aperture

This control sets the gate aperture time for the frequency, period and timed totalize functions.

Setting	Description
1us	One microsecond gate time.
10us	Ten microsecond gate time.

Setting	Description
100us	One hundred microsecond gate time.
1ms	One millisecond gate time.
10ms	Ten millisecond gate time.
100ms	One hundred millisecond gate time.
1s	One second gate time.
10s	Ten second gate time.

Table 4-80 Counter/Timer Aperture

Relevant vi(s):

Set Counter Aperture

Trigger

This pull-down control programs the trigger source.

Source	Description
None	Disables the timer/counter.
External	Sets input 3 as the trigger source.
Internal Continuous	Enables Continuous Measurements.
Internal Single	Performs one measurement with initiate.

Table 4-81 Timer/Counter Trigger Source

Relevant vi(s):

Set Counter Trigger

Initiate

Generates an immediate trigger to the timer/counter.

Relevant vi(s):

Counter Initiate Trigger

Results

Retrieve the results of the selected counter/timer function.

Relevant vi(s):

Measure Counter Result

Analyze the Execution Results

After sequence execution has been performed, the final step is to analyze the results to determine if the recorded input data is valid and if it matches the expected results.

The **Burst Error LED** and **Errors** are result indicators located on the execution panel. Additional result data can be accessed from the **Execute>Data Sequencer** menu bar, **View** selection.

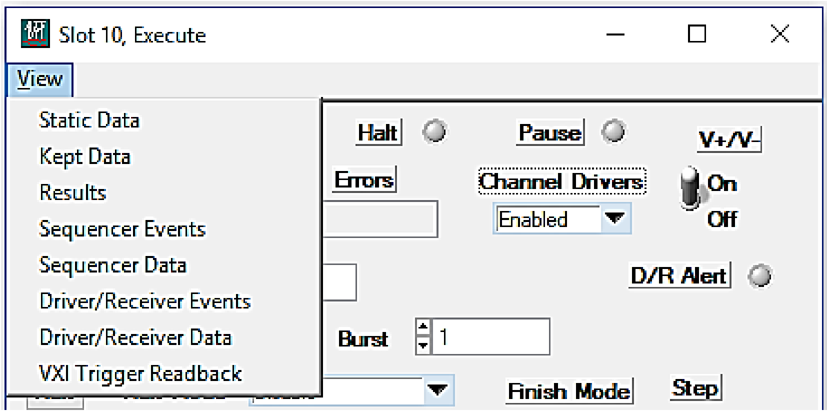


Figure 4-69 Execute View Menu

These panels query the recorded memory results and status indicators from the previous sequence execution.

Static Data

The static data display is accessed from the **View>Static Data** menu bar selection.

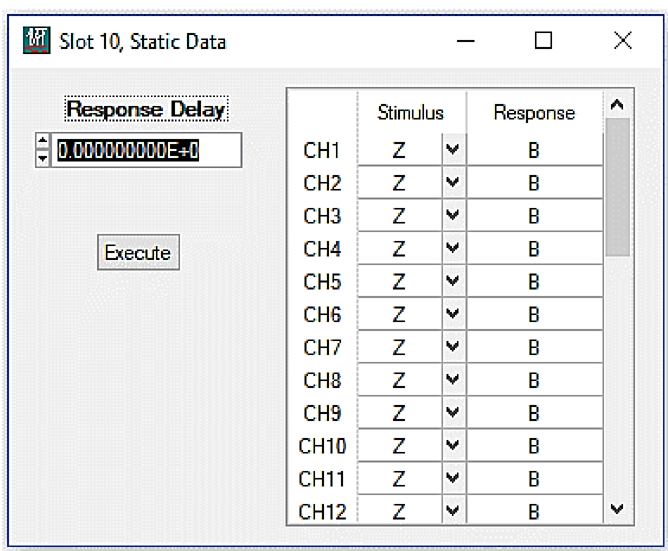


Figure 4-70 Static Data Panel

The static data panel contains controls that program the static timing and stimulus data and displays the current static response data.

Response Delay

This control sets the delay when the static input pins will be sampled from 0 to 6.5ms with 100ns resolution. The delay is from the execution of the Execute Static Pattern vi.

Relevant vi(s):

Set Static Timing

Execute Static Pattern

Stimulus

This table column contains pull down selections that sets the stimulus output state.

Setting	Description
Z	Disable the channel.
0	Drive to low level.
1	Drive to high level.
X	Uninstalled channel

Table 4-82 Static Stimulus Settings

Relevant vi(s):

Set Static Data

Response

This table column contains the stimulus input state of the previous static execution.

Code	Description
B	Response between high and low.
L	Response low level.
H	Response high level.
?	Unknown

Table 4-83 Static Stimulus Settings

Relevant vi(s):

Query Static Response

Kept Data

The kept data display is accessed from the **View>Kept Data** menu bar selection.

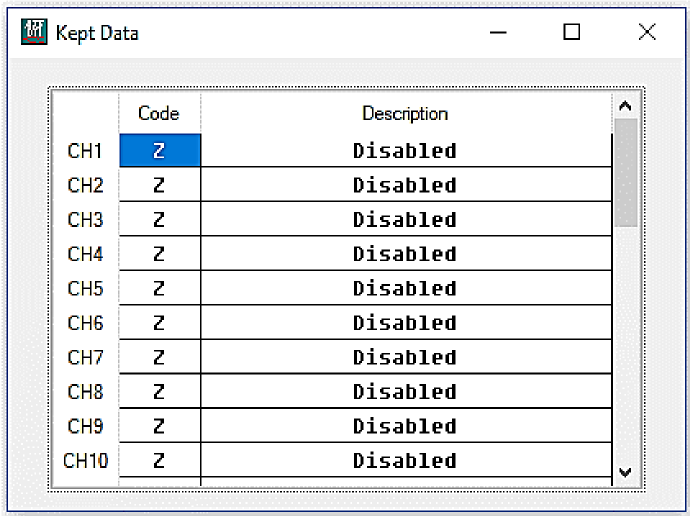


Figure 4-71 Kept Data Panel

The kept data represents the current pattern code that is not “Invert Previous Code” or “Repeat Previous Code”.

NOTE

The Kept Data is updated at the end of a pattern so the contents of the kept data when halted or paused will contain the codes from the previous pattern.

Relevant vi(s):

Query Kept Pattern

Results

The Results data display is accessed from the **View>Results** menu bar selection.

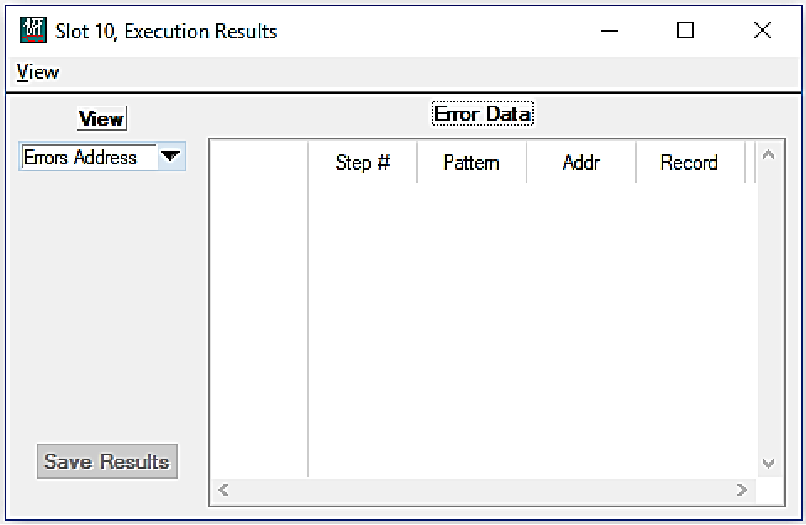


Figure 4-72 View Results Data Panel

View

This pull-down control selects the results to view.

Setting	Description
CRCs	Display the CRC data from the previous sequence execution.
Error Address	Display the error address data from the previous sequence execution.
Record Index	Display the error address data from the previous sequence execution.
Record Data	Display the error address data from the previous sequence execution.

Table 4-84 Results View Settings

Save Results

This command button displays a file save panel that allows the user to select and existing file or create a file to store the result data as a comma separated list (.csv). All numeric values are displayed as decimal.

CRC Save File Format

The CRC results are saved in the following format:

<id>,<crc><lf>

Where:

<id>	CH01 through CH32, PG0 and PG1.
<crc>	The CRC value.

Error Address Save File Format

The Error Address results are saved in the following format:

<header><line feed>

<step>,<offset>,<pma>,< data><line feed>

Where:

<header>	"STEP,OFFSET,PMA,RECORD DATA"
<step>	Step number of the error.
<offset>	Pattern number.
<pma>	Pattern Memory Address.
<data>	Record memory.

Record Index Save File Format

The Record Index results are saved in the following format:

<header><line feed>

<step>,<offset><line feed>

Where:

<header>	"STEP,OFFSET"
<step>	Step number of the error.
<offset>	Record memory offset where the results are saved.

Record Data Save File Format

The Record Data results are saved in the following format:

<header><line feed>

<step>,<offset>,<data><line feed>

Where:

<header>	"STEP,OFFSET,RECORD DATA"
<step>	Step number of the error.
<offset>	Pattern number.
<data>	Record Memory contents.

CRCs Display

The CRC memory display is accessed from the **View>Results** menu bar selection and setting the **View** control to **CRCs**.

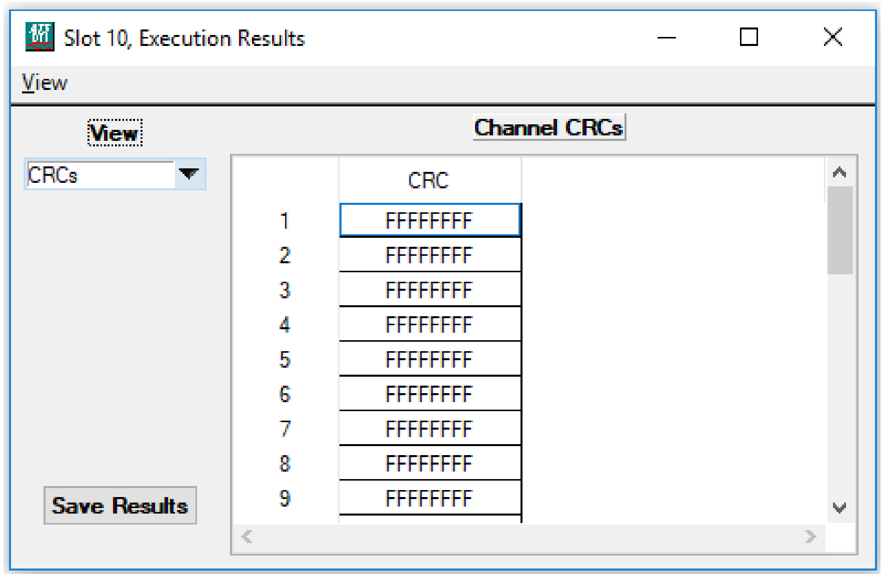


Figure 4-73 View CRC Panel

CRCs can be accumulated for all 32 channels.

Relevant vi(s):

Query CRC16 Results

- **Query CRC32 Results**

Error Address Display

The Error Address memory display is accessed from the **View>Results** menu bar selection and setting the **View** control to **Errors Address**.

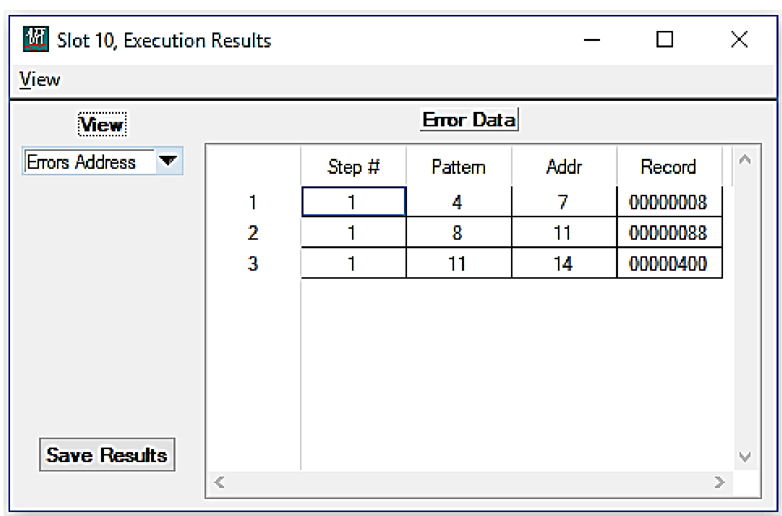


Figure 4-74 View Errors Address Panel

The error address memory records the sequence step and pattern address of the first 1024 errors of a sequence execution and is displayed in the **Step #** and **Addr** columns. The **Pattern** column is calculated based on the **Record Type** setting and **Record** column is read from the record memory.

The relevant vi for **Step #** and **Addr** data is:

Query Error Address

The relevant vi for **Record** data is:

Query Record Data

The relevant vi for **Pattern** data is:

Query Pattern Set (if **Record Type** set to Normal)

Query Record Index (if **Record Type** set to Indexed)

The View menu selection allows the address column of the error address panel to toggle between decimal and hexadecimal.

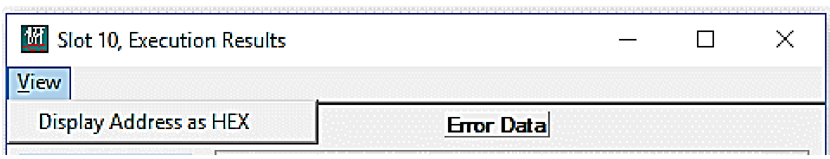


Figure 4-75 Execution Results View Menu

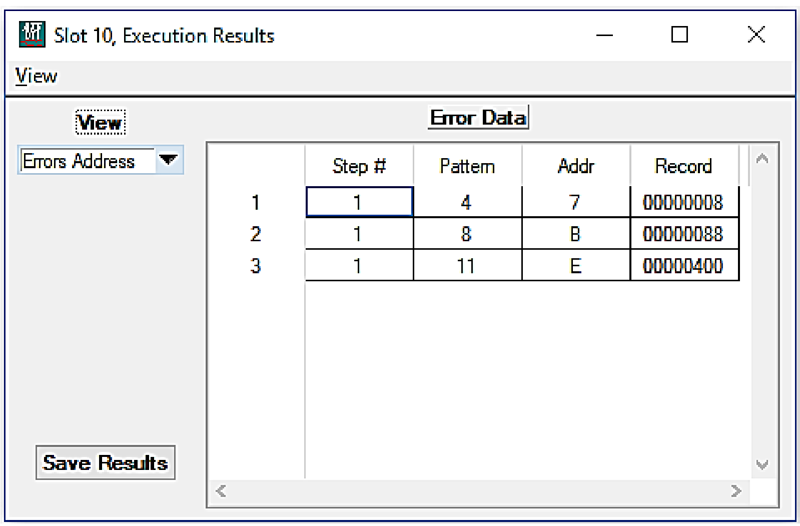


Figure 4-76 View Errors Address Panel Hex

Record Index Display

The record index memory display is accessed from the **View>Results** menu bar selection and setting the **View** control to **Record Index**.

The record index memory stores the sequence step and pattern index of the first 1024 steps of a sequence execution.

When the record type is set to indexed, the sequence results are stored sequentially in the record memory starting at offset 0. The record index memory allows the user to determine sequence step order that filled the record memory.

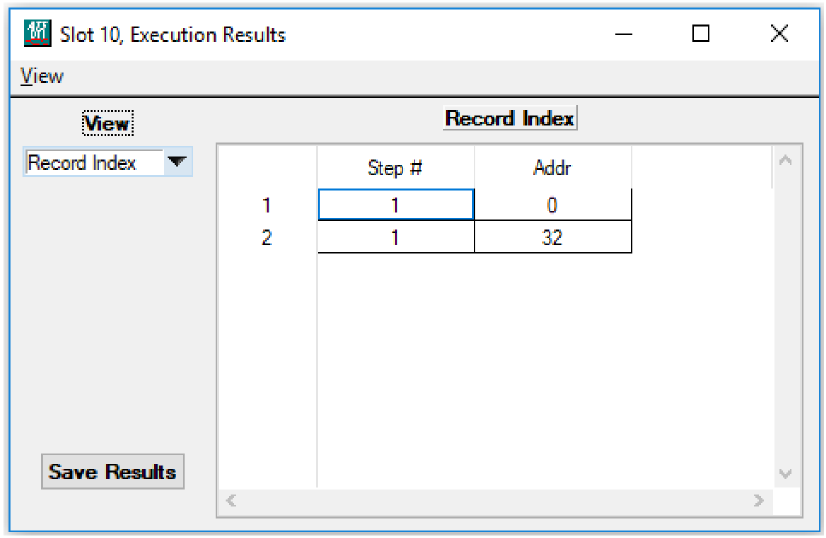


Figure 4-77 Record Index Panel

Relevant vi(s):

Query Record Index

Record Data Display

The record memory display is accessed from the **View>Results** menu bar selection and setting the **View** control to **Record Data**.

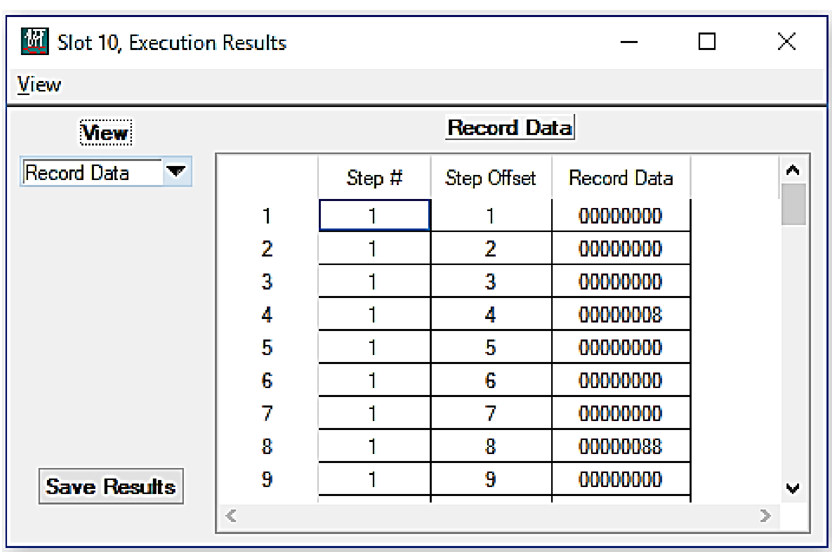


Figure 4-78 View Record Data Panel

The Record Data contains either the error or response results from the previous sequence burst (see **Step Record Mode** in this chapter).

The least significant bit of the record data (in hex) represents the error/response for channel 1 and the most significant bit represents channel 32. Error data stores a 1 to indicate a channel did not match its programmed expect value and a 0 indicates no error. Response data stores a 1 to indicate a high level and a 0 to indicate a low level. The compare level used for recoding response data is set by the Raw Record Basis (see **Raw Record Basis** in this chapter).

NOTE

If there is a Capture Fault on a channel for one or more patterns, an error will be registered. If an error is not registered it means that the channel for this pattern was not only as expected but also that there was a valid capture. If an error is registered it could mean that the channel for this pattern was either not as expected or there was a capture fault. Capture faults are registered separately so that one can determine if there was a capture fault for this channel on one or more patterns. If so, one can look for programming faults and fix them first. Once the capture faults are taken care of, any remaining errors will now be bona fide errors (channel data not as expected). See the following **Sequence Events** and **Driver/Receiver Data Panel** sections for more information about capture faults.

Relevant vi(s):

Query Record Data

Status Indicator Panels

The status indicator panels allow the operator to view the available status results to determine if the previous execution sequence is valid. The following panels are available:

- Sequencer Events
- Sequencer Data
- Driver/Receiver Events
- Driver/Receiver Data
- Backplane Trigger Readback

Sequencer Events

The sequence events display is accessed from the **View>Sequencer Events** menu bar selection.

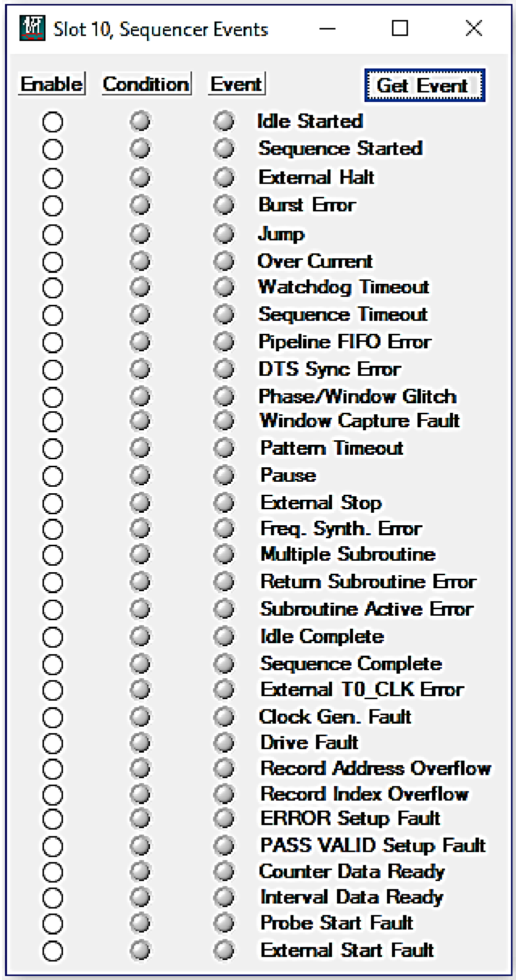


Figure 4-79 Sequencer Event Status Panel

The following sequencer enable, condition and event bits are defined:

Bit	Name	Description
0	Idle Started	The idle state has been entered
1	Sequence Started	The sequence active state has been entered.
2	External Halt	One or more external halts occurred.
3	Burst Error	One or more errors occurred.
4	Jump	One or more jumps occurred.
5	Over-Current	One or more channels generated an over-current event.
6	Watchdog Timeout	A watchdog timeout occurred.
7	Sequence Timeout	A sequence timeout occurred.
8	Pipeline FIFO Error	Pipeline depth inadequate for the Data Rate.
9	DTS Sync Error	The DTS sync error flag is set. The error step and error pattern address are available in the sequencer status panel.
10	Phase/Window Glitch	A phase or window pulse less than 8ns was detected.
11	Window Capture Fault	An expect pattern code was programmed on a channel with the capture mode set to none or the window was missing.
12	Pattern Timeout	A pattern timeout occurred.

Bit	Name	Description
13	Pause	A pause occurred.
14	External Stop	External stop signal received.
15	Freq. Synth. Error	The frequency synthesizer is selected as the master clock and is running slower than 40 kHz.
16	Multiple Subroutine	Attempt to jump to a subroutine when already in one.
17	Return Subroutine Error	Return encountered when not in a subroutine.
18	Subroutine Active Error	Sequence completed while still in a subroutine.
19	Idle Complete	Idle sequence completed.
20	Sequence Complete	Sequence completed.
21	External T0_CLK Error	The external T0_CLK is too fast or glitchy. The edges which cause the “too fast” condition are ignored such that the resultant T0_CLK period will not be allowed to be <16 Master clocks when the probe is enabled (<10 Master Clocks when not) OR, the T0_CLK is too slow (period >65.5 us with a 500 MHz master clock...or proportionately slower for a slower master clock).
22	Clock Gen. Fault	The fault is automatically corrected but one or more patterns may have been corrupted.
23	Drive Fault	A Drive Fault occurred.
24	Record Address Overflow	Indicates that the data recorded at the last memory address may be corrupted.
25	Record Index Overflow	Indicates that there is more data recorded than can be reconstructed.
26	ERROR Setup Fault	DTS Error Signal not assigned.
27	PASS VALID Setup Fault	DTS Pass Valid Signal not assigned.
28	Counter Data Ready	Frequency counter data ready.
29	Interval Data Ready	Interval Timer data ready.
30	Probe Start Fault	Probe button pushed while probe is not enabled or memory is not granted.
31	External Start Fault	External start signal while memory is not granted.

Table 4-85 Sequence Enable/Condition/Event Bit Descriptions

Enable

These radio buttons enable/disable the associated event from setting the sequencer interrupt event.

Relevant vi(s):

Set Event Enable

Condition

These LEDs indicate the current state of the associated signal.

Relevant vi(s):

Query Sequencer Condition

Event

These LEDs indicate if the state of the associated signal went true.

Relevant vi(s):

Query Sequencer Event

Clear Event

This command button resets the event LEDs.

Sequencer Data Panel

The sequence status display is accessed from the **View>Sequencer Data** menu bar selection.

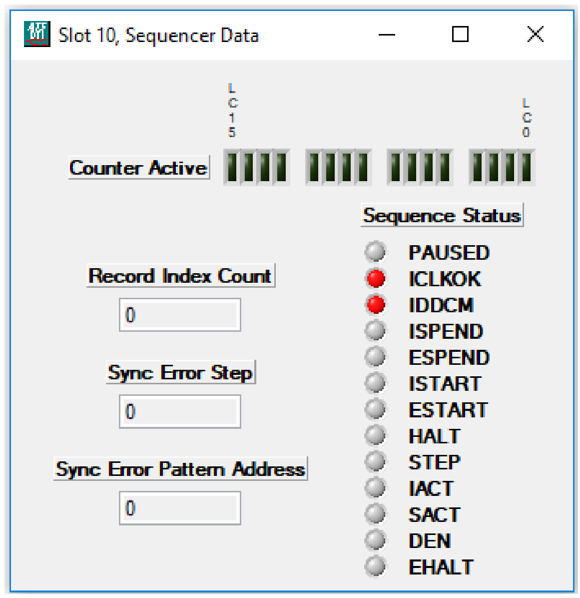


Figure 4-80 Sequencer Data Panel

Counter Active

This set of LEDs indicates whether the loop counter is active or not. An active counter will have its LED illuminated.

Relevant vi(s):

Query Sequencer Counter Status

Record Index Count

This indicator displays the number of valid entries in the record index memory.

Relevant vi(s):

Query Sequencer Record Index

Sync Error Step

This indicator displays the step number that was active when the DTS sync error occurred.

Relevant vi(s):

Query Sequencer Sync Error

Sync Error Pattern Address

This indicator displays the pattern address that was active when the DTS sync error was detected.

This pattern address may be up to 5 patterns later than the first detection of a sync error. Also, the Sync Error Step and Pattern Address is only relevant on coupled sequencers.

Relevant vi(s):

Query Sequencer Sync Error

Status

These LED indicators display the sequence status bits.

The following sequencer status bits are defined:

Bit	Name	Description
0	PAUSED	Sequencer is paused
1	ICLKOK	500 MHz Clock OK
2	IDDCM	Input Delay DCM locked
3	ISPEND	Internal Stop Pending
4	ESPEND	External Stop Pending
5	ISTART	Internal Start Pending
6	ESTART	External Start Pending
7	HALT	Sequencer is Halted
8	STEP	Single Step Pending
9	IACT	Idle Sequence Active
10	SACT	Sequence Active
11	DEN	Drivers Enabled
12	EHALT	External Halt Pending

Table 4-86 Sequence Status Bit Descriptions

Relevant vi(s):

Query Sequencer Status

Driver/Receiver Events Panel

The Driver/Receiver events display is accessed from the **Execute > DS*x* > View > Driver/Receiver Events** menu bar selection.

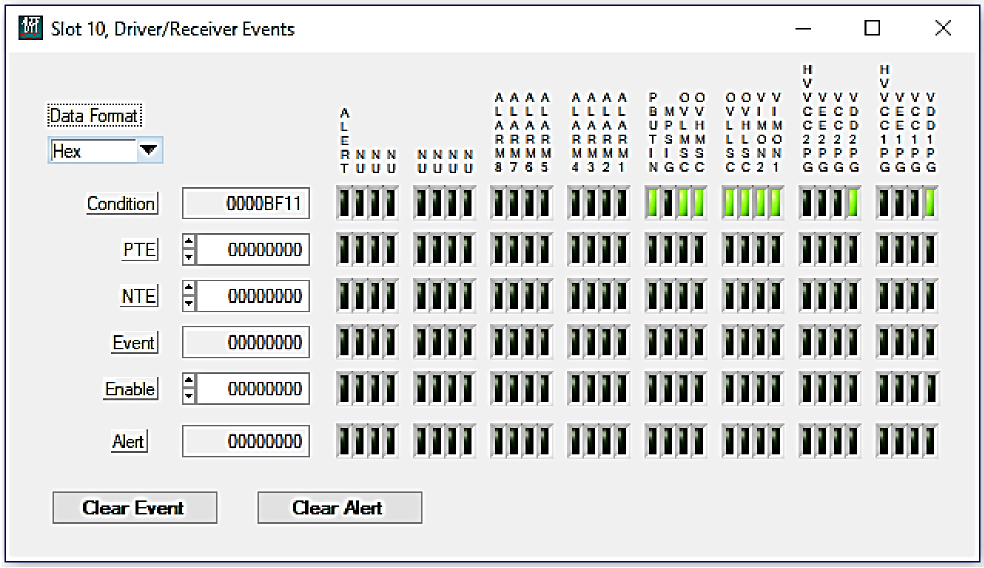


Figure 4-81 Driver/Receiver Events Panel

The following Driver/Receiver Event bits are defined:

Bit	Name	Description	Alert Bit
0	VDD1PG	VDD for CH1 through CH16 power good	Yes
1	VCC1PG	VCC for CH1 through CH16 power good	Yes
2	VEE1PG	VEE for CH1 through CH16 power good	Yes
3	HVVCC1PG	HVVCC for CH1 through CH16 power good	Yes
4	VDD2PG	VDD for CH17 through CH32 power good	Yes
5	VCC2PG	VCC for CH17 through CH32 power good	Yes
6	VEE2PG	VEE for CH17 through CH32 power good	Yes
7	HVVCC2PG	HVVCC for CH17 through CH32 power good	Yes
8	VIMON1	Not used	No
9	VIMON2	Not used	No
10	OVHLSC	Over voltage high CH1 through CH16 > HV-VCC1	Yes
11	OVLLSC	Over voltage low CH1 through CH16 < VEE1	Yes
12	OVHMSC	Over voltage high CH17 through CH32 > HV-VCC2	Yes
13	OVLMSC	Over voltage low CH17 through CH32 < VEE2	Yes
14	MPSIG	MPSIG Level	No
15	PBUTIN	One or more channels had an over-voltage.	No
16	ALARM1	CH1 to CH4 Over Temperature	Yes
17	ALARM2	CH5 to CH8 Over Temperature	Yes
18	ALARM3	CH9 to CH12 Over Temperature	Yes
19	ALARM4	CH13 to CH16 Over Temperature	Yes
20	ALARM5	CH17 to CH20 Over Temperature	Yes
21	ALARM6	CH21 to CH24 Over Temperature	Yes
22	ALARM7	CH25 to CH28 Over Temperature	Yes
23	ALARM8	CH29 to CH32 Over Temperature	Yes
31	ALERT	One or more alert bits set	No

Table 4-87 Sequence Status Bit Descriptions

Condition

These LEDs and corresponding numeric control indicate the current state of the associated signal.

Relevant vi(s):

Query FE Condition

PTE/NTE

These numeric controls set the positive transition enable (PTE) and negative transition enable (NTE).

The PTE allows low to high transitions in the condition register to set the corresponding bit in the event regiter.

The NTE allows high to low transitions in the condition register to set the corresponding bit in the event regiter.

Event

These LEDs indicate if the enabled transition state (PTE or NTE) went true.

Relevant vi(s):

Query FE Event

Enable

This numeric control enable/disables the associated bit in the event register from setting the Driver/Receiver interrupt event.

Relevant vi(s):

Set Event Enable

Alert

These LEDs indicate if an alert bit is set.

Any of the over voltage alerts will open all connect relays.

All other alert bits will open the connect relays and turn off VCC, VEE and HV_VCC regulators.

Relevant vi(s):

Query FE Alert

Clear Event

This command button resets the event LEDs.

Reset Alert

This command button resets the alert register and LEDs.

Resetting the alert register will reconnect any channel whose connect setting was closed but will not turn on the VCC, VEE and HV_VCC regulators.

Driver/Receiver Data Panel

The Driver/Receiver data display is accessed from the **View>Driver/Receiver Data** menu bar selection.

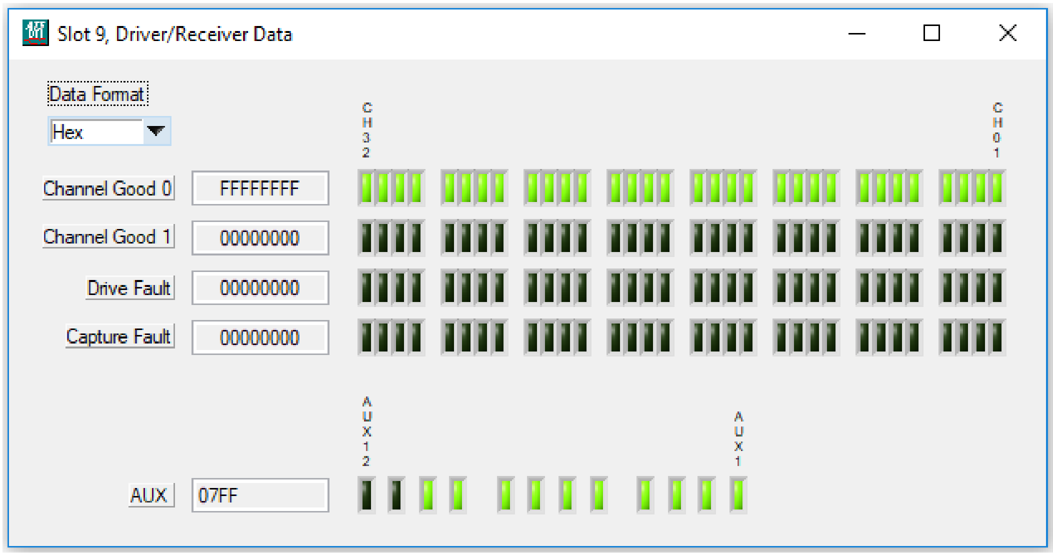


Figure 4-82 Driver/Receiver Data Panel

This panel displays the following Driver/Receiver data:

Channel Good 0	A ‘1’ (LED illuminated) indicates that the channel is currently lower than the low comparator (CVL).
Channel Good 1	A ‘1’ (LED illuminated) indicates that the channel is currently higher than the high comparator (CVH).
Drive Fault	A ‘1’ (LED illuminated) indicates that the channel has triggered a drive fault event.
Capture Fault	A ‘1’ (LED illuminated) indicates that the channel has triggered a Capture Fault. A Capture Fault Event occurs if there was an Expect without an appropriate Capture Mode (i.e. a Capture Mode of “none”) or an Expect and a Capture Mode but without appropriate Window edges within the Pattern period. Capture Faults automatically generate an Error for that Pattern. The channel(s) with a Capture Fault can be queried which may help narrow down where the Capture Fault occurred.
AUX	A ‘1’ (LED illuminated) indicates that the channel is currently higher than the high comparator.

Relevant vi(s):

Query Sequencer Channels

Query Sequencer Aux

Query Sequencer Drive Fault

Query Capture Fault

PXI Trigger Readback Panel

The PXI trigger readback display is accessed from the **View>PXI Trigger Readback** menu bar selection.

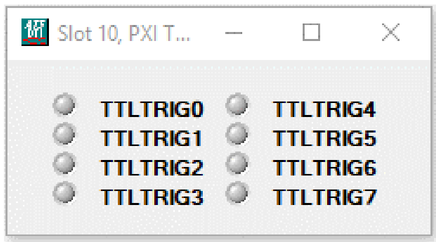


Figure 4-83 PXI Trigger Readback Panel

This panel displays the current level of the eight TTL backplane triggers. The LED illuminated indicates a high state.

Note: *A "high" TTLTRG signal is active "low" on the backplane.*

Relevant vi(s):

Query TTL Triggers

Instrument Functions

The Instrument menu bar selections perform the following:

- Self-test functions
- Calibration
- Firmware Updates
- Temperature Monitoring
- Voltage Monitoring

Self Test

The self-test function is accessed from the **Instrument >Self Test** menu bar selection.

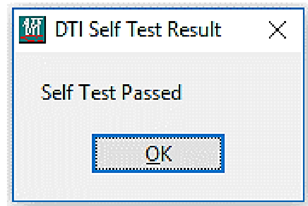


Figure 4-84 Self Test Result Message

The self-test function saves the current instrument configuration and performs a verification program on the hardware. The results are returned as a bit encoded number where a 1 indicates a failure. The instrument configuration is re-loaded after the test.

The self-test bit codes are:

Bit	Description
0	500MHz frequency timeout
1	500 MHz frequency out of range
2	Clock synthesizer frequency timeout
3	Clock synthesizer out of range
4	CLK10 frequency timeout
5	CLK10 frequency out of range
6	Pulse Generator frequency timeout
7	Pulse Generator out of range
8	Digital board POST error
9	Driver/receiver POST error
10	Sequence RAM test failed
11	Timing set RAM test failed
12	Persistence RAM test failed
13	Waveform RAM test failed
14	Record index RAM test failed
15	Error address RAM test failed
16	Pattern 0 (CH1-CH8) RAM test failed
17	Pattern 1 (CH9-CH16) RAM test failed
18	Pattern 2 (CH17-CH24) RAM test failed
19	Pattern 3 (CH25-CH32) RAM test failed
20	Record RAM test failed
21	Flag RAM test failed

Table 4-88 Self-Test Result Code Descriptions

Relevant vi(s):

Self-Test

Full RAM Test

The full RAM test function is accessed from the **Instrument>Full RAM Test** menu bar selection.

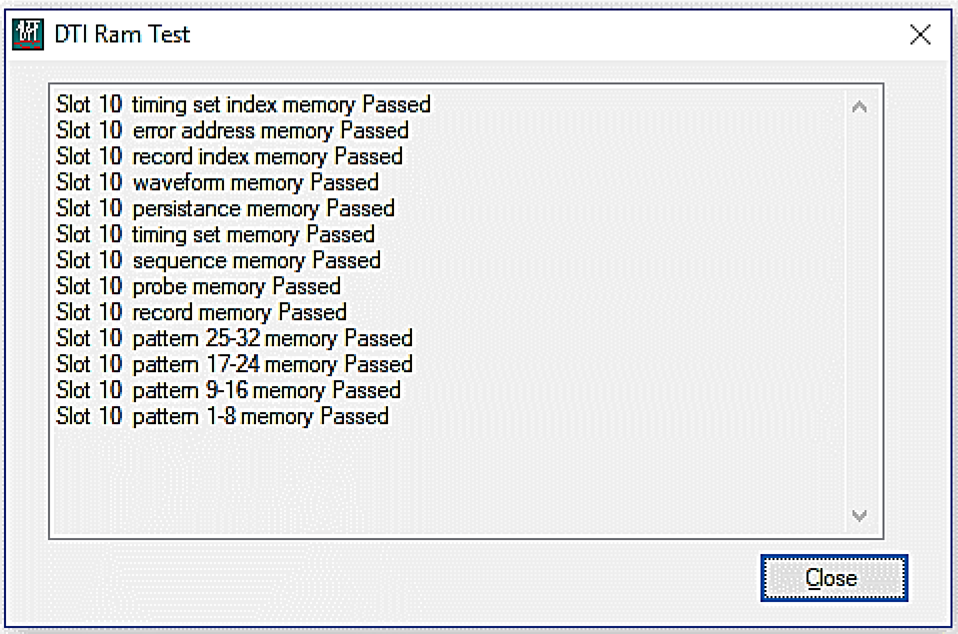


Figure 4-85 Full RAM Test Results Panel

The full RAM test function saves the instrument configuration and performs a full RAM test on all the internal and external memories. The full RAM test performs multiple read/write cycles to each RAM at every address location. The full RAM test utilizes special hardware to test the pattern, record and probe memories at speed.

Relevant vi(s):

Ram Test

Calibration Panel

The calibration function is accessed from the **Instrument >Calibrate** menu bar selection. Programmable Channel Calibration for field calibration procedure.

Calibration data is stored on the Driver/Receiver board in non-volatile memory. The calibration procedure requires that the module be reset. Any unsaved data will be lost.

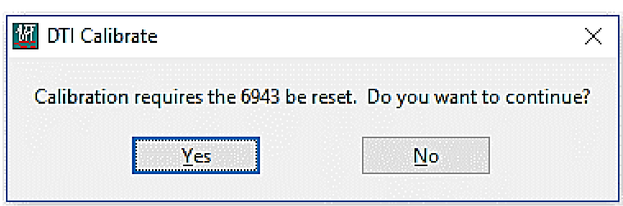


Figure 4-86 Calibration Confirmation Panel

Selecting **Yes** displays the main calibration panel.

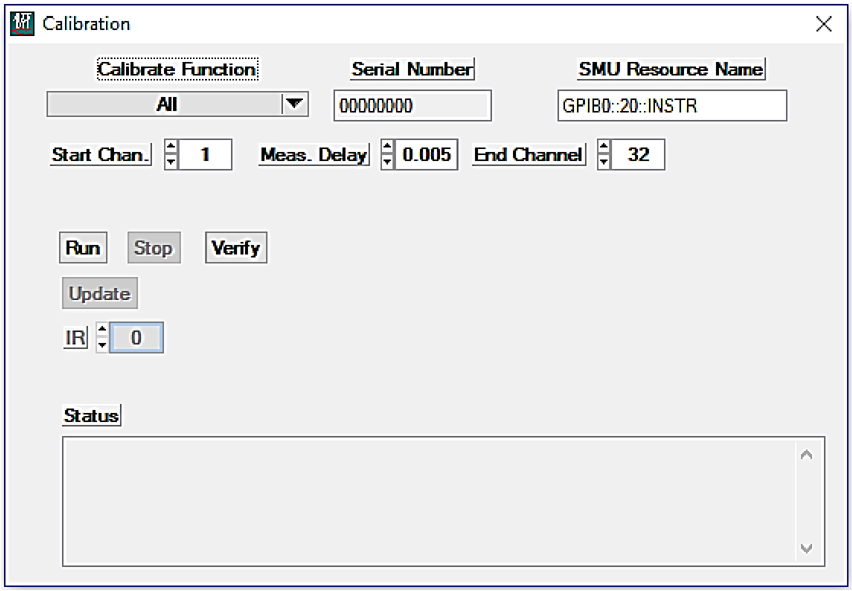


Figure 4-87 Calibration Panel

Calibrate Function

This pull-down control selects the calibrate function.

Setting	Description
All	Selects all the calibrations
ADC Reference	Measure voltage references used to calibrate the ADC: +5V +3.33V +1.66V -1V
Load Reference	Measure load resistors used for validation: 50Ω 10KΩ
ADC Gain	ADC Programmable Gain Instrumentation Amplifier calibration.
Load PGIA	Programmable Gain Instrumentation Amplifier used for validation.
Measure Voltage	Measure voltage ASIC calibration.
DAC Overlap	ASIC DAC overlap calibration

Setting	Description
PMU	ASIC PMU all ranges calibration Force Voltage Measure Current Offset Common mode adjust positive Common mode adjust negative Measure current source Measure current sink Current clamp high Current clamp low Force current Voltage clamp high Voltage clamp low
Drive Levels	ASIC driver levels calibration DVH DVL VTT
Compare Levels	ASIC comparator levels calibration CVH CVL
Active Load	ASIC active load calibration. VCOM Source VCOM Sink Source Load Sink Load
Delete Calibration	Calibration data in non-volatile memory

Table 4-89 Calibrate Function Settings

Serial Number

This control displays the Driver/Receiver board serial number.

SMU Resource Name

This control specifies the SMU resource name used for calibrating the voltage reference, load reference, load PGIA, measure voltage, PMU and Active Load.

The SMU can be any of the Keithley Model 24nn series.

Start Chan.

This numeric control sets the first channel to be calibrated. The valid range is from 1 (CH1) to 32 (CH32).

Meas. Delay

This numeric control sets the delay (in seconds) between changing a channel level and measuring the channel. The valid range is from 0.005 to 1.0.

End Channel

This numeric control sets the number of channels to be calibrated, starting with the **Start Chan.** setting. The valid range is from 1 to 32.

Run

This command button executes the selected calibrate function.

The SFP will prompt the operator to confirm the action and then apply power to the Driver/Receiver board.

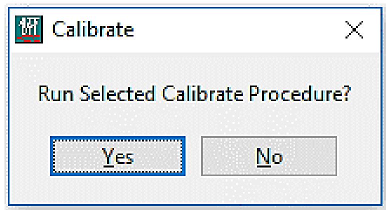


Figure 4-88 Confirm Calibrate Panel

The selected calibration procedures will begin when the temperature reaches 80º C or the **Continue** command button is pressed. The unit should be calibrated at its normal application temperature. Refer to the **Calibration Temperature** section in Chapter 5 for more information.

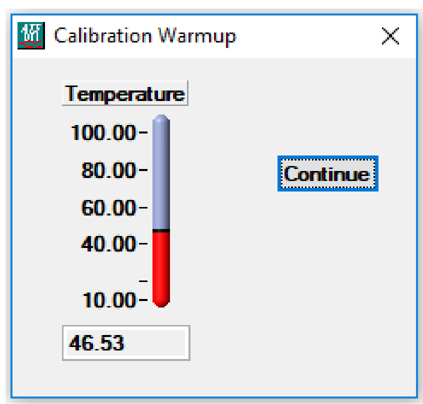


Figure 4-89 Calibrate Warm-up Panel

After warm-up the operator will be prompted to connect the EXTFORCE (E_F) pin to the SMU.

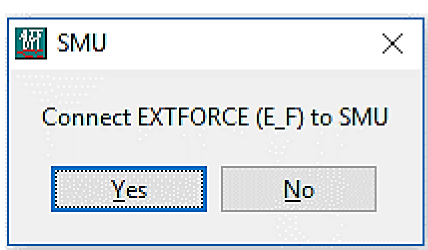


Figure 4-90 Connect EXTFORCE to SMU

Once calibration has begun, progress data is displayed in the **Status** control.

The calibration run procedure creates a file “calData_<SN>.txt” and writes calibration data analyses data. This file can be used to validate calibration results.

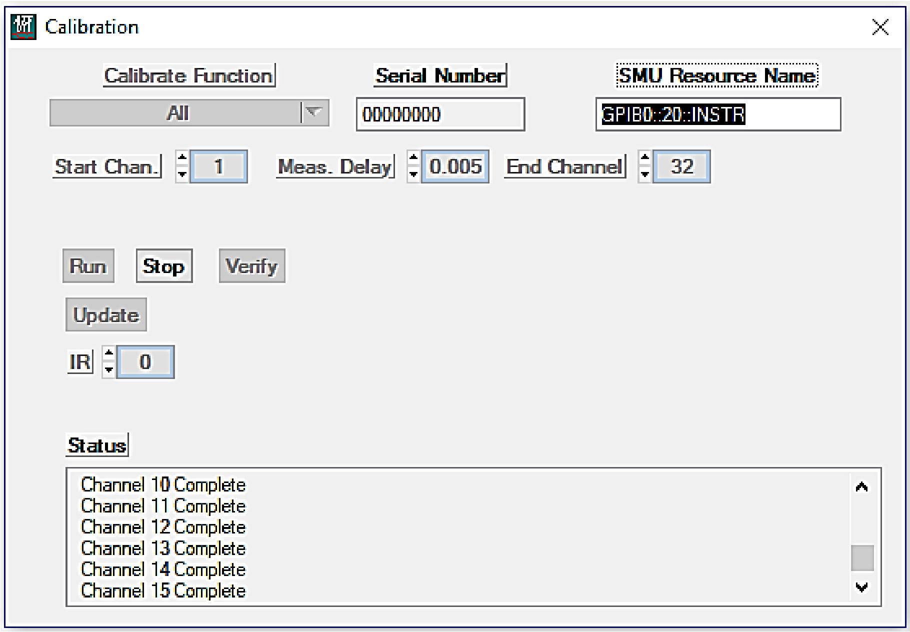


Figure 4-91 Calibrate Run Panel

Relevant vi(s):

- **Set Power Settings**
- **Set Power Connect**
- **Calibrate Channel**

Verify

This command button executes the selected calibrate function verify routine.

The SFP will prompt the operator to confirm the action and then apply power to the Driver/Receiver board.

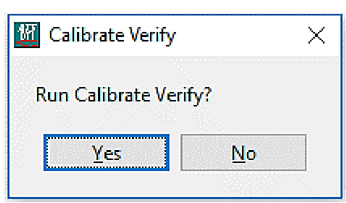


Figure 4-92 Confirm Verify Panel

The operator will be prompted to select the directory where the verification report will be created and saved.

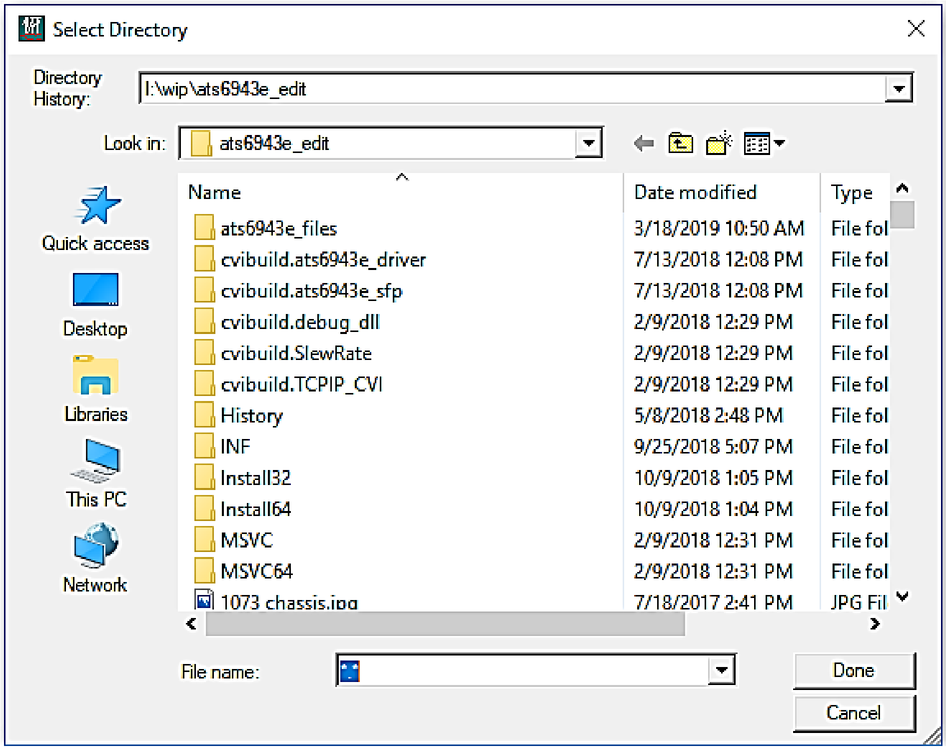


Figure 4-93 Verify Select Directory Panel

Verification will begin when the temperature reaches 80º C or the **Continue** command button is pressed. The unit should be verified at its normal application temperature. Refer to the **Calibration Temperature** section in Chapter 5 for more information. Once verification has begun, progress data is displayed in the **Status** control.

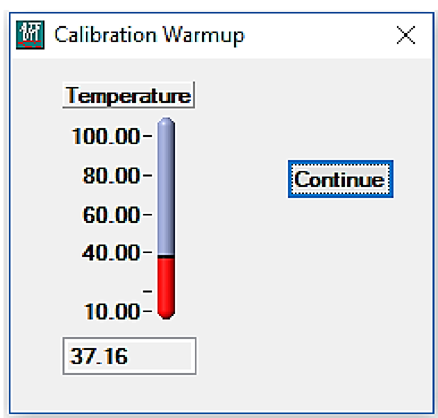


Figure 4-94 Verify Warm-up Panel

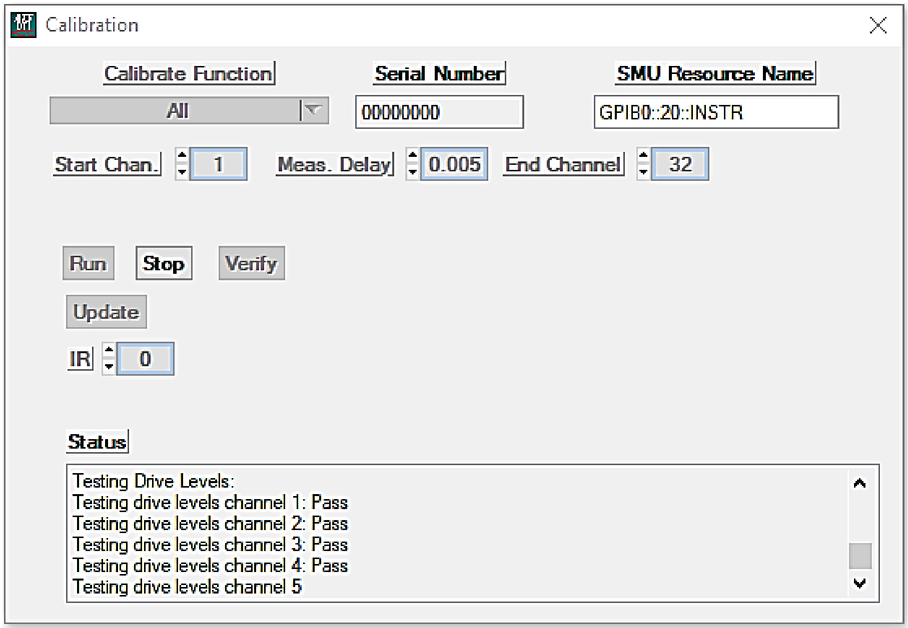


Figure 4-95 Verify Run Panel

Relevant vi(s):

Verify Channel Calibration

Stop

This command button stops a calibration or verification run.

Update

This command button writes the new calibration data to non-volatile memory.

Update Firmware

The update firmware panel is accessed from the **Instrument>Update Firmware** menu bar selection.

There are six possible FPGA/firmware updates on a module. The file name identifies which of the six updates is targeted.

Update Target	File Prefix
Digital Board FPGA and µP	801045-001
Driver/Receiver FPGA and µP	801029-001
Sequencer FPGA	801058-001
Inter-module FPGA Primary	801057-001
Inter-module FPGA Secondary	801057-002
Inter-module FPGA Terminator	801057-003

Table 4-90 Update Firmware File Prefix

After selecting the update file, the non-volatile memory will be programmed with the selected file. All targets will be re-loaded after the update except for the digital board FPGA and µP, power must be cycled for this update to load.

Relevant vi(s):

Update Module FPGA

Voltage Monitor

This voltage monitor panel is accessed from the **Instrument>Voltage Monitor** menu bar selection.

This panel displays voltage and current data from the digital and driver board and is used for validation and test.

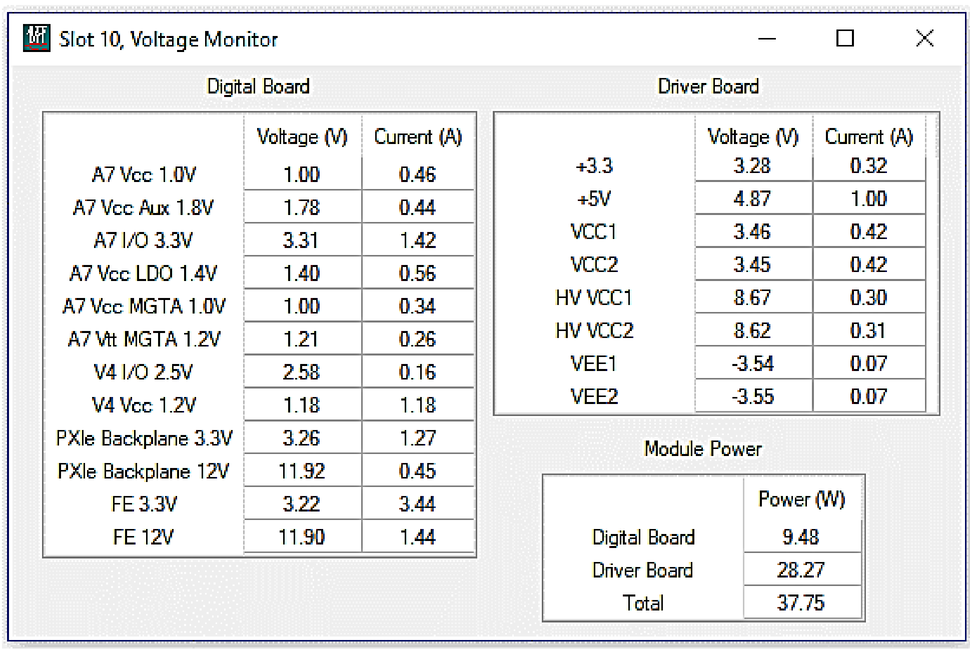


Figure 4-96 Voltage Monitoring Panel

Relevant vi(s):

Query Voltage Monitor

ADC Monitor

This ADC monitor panel is accessed from the **Instrument>ADC Monitor** menu bar selection.

This panel displays voltage data and logic settings from the driver board and is used for factory calibration, validation and test.

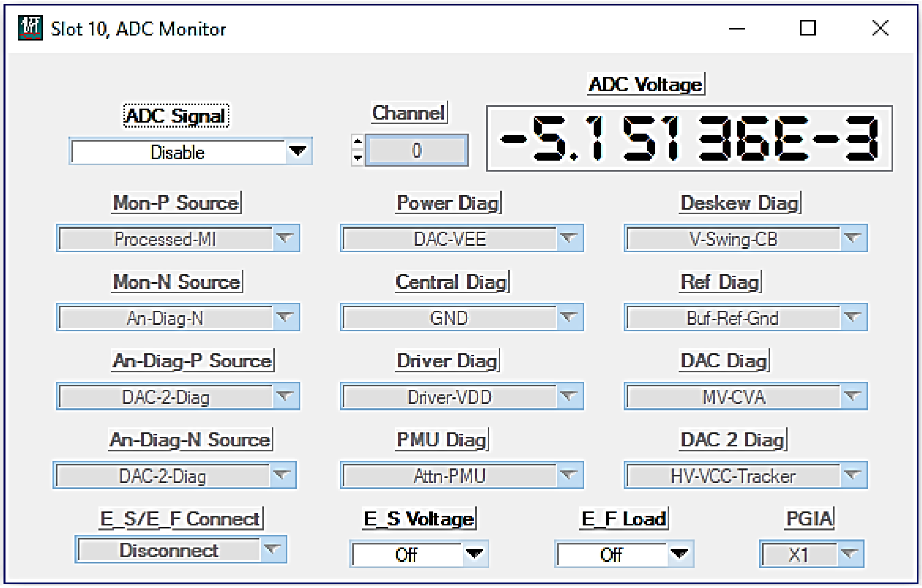


Figure 4-97 ADC Monitor Panel

Figure 4-98 shows the ADC logic on the driver board.

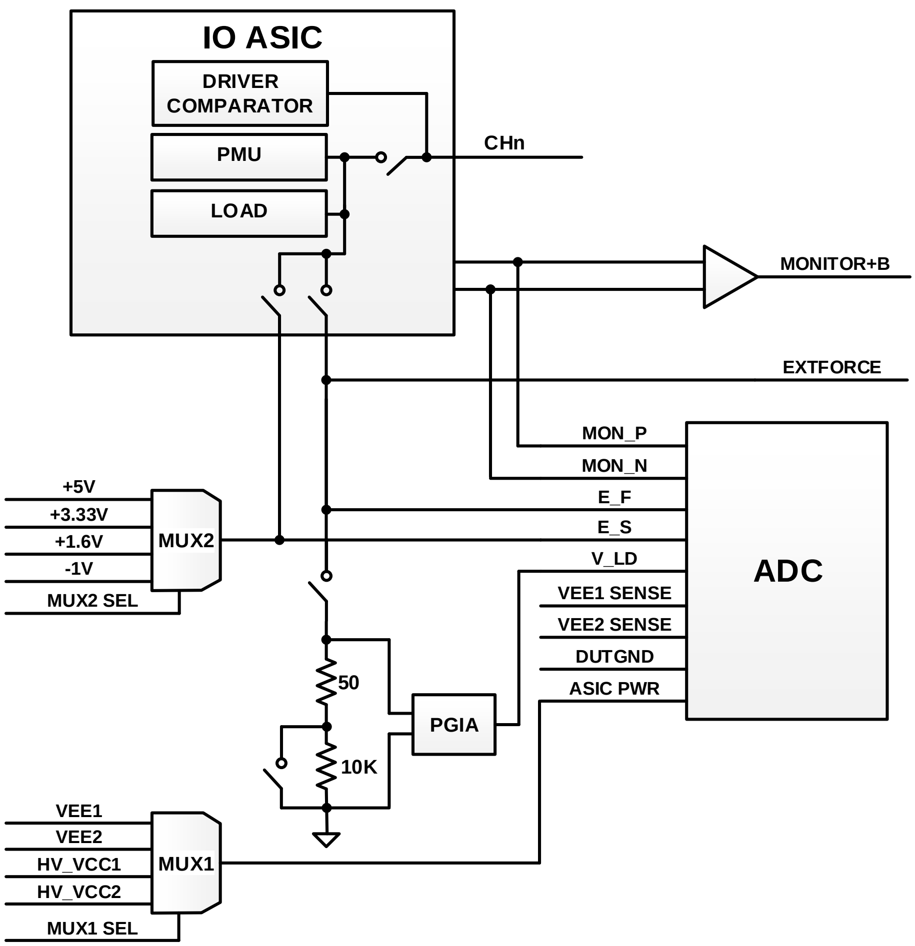


Figure 4-98 ADC Monitor Logic

ADC Signal

This pull-down control displays the signal options for the ADC input.

Setting	Description
Disable	No signal selected
Monitor	Selects the ASIC MON_P and MON_N differential pair signals
VEE1 Current	Selects the VEE1 current sense signal
Variable Load	Selects the V_LD signal
External Sense	Selects the E_S signal and disables MUX2
+3.33V	Selects the E_S signal and sets MUX2 to +3.33V
+1.66V	Selects the E_S signal and sets MUX2 to +1.66V
+5V	Selects the E_S signal and sets MUX2 to +5V
-1V	Selects the E_S signal and sets MUX2 to -1V
External Force	Selects the E_F signal
VEE2 Current	Selects the VEE2 current sense signal
VEE1	Selects the ASIC PWR signal and sets MUX1 to VEE1
VEE2	Selects the ASIC PWR signal and sets MUX1 to VEE2
HV_VCC1	Selects the ASIC PWR signal and sets MUX1 to HV_VCC1
HV_VCC2	Selects the ASIC PWR signal and sets MUX1 to HV_VCC2
DUTGND	Selects the DUTGND signal

Table 4-91 ADC Signal Settings

Relevant vi(s):

Query ADC

Query ADC Average

Set Monitor Signal

Channel

This numeric control selects the channel when the ADC source is set to Monitor, Variable Load, External Secse or External Force.

Relevant vi(s):

Set Monitor Signal

Mon-P Source

This pull-down control selects the source of the ASIC MON_P signal when the ADC signal is set to Monitor.

Relevant vi(s):

Set Monitor Signal

Mon-N Source

This pull-down control selects the source of the ASIC MON_N signal when the

ADC signal is set to Monitor.

Relevant vi(s):

Set Monitor Signal

An-Diag-P Source

This pull-down control select the source of the An-Diag-P signal when the Mon-P signal is set to An-Diag-P.

Relevant vi(s):

Set Monitor Signal

An-Diag-N Source

This pull-down control select the source of the An-Diag-N signal when the Mon-N signal is set to An-Diag-N.

Relevant vi(s):

Set Monitor Signal

Diag Control Selections

The Diag pull-down controls (Power, Central, Driver, PMU, Deskew, Ref, DAC and DAC 2) are used to select the diagnostic signal when An-Diag-P or An-Diag-N is selected.

E_S/E_F Connect

This pull-down control programs the selected channels E_F and E_S connections in the IO ASIC.

Relevant vi(s):

Set Reference Connect

E_S Voltage

This pull-down control programs the MUX2 selection.

Relevant vi(s):

Set Reference Output

E_F Load

This pull-down control programs the E_F load connection relays.

Relevant vi(s):

Set Sense Connect

PGIA

This pull-down control selects the PGIA gain when the ADC signal is to the Chip Temperature Panel

This chip temperature panel is accessed from the **Instrument>Chip Temperature** menu bar selection.

This panel displays the chip temperature read from the IO ASIC and allows an alert temperature to be set.

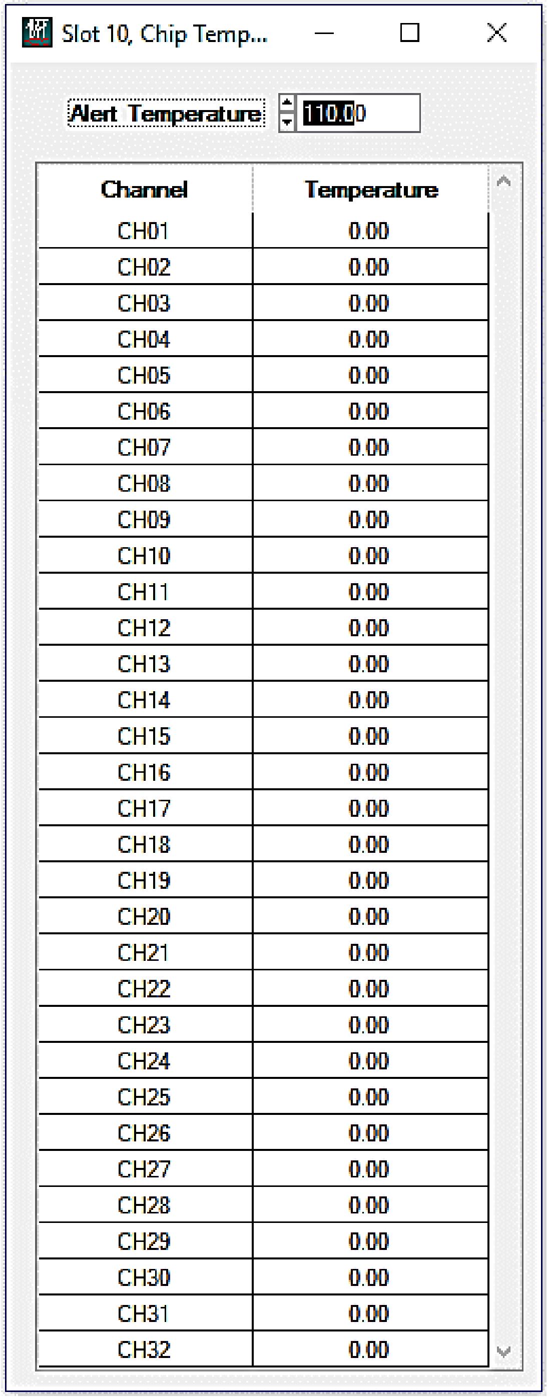


Figure 4-99 Chip Temperature

Relevant vi(s):

Query Channel Temperature

Alert Temperature

The alert temperature can be set from 70 to 130 degrees Celsius in increments of

10 degrees.

There are eight signals that are monitored, one signal per ASIC and four channels per ASIC.

If any of the four channels exceed the alert temperature, an alarm event is generated than will shut down the ASIC power supplies and open the connect relays.

Relevant vi(s):

Set Temperature Alarm

SFP Close Message

This panel is used to close the soft front panel.

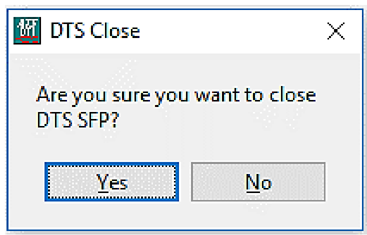


Figure 4-100 SFP Close Message

If "Yes" is selected, the following panel will be displayed:

Figure 4-101 SFP Reset Message

Relevant vi(s):

Close

Reset

LabVIEW vi's and API Library

This chapter describes the LabVIEW vi's and also shows the equivalent prototype functions of the C API library. The LabVIEW and API Library is segmented into the following hierarchical classes.

Class	Description
Root	Contains the initialize and close functions used to initiate and close a communication session.

File Functions	This class contains functions to load and save instrument configuration settings.
Application	This class contains DTS application functions.
Configuration	This class contains the functions to configure the DTI structures.
Module	This class contains the module configuration.
Data Sequencer	This class contains the data sequencer configuration.
Timers	This class contains the timer configuration.
Triggers	This class contains the trigger configuration.
Front-end	This class contains the front-end settings.
Dynamic	This class contains the front-end dynamic channel configuration.
Static	This class contains the front-end static channel configuration.
Auxiliary	This class contains the front-end AUX channel configuration.
Edit	This class contains the functions for editing timing sets, sequence settings and steps, pattern memory and waveforms.
Timing Set Functions	This class contains functions to program and query timing sets.
Sequence Functions	This class contains functions to program and query sequence control and memory.
Sequence Step Functions	This class contains functions to program and query sequence step memory.
Pattern Set Functions	This class contains functions to program and query pattern sets.
Waveform Functions	This class contains functions to program and query waveforms.
Executions Functions	This class contains functions to execute, control and query sequence execution.
Execution Results	This class contains functions which returns the sequence execution results.
Sync Output	This class contains functions which programs the sync pulse.
Pulse Generator	This class contains functions which programs the pulse generator.
PMU	This class contains functions that support the PMU functions.
Counter Timer	This class contains functions that support the PMU functions.
Status and Events	This class contains functions which programs the event generation and status reporting capabilities.
PXI Trigger	This class contains functions which programs the TTLTRG signal routing.
PXI Interrupt	This class contains functions which programs the PXI interrupt event and enables.
Utility Functions	This class contains the utility functions.
Auto Connect	This class contains the auto connect functions.
Calibration	This class contains the calibration functions.
Low Level Routines	This class contains the low level functions.
Register Access	This class contains the register access functions.

Table 4-92 LabVIEW vi and API Class Hierarchy

Default Conditions

The default conditions are the settings that the module will adopt when powered up or reset.

Parameter	Power on/Reset Value
Module Configuration	

Parameter	Power on/Reset Value
Inter Module Mode	Independent
IO Max 1..16, IO Max 17..32	7.0
Delay Signal, Delay and Coarse Delay	Auto Calibrated
PXI Triggers	All disabled
D/R Properties	
DUT_GND	Sig Gnd
MFSIG Source	MPSIG
MPSIG Signal	None
Sequencer Record Mode	Disabled
Configure Data Sequencer Clock	
Master Clock	500MHz
System Clock	Internal T0CLK
Synthesizer	Disabled
Configure Data Sequencer Timers	
Watchdog	Event Only, 40ns
Sequence Timeout	Off, 20ns
Pattern Timeout	20ns
Pattern Delay 1 and 2	20ns
Configure Data Sequencer Triggers	All disabled
Configure Data Sequencer Pulse Generator	Disabled
Configure Data Sequencer Settings	
Error Record Basis	Good 1
Raw Record Basis	Good 0
Error Count Basis	Local
Record Offset	10
Record Type	Normal
Error Address Basis	Local
Timing Mode	Per Step Multi
Output to Input Disable	Phase
Pass Fail Basis	Local
Drive Fault	Disable
Pass Valid Mode	Disable
Static State	Off
Jump Pass Fail	Normal
Phase 3 Mode	Normal
Window 3 Mode	Normal
Window 3 Delay	0
CRC Preload	Ones
CRC Algorithm	0x82608EDB
Capture Mask	0
Configure Channels	
Function	Disabled
Stimulus Signal	Phase 1
Stimulus Format	Non Return
Capture Signal	Window 1
Capture Mode	Masked
Static Mode	Off
Channel Properties	
DVH	2.4V
VTT	0V
DVL	0V
CVH	2.0V
CVH	0.8V
Channel Connect	Open

Parameter	Power on/Reset Value
Slew Load State Comparator Delay Termination Channel Mode	Low Power Off 0 50 Single-ended
Config AUX Outputs	All disabled
Auto Trigger Mode (both channels)	Manual
Edit Timing Sets	All timing sets set to: Phase Assert 0 Phase Return 12 Window Open 0 Window Close 12
Edit Pattern Sets	Step 0 pattern set to: Size 1 Offset 0 Test ‘n’ (Disable pass/fail) Data ‘R’ (Repeat previous)
Edit Waveforms	Waveforms 1 to 4 set to: Table Size 16 x 1K Definition 0 Waveforms 5 and 6 set to: Definition 0,0,0,0
Edit Sequence Parameters Loop Counter Terminal Count Action Pipeline Vector Strobe Vector Bits Vector Table Channel Test	 Reload 0 Window 1 All Bits set to None All Vector Bit Index set to step 0 All Mask and Expect set to 0
Edit Sequence Steps	Step 0: Period 100 CPP 1 Jump Type None Last Step True Sequence TO Reset Gosub Return False SF1/SF2 Low PF Clear Default Record Mode None Properties: Pause Signal None Waveforms Disabled Phase Trigger System Clock

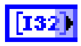
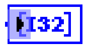
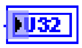
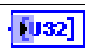
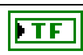
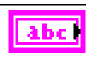
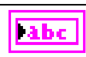
Parameter	Power on/Reset Value
Execute Data Sequencer	
Channel Drivers	Disabled
V+/V-	Off
Burst Count	1
Halt Mode	Disabled
Stop Mode	End of Pattern
Finish Mode	Standby Step 0
CRC Type	CRC32
Sync Pulse 1 and 2	Disabled
Sequencer Event Enable	0
D/R PTE and NTE	0
D/R Event Enable	0
Execute PMU FV	
Current Range	5µA
FV	0V
Current Clamp High	0mA
Current Clamp Low	0mA
Execute PMU FI	
Current Range	5µA
Connect Voltage (FV)	0V
FI	0A
Voltage Clamp High	0V
Voltage Clamp Low	0V
Execute Counter/Timer	
Function	Frequency
Input 1	CH1
Slope	Positive
Aperture	100ms
Trigger	None
Instrument Chip Temperature Alert	110C

Table 4-93 Power on/Reset Defaults

LabVIEW Parameter Types

The LabVIEW parameter list follows the description with the defaults and valid ranges belonging to the parameters. The following parameter types are used in the parameter lists to describe the parameter types.

Type	Details
I/O	Input I/O stream
I16	2-byte (16 bit) signed short input
U16	2-byte (16 bit) signed short input
I16	2-byte (16 bit) signed short output
[I16]	2-byte (16 bit) signed short array output
I32	4-byte (32 bit) signed long input

Type	Details
	4-byte (32 bit) unsigned long input
	4-byte (32 bit) signed long output
	4-byte (32 bit) signed long array input
	4-byte (32 bit) signed long array output
	4-byte (32 bit) unsigned long output
	4-byte (32 bit) unsigned long array output
	Enumerated type input
	Boolean type (true or false) input
	Boolean type (true or false) output
	Double precision floating point input
	Double precision floating point output
	String input
	String output

C API Parameter Types

The C API driver uses the VISA library to communicate with the module. The following VISA parameter types are used (defined in visatype.h header):

Type	Details
ViSession	4-byte (32 bit) unsigned long
ViPSession	reference variable (pointer) to a 4-byte (32 bit) unsigned long
ViRsrc	reference variable (pointer) to a 1-byte (8 bit) signed char
ViInt32	4-byte (32 bit) signed long
ViUInt32	4-byte (32 bit) unsigned long
ViPInt32	reference variable (pointer) to a 4-byte (32 bit) signed long
ViPUInt32	reference variable (pointer) to a 4-byte (32 bit) unsigned long
ViInt16	2-byte (16 bit) signed short
ViUInt16	2-byte (16 bit) unsigned short
ViPInt16	reference variable (pointer) to a 2-byte (16 bit) signed short
ViPUInt16	reference variable (pointer) to a 2-byte (16 bit) unsigned short
ViChar	1-byte (8 bit) signed char
ViReal64	8-byte (64 bit) floating point
ViPReal64	reference variable (pointer) to a 8-byte (64 bit) floating point
ViBoolean	2-byte (16 bit) unsigned short
ViPBoolean	reference variable (pointer) to a 2-byte (16 bit) unsigned short
ViString | ViChar	reference variable (pointer) to a 1-byte (8 bit) signed char

ViStatus	4-byte (32 bit) signed long

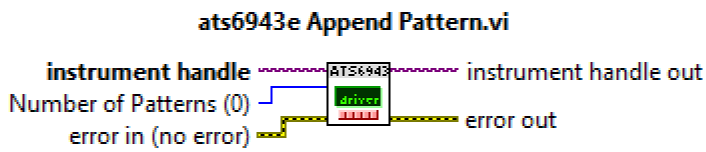
All functions use the standard calling conventions (stdcall or WINAPI). Every function returns the ViStatus (type: 32-bit integer). A negative value corresponds to an error. After a successful completion, the return status is VI_SUCCESS, which corresponds to a 0. A positive value indicates a warning. The status codes are defined in section xxx.

LabVIEW vi Descriptions

The following sections describe the LabVIEW vi functions and include the C function prototype.

Append Pattern

LabVIEW Diagram:



Description:

This vi appends the specified number of patterns to the end of an existing Pattern set.

The pattern code for all channels of the appended patterns is initialized to "Repeat Previous Code".

Use **Select Sequence Step** to select the sequence step to program.

DTS Operation:

All coupled modules.

Parameters:

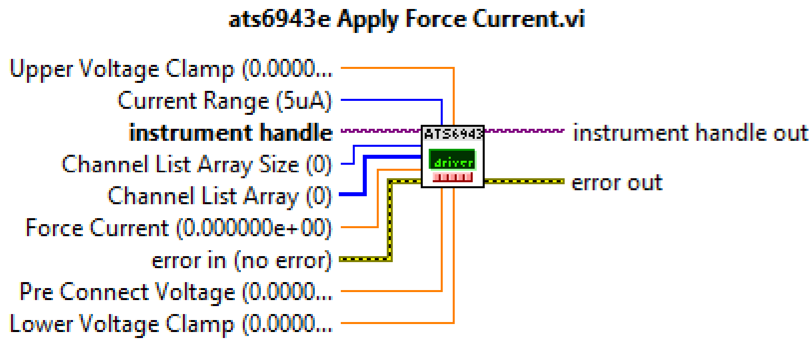
Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Number of Patterns	I32	Number of patterns to append to the existing pattern set.	1 to 262143

C Function Prototype Form:

ViStatus ats6943e_appendPattern (ViSession instrumentHandle, ViInt32 numberOfPatterns);

Apply Force Current

LabVIEW Diagram:



Description:

This vi applies the force current parameters and then connects the PMU of the specified channels. If the channel(s) function is not set to PMU FI, then only the parameters are set.

Parameters:

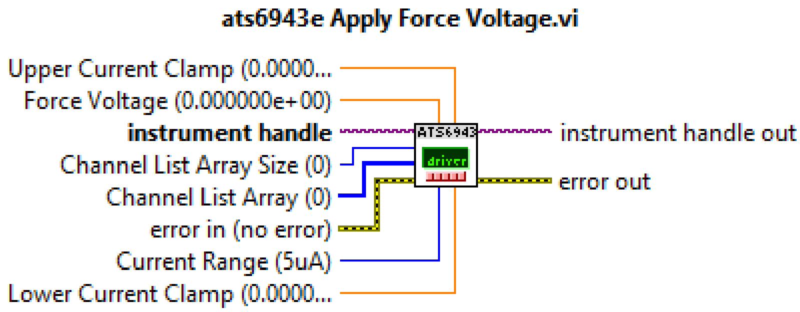
Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Force Current	DBL	The force current to set in mA.	-50 to 50 Must be within +/- 2 * Current Range.
Current Range	◂▸	Current Range Setting.	0 – 5uA 1 – 50uA 2 – 500uA 3 – 5mA 4 – 50mA
Pre Connect Voltage	DBL	Prior to connecting the PMU the voltage clamps are programmed to this value +/- 100mV, i.e., upper clamp = pre connect voltage + 0.1 and lower clamp = pre connect voltage - 0.1.	-2 to +7
Upper Voltage Clamp	DBL	Specify the upper voltage clamp value.	-2.5 to +7.5
Lower Voltage Clamp	DBL	Specify the lower voltage clamp value.	-2.5 to +7.5

C Function Prototype Form:

ViStatus ats6943e_pmuApplyFI (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 forceCurrent_mA, ViInt16 currentRange, ViReal64 preConnectVoltage, ViReal64 upperVoltageClamp, ViReal64 lowerVoltageClamp);

Apply Force Voltage

LabVIEW Diagram:



Description:

This vi applies the force voltage parameters and then connects the PMU of the specified channels. If the channel(s) function is not set to PMU FV, then only the parameters are set.

Parameters:

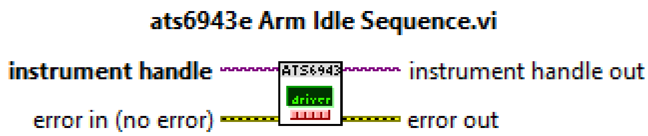
Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Force Voltage	DBL	The force voltage to set in V.	-2 to +7
Current Range		Current Range Setting.	0 – 5uA 1 – 50uA 2 – 500uA 3 – 5mA 4 – 50mA
Upper Current Clamp	DBL	Upper current clamp value in mA.	-50 to 50 Must be within +/- 2 * Current Range.
Lower Current Clamp	DBL	Lower current clamp value in mA.	-50 to 50 Must be within +/- 2 * Current Range.

C Function Prototype Form:

ViStatus ats6943e_pmuApplyFV (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 Force_Voltage, ViInt16 currentRange, ViReal64 upperCurrentClamp, ViReal64 lowerCurrentClamp);

Arm Idle Sequence

LabVIEW Diagram:



Description:

This vi arms the idle sequence.

This is used to prime the sequence logic prior to an external start trigger.

Use **Set Idle Sequence** vi to specify the idle sequence step number.

DTS Operation:

All coupled modules, primary last.

Parameters:

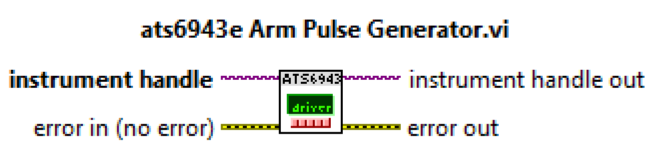
Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_armIdleSequence (ViSession instrumentHandle);

Arm Pulse Generator

LabVIEW Diagram:



Description:

Arm the pulse generator.

Parameters:

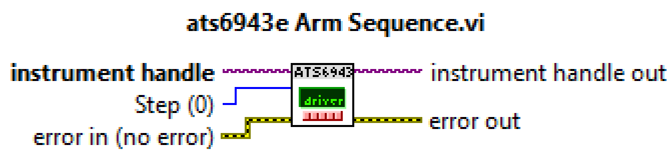
Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_armPulseGenerator (ViSession instrumentHandle);

Arm Sequence

LabVIEW Diagram:



Description:

Arms the specified sequence step.

This is used to prime the sequence logic prior to an external start trigger.

DTS Operation:

All coupled modules.

Parameters:

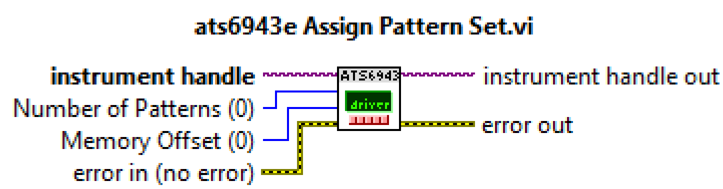
Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Step	I16	Sequence step number to arm.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_armSequence (ViSession instrumentHandle, ViInt16 step);

Assign Pattern Set

LabVIEW Diagram:



Description:

Assign a pattern set for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Assigning a pattern set consists of specifying the number of patterns and a beginning offset.

Multiple steps can be assigned to the same pattern offset.

DTS Operation:

All coupled modules sequence steps should be programmed with the same numbers of patterns in order to remain synchronized with each other.

Key Parameters:

Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Number of Patterns	I32	Number of patterns to create in the pattern set.	1 to 262144
Memory Offset	I32	The memory offset of the pattern set.	0 to 262143 The memory offset must be a multiple of 4.

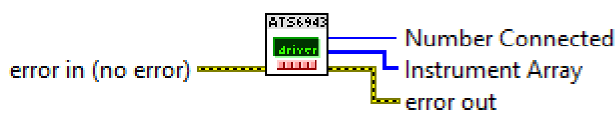
C Function Prototype Form:

ViStatus ats6943e_assignPatternSet (ViSession InstrumentHandle, ViInt32 numberOfPatterns, ViInt32 memoryOffse);

Auto Connect To DTS

LabVIEW Diagram:

ats6943e Auto Connect To DTS.vi



Description:

This vi connects to the all DTI in a system ordered by slot without resetting. Valid DTS's are separated by NULL.

Example - Seven card DTS

numberConnected returns 7.

instrumentArray[0] = Primary vi

instrumentArray[1] = Seconday vi

instrumentArray[2] = Seconday vi

instrumentArray[3] = Seconday vi

instrumentArray[4] = Seconday vi

instrumentArray[5] = Seconday vi

instrumentArray[6] = Terminator vi

Example - One 4 card DTS and two single card DTS

numberConnected returns 6.

instrumentArray[0] = Primary vi

instrumentArray[1] = Seconday vi

instrumentArray[2] = Seconday vi

instrumentArray[3] = Terminator vi

instrumentArray[4] = NULL seperator

instrumentArray[5] = Independant vi

instrumentArray[6] = NULL seperator

instrumentArray[7] = Independent vi

Key Parameters:

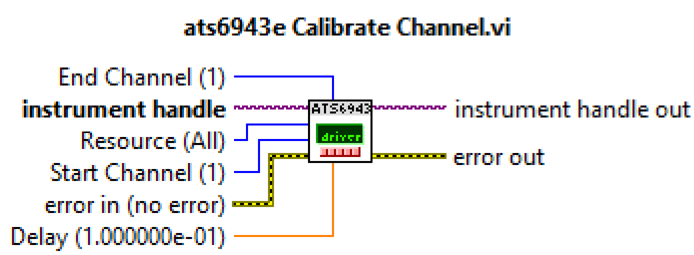
Name	Type	Description	Value
Number Connected	I16	The number of instrument handles in the array.	1 to 256
Instrument Array	[U32]	Array of instrument handles.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_autoConnectToDrs (ViSession instrumentArray[], ViInt16 arrayLength, ViPInt16 numberConnected);

Calibrate Channel

LabVIEW Diagram:



Description:

Performs calibration of the DR module resources.

Prior to calling this function, the user should allow the front-end to warm up. A warmup period is recommended so that calibration is performed at operating temperature.

The voltage reference levels must be calibrated before the measure voltage path.

The measure voltage path and DAC overlap must be calibrated before the driver levels.

The driver levels must be calibrated before the comparator levels.

The driver levels must be calibrated before the active load calibration.

Key Parameters:

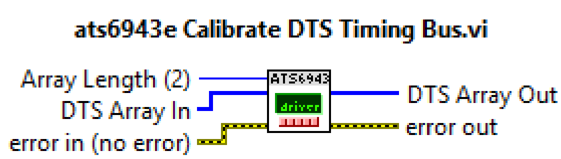
Name	Type			Description	Value
instrument handle	I/O			Identifier to a device I/O session.	0 to 2^{32}-1
Resource				Resource to calibrate.	0 – ALL 1 – ADC Gain 2 – Measure Voltage 3 – DAC Overlap 4 – PMU 5 – Programmable Gain Amplifier 6 – Driver Levels 7 – Comparator Levels 8 – Active Load 9 – Voltage Reference 10 – Load Reference
Start Channel	I16			First Channel to calibrate.	1 to 32
End Channel	I16			Last Channel to calibrate.	Start Channel to 32
Delay	DBL			If Resource set to Load Reference, then this is subtracted from the measured load. If Resource not set to Load Reference then this is the delay applied before calibration measurement for settling.	0 to 10 (Seconds/Ohms)

C Function Prototype Form:

ViStatus ats6943e_calibrateChannel (ViSession instrumentHandle, ViInt16 resource, ViInt16 startChannel, ViInt16 endChannel, ViReal64 delay_s_LoadAdjust);

Calibrate DTS Timing Bus

LabVIEW Diagram:



Description:

Calibrates the timing bus of the DTS specified by the instrument array.

The first instrument in the array must be the DTS primary module and the last instrument must be the terminating module.

Key Parameters:

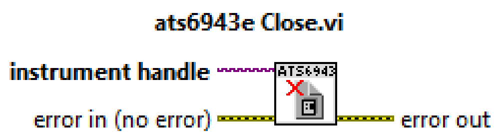
Name	Type	Description	Value
Array Length	I16	Size of the array.	2 to 13
DTS Array In	[U32]	Array of instrument handles.	0 to 2^{32}-1
DTS Array Out	[U32]	Array of instrument handles.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_calibrateDrsTimingBus ViSession instrumentArray[], ViInt16 arrayLength);

Close

LabVIEW Diagram:



Description:

This vi terminates the software connection to the PXIe 6943 and deallocates system resources associated with the session.

Parameters:

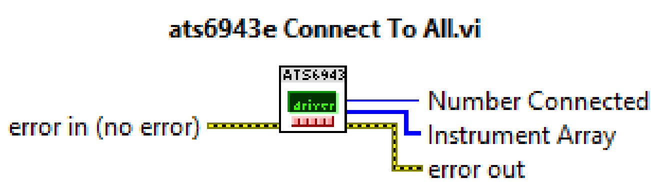
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_close (ViSession vi);

Connect to All

LabVIEW Diagram:



Description:

To establish communication with the instrument, This vi attempts to find module(s) supported by the driver in the system. It will connect to all instances of the instrument found. If no instrument of this type is found autoConnectToAll will fail.

Parameters:

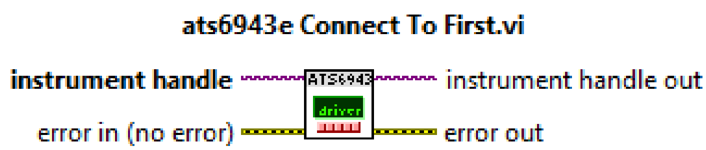
Name	Type	Description	Value
Number Connected	I16	The number of instrument handles in the array.	1 to 256
Instrument Array	[U32]	Array of instrument handles.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_autoConnectToAll (ViPSession instrumentArray, ViInt16 arrayLength, ViPInt16 numberConnected);

Connect To First

LabVIEW Diagram:



Description:

This vi searches for a module supported by the driver. It establishes communication with the first module found. If more than one module of the same type exists in the system, autoConnectToSlot may be used.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

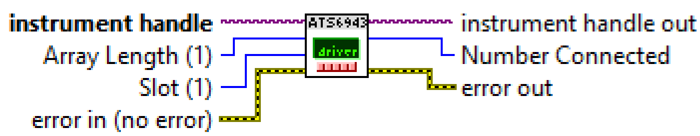
C Function Prototype Form:

ViStatus ats6943e_autoConnectToFirst (ViSession *instrumentHandle);

Connect To Slot

LabVIEW Diagram:

ats6943e Connect To Slot.vi



Description:

To establish communication with the instrument, This vi searches for a module supported by the driver. It establishes communication with the module only if it is found in the specified slot. This vi returns an array of instrument handles because multiple modules could exist in the same slot in a multi chassis system.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Array Length	I16	Size of the array.	1 to 256
Slot	I16	Slot number.	1 to 18
Number Connected	I16	The number of instrument handles in the array.	1 to 256

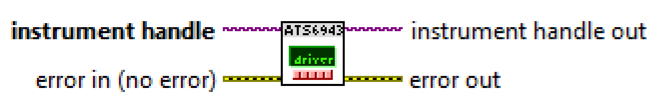
C Function Prototype Form:

ViStatus ats6943e_autoConnectToSlot (ViSession *instrumentArray, ViInt16 arrayLength, ViInt16 *numberConnected, ViInt16 slot);

Counter Initiate Trigger

LabVIEW Diagram:

ats6943e Counter Initiate Trigger.vi



Description:

This vi generates an immediate trigger for the frequency and timed totalize functions.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

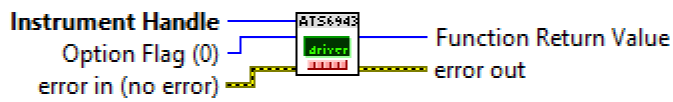
C Function Prototype Form:

ViStatus ats6943e_CounterInitiateTrigger (ViSession instrumentHandle);

Deskew DTI Channels

LabVIEW Diagram:

ats6943e Deskew DTI Channels.vi



Description:

This vi performs a channel deskew on all the channels in the DTI with closed connect relays. All channels with open connect relays will be set to default values.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Option Flag	I16	Not used.	NA

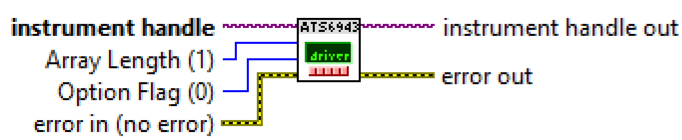
C Function Prototype Form:

ViStatus ats6943e_deskewDrmChannels (ViSession instrumentHandle, ViInt16 optionFlag);

Deskew DTS Channels

LabVIEW Diagram:

ats6943e Deskew DTS Channels.vi



Description:

Program the input conditioning of the selected channel as well as the input mode.

Parameters:

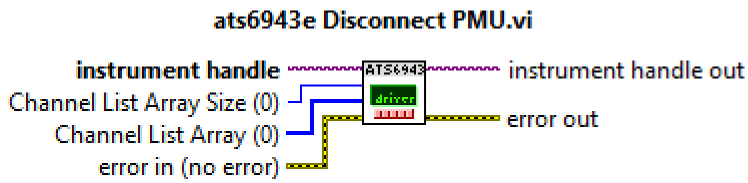
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Array Length	I16	Size of the array.	2 to 13
Option Flag	I16	Not used.	NA

C Function Prototype Form:

ViStatus ats6943e_deskewDrsChannels (ViSession *DTSArray, ViInt16 arrayLength, ViInt16 optionFlag);

Disconnect PMU

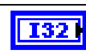
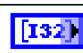
LabVIEW Diagram:



Description:

This vi disconnects the PMU of the specified channels.

Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size		The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array		Array contains the channel numbers to disconnect.	1 to 32

C Function Prototype Form:

ViStatus ats6943e_pmuDisconnect (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[]);

Error Message

LabVIEW Diagram:

Description:

This vi returns an error message for error codes specific to this instrument driver.

If the status code does not match one of the instrument specific errors than the text message will be set to "Unknown Status Error" and VI_WARN_UNKNOWN_STATUS will be returned.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Error Code	I32	This control is used to pass an instrument driver status code to the function. The function will match the status code with a corresponding text message. All the possible status codes and their associated description are listed below.	See table below
Error Message	abc	Error message corresponding to the error code.	ASCII string returned will be ≤ 2048 characters. See table below.

Table 4-94 lists the error codes returned by the driver and the associated defined constant and message.

Error Code	Defined Constant	Message
0xBFFC0001	VI_ERROR_PARAMETER1	Parameter 1 out of range
0xBFFC0002	VI_ERROR_PARAMETER2	Parameter 2 out of range
0xBFFC0003	VI_ERROR_PARAMETER3	Parameter 3 out of range
0xBFFC0004	VI_ERROR_PARAMETER4	Parameter 4 out of range
0xBFFC0005	VI_ERROR_PARAMETER5	Parameter 5 out of range
0xBFFC0006	VI_ERROR_PARAMETER6	Parameter 6 out of range
0xBFFC0007	VI_ERROR_PARAMETER7	Parameter 7 out of range
0xBFFC0008	VI_ERROR_PARAMETER8	Parameter 8 out of range
0xBFFC0011	VI_ERROR_FAIL_ID_QUERY	Identification query failed
0xBFFC0800	ATS6943E_ERROR_FILE_OPEN	Error opening file
0xBFFC0801	ATS6943E_ERROR_FILE_CLOSE	Error closing file
0xBFFC0802	ATS6943E_ERROR_FILE_WRITE	Error writing file
0xBFFC0803	ATS6943E_ERROR_FILE_READ	Error reading file
0xBFFC0804	ATS6943E_ERROR_SESSION	Invalid 6943 session pointer
0xBFFC0805	ATS6943E_ERROR_FILE_HEAD	Missing configuration file header
0xBFFC0807	ATS6943E_ERROR_ALLOC	Unable to allocate memory
0xBFFC0808	ATS6943E_ERROR_MEMORY_BUSY	Error requesting memory, memory Busy
0xBFFC0809	ATS6943E_ERROR_MEMORY_GRNT	Error requesting memory, memory not granted
0xBFFC080C	ATS6943E_ERROR_PATSETNUM	Invalid pattern set number specified
0xBFFC080D	ATS6943E_ERROR_SEQSTEPND	Sequence Step Not Initialized
0xBFFC080E	ATS6943E_ERROR_PATSETMAX	Maximum number of patterns exceeded
0xBFFC080F	ATS6943E_ERROR_PATSETOM4	Specified offset must be multiple of four
0xBFFC0810	ATS6943E_ERROR_PATDATA	Invalid pattern data code in array

Error Code	Defined Constant	Message
0xBFFC0813	ATS6943E_ERROR_BUSY	Operation not valid while sequence active
0xBFFC0814	ATS6943E_ERROR_INVCL	Invalid channel list
0xBFFC0818	ATS6943E_ERROR_FRONTEND	Installed front-end board does not support this function
0xBFFC081B	ATS6943E_ERROR_INVTM	Invalid function/setting for current timing mode
0xBFFC081C	ATS6943E_ERROR_INVCM	Invalid setting for current clock source
0xBFFC081D	ATS6943E_ERROR_CACHE	Cannot disable cache mode, cache not empty
0xBFFC081E	ATS6943E_ERROR_VSWING	Min/max voltage swing exceeded
0xBFFC0822	ATS6943E_ERROR_DCMANLOCK	Input delay DCM, sequencer A not locked
0xBFFC0825	ATS6943E_ERROR_WFDATA	Waveform transition data error
0xBFFC0826	ATS6943E_ERROR_IDTS	Invalid DTS configuration specified
0xBFFC0827	ATS6943E_ERROR_DRSCAL1	DTS calibration assert out of range
0xBFFC0828	ATS6943E_ERROR_DRSCAL2	DTS calibration delay out of range
0xBFFC0829	ATS6943E_ERROR_500CLK	500MHz Clock Not Ready
0xBFFC082A	ATS6943E_ERROR_IDANLOCK	Input delay control, sequencer A not locked
0xBFFC082C	ATS6943E_ERROR_TIMCALTO	Timing signal calibration timeout
0xBFFC082F	ATS6943E_ERROR_CNTRNRDY	Counter/Timer measurement not ready
0xBFFC0835	ATS6943E_ERROR_DRSDELAYMIN	Delay value too low error
0xBFFC0836	ATS6943E_ERROR_DRSDELAYMAX	Delay value too high error
0xBFFC0837	ATS6943E_ERROR_CNTRCHAN	Invalid counter channel setting
0xBFFC083D	ATS6943E_ERROR_PRBDM2	Settings conflict; Probe Data Mode set to Compare
0xBFFC083F	ATS6943E_ERROR_MODLINK	Settings conflict; Interconnect setting not supported
0xBFFC0840	ATS6943E_ERROR_DESKEWDELTA	Channel deskew delta exceeds maximum value
0xBFFC0841	ATS6943E_ERROR_DESKEWRANGE	Channel deskew record offset exceeds maximum value
0xBFFC0845	ATS6943E_ERROR_STATICEN	Settings conflict; Static state not enabled
0xBFFC084E	ATS6943E_ERROR_HALTMODE	Invalid halt mode setting
0xBFFC0850	ATS6943E_ERROR_STATICTO	Static data execution timeout
0xBFFC0851	ATS6943E_ERROR_ERRPDEL	Error pulse delay calibration failure
0xBFFC085A	ATS6943E_ERROR_PONVERR	VCC or VEE power failure
0xBFFC085B	ATS6943E_ERROR_PONHVERR	HV_VCC power failure
0xBFFC085C	ATS6943E_ERROR_PONOT	Over temperature flag set
0xBFFC085D	ATS6943E_ERROR_PONVE1	VEE1 voltage level error
0xBFFC085E	ATS6943E_ERROR_PONVE2	VEE2 voltage level error
0xBFFC085F	ATS6943E_ERROR_PONVI	Voltage monitor IIC interface error
0xBFFC0860	ATS6943E_ERROR_PON12	+12V voltage or current level error

Error Code	Defined Constant	Message
0xBFFC0861	ATS6943E_ERROR_PON5	+5V voltage or current level error
0xBFFC0862	ATS6943E_ERROR_PONVC1	VCC1 voltage or current level error
0xBFFC0863	ATS6943E_ERROR_PONVC2	VCC2 voltage or current level error
0xBFFC0864	ATS6943E_ERROR_PON	Power on failure
0xBFFC0865	ATS6943E_ERROR_PMULOAD	Settings conflict; PMU or LOAD enabled
0xBFFC0866	ATS6943E_ERROR_MONTO	Monitor update command timeout
0xBFFC0867	ATS6943E_ERROR_FUNCAL	Settings conflict; Active load only allowed in Dynamic HiZ function
0xBFFC0868	ATS6943E_ERROR_HEADER	Missing FPGA file header
0xBFFC0869	ATS6943E_ERROR_FPGALOAD	Loading new FPGA failed
0xBFFC086A	ATS6943E_ERROR_NVUPDATE	NVM update invalid password
0xBFFC086B	ATS6943E_ERROR_CFUNCSC	Channel function setting conflict
0xBFFC086C	ATS6943E_ERROR_SMUINIT	Error initializing SMU for calibration
0xBFFC086D	ATS6943E_ERROR_VREF1	+5V voltage reference out of range
0xBFFC086E	ATS6943E_ERROR_VREF2	+3.33V voltage reference out of range
0xBFFC086F	ATS6943E_ERROR_VREF3	+1.66V voltage reference out of range
0xBFFC0870	ATS6943E_ERROR_VREF4	-1V voltage reference out of range
0xBFFC0871	ATS6943E_ERROR_FICAL	Force current calibration out of range
0xBFFC0872	ATS6943E_ERROR_LREF1	50 ohm reference out of range
0xBFFC0873	ATS6943E_ERROR_LREF2	10K ohm reference out of range
0xBFFC0874	ATS6943E_ERROR_SMUDATA	Error configuring SMU for calibration
0xBFFC0875	ATS6943E_ERROR_TEMPENV	Unable to get TEMP environment variable.
0xBFFC0876	ATS6943E_ERROR_LEGACY	Legacy setting not supported.
0x3FFC0800	ATS6943E_WARN_PATMOREDATA	More pattern data than pin list data
0x3FFC0802	ATS6943E_WARN_CLVALUESNEQ	Channel list query data mixed
0x3FFC0803	ATS6943E_WARN_MOREDATA	More data available
0x3FFC0804	ATS6943E_WARN_FELOAD	Front-end module data not loaded due to incompatible type
0x3FFC0806	ATS6943E_WARN_OG	Offset/Gain DAC values out of range
0x3FFC080E	ATS6943E_WARN_FEID	Unknown Front-end Module
0x3FFC0810	ATS6943E_WARN_DRMCAL	DRM Auto calibration out of range
0x3FFC0811	ATS6943E_WARN_DESKEW	Channel deskew exceeds maximum delay
0x3FFC0812	ATS6943E_WARN_MIC	Invalid module inter connect setting front panel timing bus
0x3FFC0813	ATS6943E_WARN_MICLC	Invalid module inter connect setting from configuration
0x3FFC0817	ATS6943E_WARN_CNTRRES	Measurement accuracy exceeds 0.1%
0x3FFC0818	ATS6943E_WARN_CNTROF	Counter/Timer overflow

Error Code	Defined Constant	Message
0x3FFC0819	ATS6943E_WARN_FENS	Installed front-end board does not support current setting(s)
0x3FFC081A	ATS6943E_WARN_DRMTEST	DRM calibration failure
0x3FFC0820	ATS6943E_WARN_PLCONV	Invalid pipeline conversion from mask to depth.
0x3FFC0822	ATS6943E_WARN_OGUTP	Calibration values out of range.
0x3FFC0829	ATS6943E_WARN_SIMULATION_ON	WARNING: Simulation mode is presently active. The operations and data returned are simulated
0x3FFC082A	ATS6943E_WARN_FUNCTION_STUB	WARNING: The function/setting is a stub for legacy support and has no effect on the operation of the instrument
0x3FFC082B	ATS6943E_WARN_POST	POST error detected
1	VI_MORE_INST_PRESENT	More 6943 modules are present
0x3FFC0104	VI_WARN_NSUP_ERROR_QUERY	

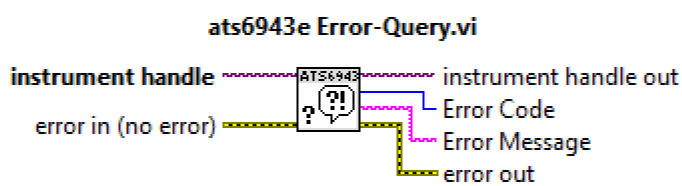
Table 4-94 Error Codes and Messages

C Function Prototype Form:

ViStatus ats6943e_error_message (ViSession instrumentHandle, ViStatus statusCode, ViChar message[]);

Error-Query

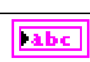
LabVIEW Diagram:



Description:

This vi is not supported by the PXIe 6943.

Parameters:

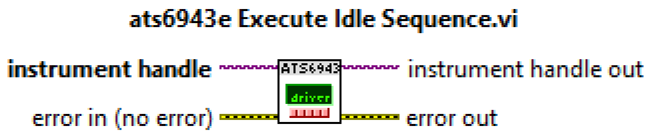
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Error Code		Returned error code.	0x3FFC0104 Error query not supported
Error Message		Error message.	Empty string (“”)

C Function Prototype Form:

ViStatus ats6943e_error_query (ViSession instrumentHandle, ViInt32 *errorCode, ViChar errorMessage[]);

Execute Idle Sequence

LabVIEW Diagram:



Description:

This vi executes the idle sequence.

DTS Operation:

Primary only. Coupled DTIs must execute **Arm Idle Sequence** before executing this function.

Parameters:

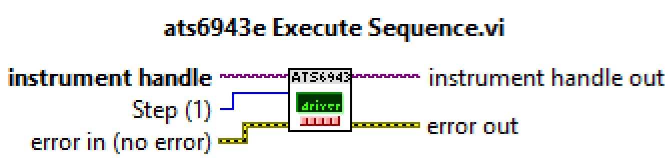
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_executeIdleSequence (ViSession instrumentHandle);

Execute Sequence

LabVIEW Diagram:



Description:

This vi executes the specified sequence.

DTS Operation:

Primary only. Coupled DTIs must execute **Arm Sequence** before executing this function.

Parameters:

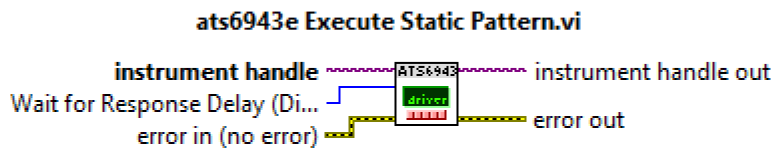
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Step	I16	Sequence step number to arm.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_executeSequence (ViSession instrumentHandle, ViInt16 step);

Execute Static Pattern

LabVIEW Diagram:



Description:

This vi executes a static pattern.

When the pattern is executed, the channels configured as static will be driven to the specified state and then sampled when the response delay has completed.

If a previous static execution has not completed, This vi will wait up to 10ms for it to complete or timeout.

If the wait for response delay flag is enabled, This vi will wait up to 10ms after execution for the response data to complete or timeout..

Parameters:

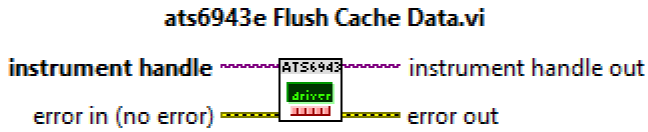
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Wait for Response Delay		This control is used to enable/disable the wait for response delay.	0 = Disabled 1 = Enabled

C Function Prototype Form:

ViStatus ats6943e_executeStaticPattern (ViSession instrumentHandle, ViInt16 waitForResponseDelay);

Flush Cache Data

LabVIEW Diagram:



Description:

This vi flushes any cache data that is waiting to be written to the DTI.

Parameters:

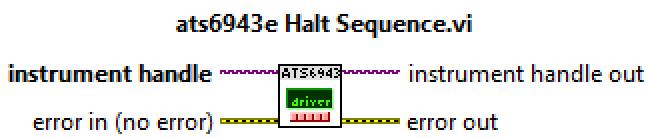
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_flushCacheData (ViSession instrumentHandle);

Halt Sequence

LabVIEW Diagram:



Description:

This vi generates a halt command to the specified sequencer.

The halt mode must be set prior to calling **Set Halt Mode**.

Parameters:

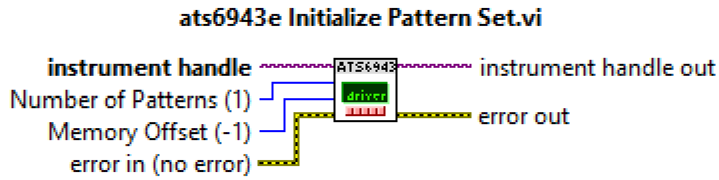
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_haltSequence (ViSession instrumentHandle);

Initialize Pattern Set

LabVIEW Diagram:



Description:

This vi initializes a block of external memory called a pattern set that is associated with the selected pattern step using **Select Sequence Step**.

A pattern set consists of one or more consecutive memory locations called a pattern (262144 maximum).

Each pattern contains a code for each channel that determines its input/output operation. The Fourteen codes are:

1. Disable channel
2. Collect CRC
3. Drive Low
4. Drive High
5. Repeat Previous Code
6. Invert Previous Code
7. Expect Valid Low
8. Expect Valid High
9. Expect Valid
10. Expect Between
11. Drive Low, Expect Low
12. Drive High, Expect High
13. Drive Low, Expect High
14. Drive High, Expect Low

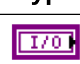
All channel are set to "Repeat Previous Code" by this function.

Once initialized, the pattern set can be programmed using **Load Pattern Memory** and/or **Set Pattern Data**.

DTS Operation:

All coupled modules sequence steps should be programmed with the same numbers of patterns in order to remain synchronized with each other.

Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

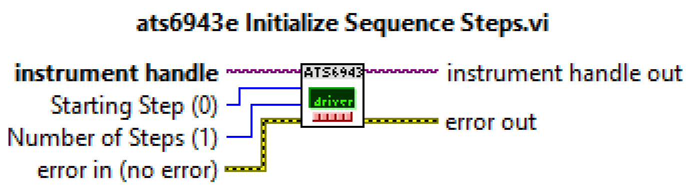
Name	Type	Description	Value
Number of Patterns	I32	This control is used to set the number of patterns to create in the pattern set.	1 to 262144
Memory Offset	I32	This control is used to set the memory offset of the pattern set in the external memory.	-1 to 262140 The memory offset must be a multiple of 4. Passing a -1 specifies the next available memory offset will be assigned.

C Function Prototype Form:

ViStatus ats6943e_initPatternSet (ViSession instrumentHandle, ViInt32 numberOfPatterns, ViInt32 memoryOffset);

Initialize Sequence Steps

LabVIEW Diagram:



Description:

This vi initializes the sequence step logic to the following defaults:

1. All Phases set to Assert 0 and Return 12 if non-indexed mode.
2. All Windows set to Open 0, Close 12 if non-indexed mode.
3. System Clock Period set to 100.
4. CPP set to 1.
5. Jump set to None.
6. Loop Counter and Count set to 0.
7. Last Step set true.
8. Sequence Timeout set to Reset.
9. Sequence Flags set low.
10. Record mode set to Record Errors.
11. Waveforms disabled.
12. Phase Trigger set to System Clock.
13. Handshake Enable and Modifier set to None.
14. Pattern Set marked as not assigned.

If the timing mode is set to indexed, the timing set number and the timing set settings are not modified.0

DTS Operation:

All coupled modules sequence steps should be programmed with the same

CPP, Last Step, Jump Type and Jump Step in order to remain synchronized with each other.

Parameters:

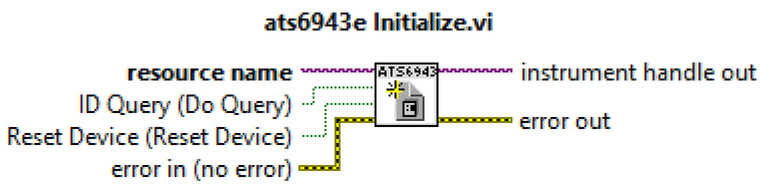
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Starting Step	I16	Starting step number to initialize.	0 to 4095
Number of Steps	I16	Number of steps to initialize.	1 to 4096

C Function Prototype Form:

ViStatus ats6943e_initSequenceSteps (ViSession instrumentHandle, ViInt16 startingStep, ViInt16 numberOfSteps);

Initialize

LabVIEW Diagram:



moni

Description:

This vi is used to establish a communication connection with a DTI. It returns a unique session pointer that is used by other vi's to identify a specific DTI. After a valid communication link is established the user can additionally perform the following options:

1. Reset to power on values.
2. Verify using register ID query.

Parameters:

Name	Type	Description	Value
resource name	I/O	Specifies with which remote instrument to establish a communication session.	Input String
ID Query	TF	This control specifies whether to perform an identification query after connecting.	0=No 1=Yes
Reset Device	TF	This control specifies whether to perform a reset after connecting.	0=No 1=Yes
instrument handle out	I/O	Identifier to a device I/O session	0 to 232-1

Based on the syntax of the Resource Name, the Initialize function configures the I/O interface and generates an Instrument Handle. Optional parameters are shown in square brackets ([]).

PXI[bus]::device[::function][::INSTR]

Bus and device numbers can be viewed from the resource manager display.

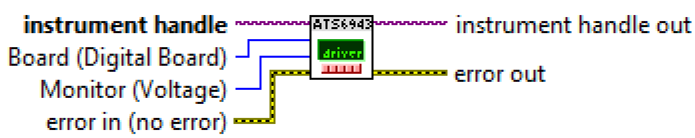
C Function Prototype Form:

ViStatus ats6943e_init (ViRsrc resourceName, ViBoolean IDQuery, ViBoolean resetDevice, ViSession *instrumentHandle);

Initiate Monitor

LabVIEW Diagram:

ats6943e InitiateMonitor.vi

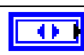
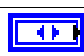


Description:

This vi initiates a board monitor sequence. The DB has voltage and temperature monitors. The DR board only has voltage monitors.

This vi should be called prior to calling **Query Voltage Monitor** or **Query Temperature Monitor**.

Parameters:

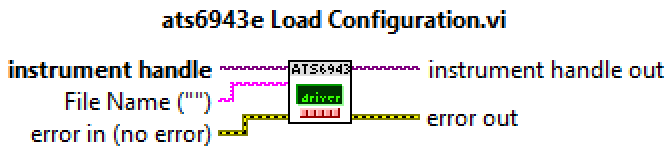
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Board		This control specifies which board monitor to initiate.	0 = Digital Board 1 = Driver Receiver
Monitor		This control specifies which monitor to initiate.	0 = Voltage 1 = Temperature

C Function Prototype Form:

ViStatus ats6943e_initiateMonitor (ViSession instrumentHandle, ViInt16 board, ViInt16 monitor);

Load Configuration

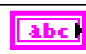
LabVIEW Diagram:



Description:

This vi loads a configuration file that was saved using **Save Configuration**

Parameters:

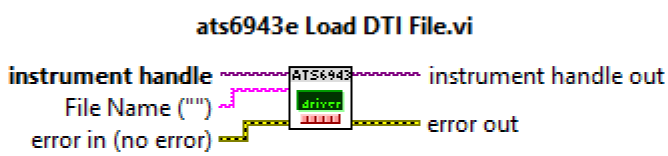
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
File Name		This control specifies the file to load.	String Input

C Function Prototype Form:

ViStatus ats6943e_loadConfiguration (ViSession instrumentHandle, ViChar fileName[]);

Load DTI File

LabVIEW Diagram:



Description:

This vi loads the serial numbers and ASSY revisions from a file.

Parameters:

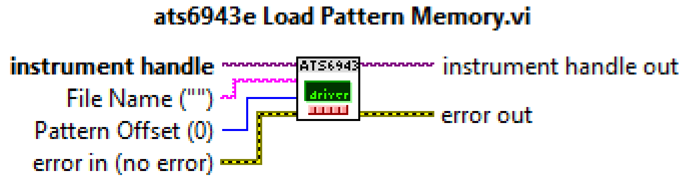
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
File Name		This control specifies the file to load.	String Input

C Function Prototype Form:

ViStatus ats6943e_loadDrmFile (ViSession instrumentHandle, ViChar fileName[]);

Load Pattern Memory

LabVIEW Diagram:



Description:

This vi loads the DTI pattern memory from the file name specified.

The header contains the number of patterns and the format. The format of the header is:

[ATS6943 PAT DUMP <dd> <nnnnnn>]

where:

<dd> is the format;

00 = Pattern data as ASCII Hex

01 = Pattern data as ASCII String

02 = Pattern data as Binary

03 = Pattern data and flags as ASCII Hex

04 = Pattern data and flags as ASCII String

05 = Pattern data and flags as Binary

<nnnnnn> is the number of patterns..

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
File Name	abc	This control specifies the file to load.	String Input
Pattern Offset	I32	This control is used to set the memory offset of the pattern set in the external memory.	0 to 262143

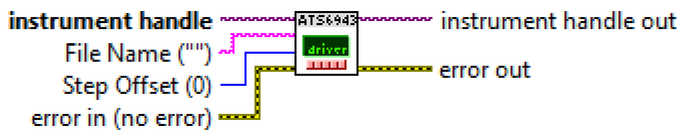
C Function Prototype Form:

ViStatus ats6943e_loadPatternMemory (ViSession instrumentHandle, ViChar fileName[], ViInt32 patternOffset);

Load Sequence Memory

LabVIEW Diagram:

ats6943e Load Sequence Memory.vi



Description:

This vi loads the DTI sequence memory from the file name specified.

The header contains the number of steps and the format. The format of the header is:

[ATS6943E SEQ DUMP <dd> <nnnn>]

where:

<dd> is the format;

0 = ASCII Hex.

1 = Binary

<nnnn> is the number of steps.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
File Name	abc	This control specifies the file to load.	String Input
Step Offset	I16	This control specifies the step offset where the first sequence will be loaded.	0 to 4095

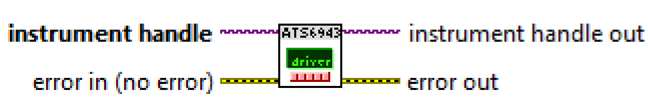
C Function Prototype Form:

ViStatus ats6943e_loadSequenceMemory (ViSession instrumentHandle, ViChar fileName[], ViInt16 stepOffset);

Master Reset Sequence

LabVIEW Diagram:

ats6943e Master Reset Sequence.vi



Description:

This vi performs a master reset on the sequence logic.

DTS Operation:

Primary only if "Master Reset" assigned to a common TTLT trigger.

All coupled modules if "Master Reset" not assigned to a common TTL Bus trigger.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

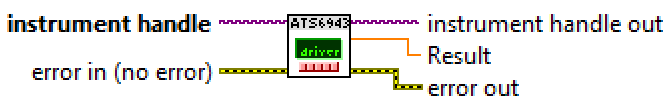
C Function Prototype Form:

ViStatus ats6943e_masterResetSequence (ViSession instrumentHandle);

Measure Counter Result

LabVIEW Diagram:

ats6943e Measure Counter Results.vi



Description:

This vi returns the results of the selected counter function.

Parameters:

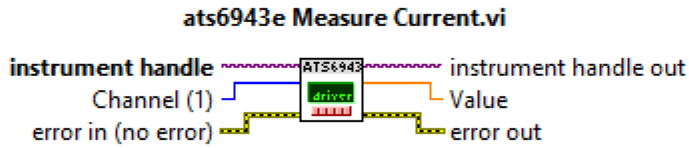
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Result	DBL	This control returns the most recent result from the counter.	-Inf to +Inf

C Function Prototype Form:

ViStatus ats6943e_measureCounterResults (ViSession instrumentHandle, ViReal64 *result);

Measure Current

LabVIEW Diagram:



Description:

This vi measures the currrent at the driver/receiver pin. Only channels set to PMU FV can measure the current.

Parameters:

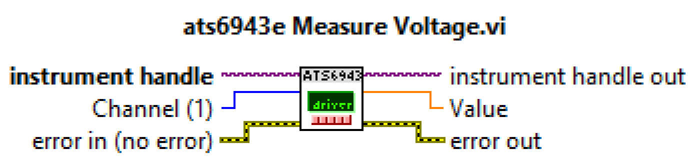
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel	I16	Channel selection.	1 to 32
Value	DBL	Measure result.	-0.1 to 0.1

C Function Prototype Form:

ViStatus ats6943e_pmuMeasureCurrent (ViSession instrumentHandle, ViInt16 channel, ViReal64 *value);

Measure Voltage

LabVIEW Diagram:



Description:

This vi measures the voltage at the driver/receiver pin.

Parameters:

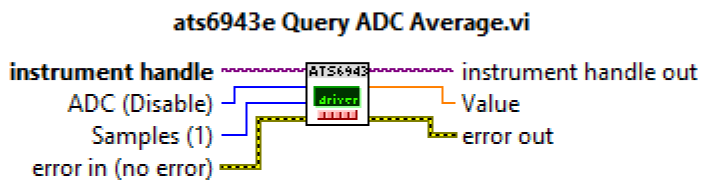
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel	I16	Channel selection.	1 to 32
Value	DBL	Measure result.	-2.0 to +7.0

C Function Prototype Form:

ViStatus ats6943e_pmuMeasureVoltage (ViSession instrumentHandle, ViInt16 channel, ViReal64 *value);

Query ADC Average

LabVIEW Diagram:



Description:

This vi returns the value of the selected ADC input averaged by the specified "Samples" parameter.

The ADC has eight inputs:

1. Driver/Receiver monitor.
   The specific monitor signal is selected using **Set Monitor Signal**.
2. VEE1 Current Monitor
3. Programmable load
4. E_S Mux (Reference Voltage)
   a. Mux disabled
   b. Mux set to +3.33V
   c. Mux set to +1.66V
   d. Mux set to +5.0V
   e. Mux set to -1.0V
5. E_F Pin
6. VEE2 Current Monitor
7. Driver/Receiver Voltage Mux
   a. VEE1
   b. VEE2
   c. HV_VCC1
   d. HV_VCC2
8. DUTGND

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
ADC		ADC input	-1 = ADC disabled 0 = Driver/Receiver Monitor

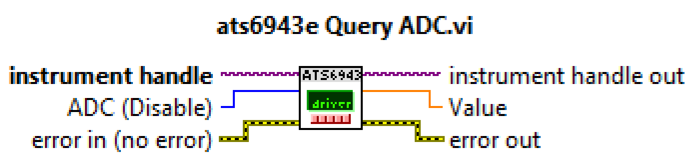
			1 = VEE1 Current 2 = Programmable Load 3 = E_S Pin Mux Off 4 = E_S Pin Mux +3.3V 5 = E_S Pin Mux +1.66V 6 = E_S Pin Mux +5V 7 = E_S Pin Mux -1V 8 = E_F Pin 9 = VEE2 Current 10 = VEE1 Voltage 11 = VEE2 Voltage 12 = HV_VCC1 Voltage 13 = HV_VCC2 Voltage 14 = DUTGND
Samples	I16	Number of samples to average	1 to 32767
Value	DBL	ADC result	-5.0 to +10.0

C Function Prototype Form:

ViStatus ats6943e_queryAdcAverage (ViSession instrumentHandle, ViInt16 ADC, ViInt16 samples, ViReal64 *value);

Query ADC

LabVIEW Diagram:



Description:

This vi returns the value of the selected ADC input.

The ADC has eight inputs:

1. Driver/Receiver monitor.
   The specific monitor signal is selected using **Set Monitor Signal**.
2. VEE1 Current Monitor
3. Programmable load
4. E_S Mux (Reference Voltage)
   f. Mux disabled
   g. Mux set to +3.33V
   h. Mux set to +1.66V
   i. Mux set to +5.0V
   j. Mux set to -1.0V
5. E_F Pin
6. VEE2 Current Monitor

7. Driver/Receiver Voltage Mux
    e. VEE1
    f. VEE2
    g. HV_VCC1
    h. HV_VCC2
8. DUTGND

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
ADC		ADC input	-1 = ADC disabled 0 = Driver/Receiver Monitor 1 = VEE1 Current 2 = Programmable Load 3 = E_S Pin Mux Off 4 = E_S Pin Mux +3.3V 5 = E_S Pin Mux +1.66V 6 = E_S Pin Mux +5V 7 = E_S Pin Mux -1V 8 = E_F Pin 9 = VEE2 Current 10 = VEE1 Voltage 11 = VEE2 Voltage 12 = HV_VCC1 Voltage 13 = HV_VCC2 Voltage 14 = DUTGND
Value	DBL	ADC result	-5.0 to +10.0

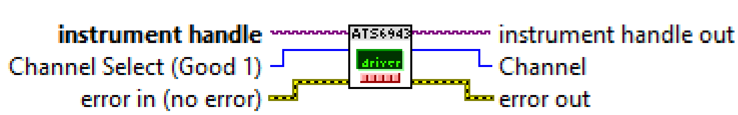
C Function Prototype Form:

ViStatus ats6943e_queryAdc (ViSession instrumentHandle, ViInt16 ADC, ViReal64 *value);

Query Aux Channel Select

LabVIEW Diagram:

ats6943e Query Aux Channel Select.vi



Description:

Query the aperture time.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel Select		This control specifies which channel select to query.	0 = Good 1 1 = Good 0
Channel	I16	This control returns the channel assigned to the select signal.	1 to 32

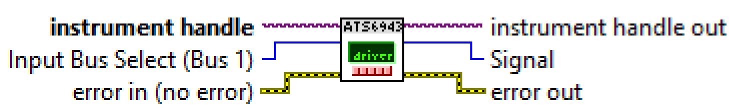
C Function Prototype Form:

ViStatus ats6943e_queryAuxChannelSelect (ViSession instrumentHandle, ViInt16 channelSelect, ViInt16 *channel);

Query Aux Input Bus Select

LabVIEW Diagram:

ats6943e Query Aux Input Bus Select.vi



Description:

This vi returns the signal source of the specified bus select signal.

Refer to **Set Aux Input Bus Select** for returned values.

Parameters:

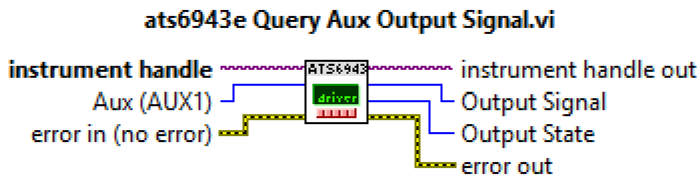
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Input Bus Select		This control specifies which input bus to query.	0 = Bus 0 1 = Bus 1 2 = Bus 2 3 = Bus 3
Signal	I16	This control returns the signal assigned to the input bus.	See description above

C Function Prototype Form:

ViStatus ats6943e_queryAuxInputBusSelect (ViSession instrumentHandle, ViInt16 inputBusSelect, ViInt16 *signal);

Query Aux Output Signal

LabVIEW Diagram:



Description:

This vi returns the output signal and output state of the specified auxiliary signal.

Refer to **Set Aux Input Bus Select** for returned values.

Parameters:

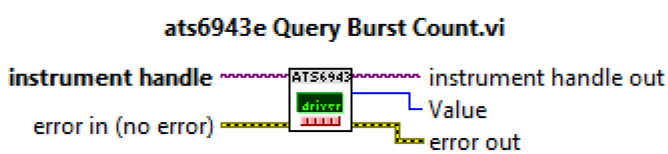
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Aux		This control specifies which AUX signal to query.	0 = AUX1 . . 11 = AUX12
Output Signal	I16	This control returns the signal assigned to the AUX signal.	See description above
Output State	I16	This control returns the output state of the specified aux signal.	See description above

C Function Prototype Form:

ViStatus ats6943e_queryAuxOutputSignal (ViSession instrumentHandle, ViInt16 aux, ViInt16 *outputSignal, ViInt16 *outputState);

Query Burst Count

LabVIEW Diagram:



Description:

This vi returns the sequence burst count.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

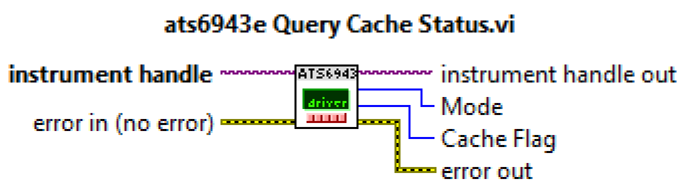
Value	I16	This control is used to return the burst count value.	0 to 1048576

C Function Prototype Form:

ViStatus ats6943e_queryBurstCount (ViSession instrumentHandle, ViInt32 *value);

Query Cache Status

LabVIEW Diagram:



Description:

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Mode	I16	This control is used to return the cache mode.	0 = Disabled 1 = Enabled
Cache Flag	I16	This control is used to return the cache flag.	0 to 127

The cache flag indicates if there is cached data. A zero indicates that there is no cached data. A one in the specified bit position indicates that cached data is present.

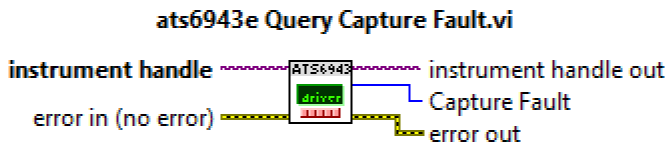
Bit	Memory
0	Pattern Data Channels 1-8
1	Pattern Data Channels 9-16
2	Pattern Data Channels 17-24
3	Pattern Data Channels 25-32
4	Sequence Data
5	Pattern Flag Data
6	Timing Set Data

C Function Prototype Form:

ViStatus ats6943e_queryCacheStatus (ViSession instrumentHandle, ViInt16 *mode, ViInt16 *cacheFlag);

Query Capture Fault

LabVIEW Diagram:



Description:

This vi returns the capture fault register.

A capture fault is generated when an expect pattern code is programmed and the capture mode is set to "None" or the window setting is not valid.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Capture Fault	U32	This control is used to return the capture fault register.	0 to FFFFFFFF hex

A capture fault indicates that an expect pattern code was executed but either the capture mode or window was invalid.

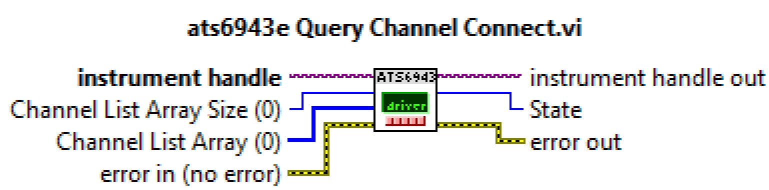
Bit 0 is mapped to channel one and Bit 31 is mapped to channel 32. A one indicates that a capture fault occurred on the corresponding channel.

C Function Prototype Form:

ViStatus ats6943e_queryCaptureFault (ViSession instrumentHandle, ViUInt32 *captureFault);

Query Channel Connect

LabVIEW Diagram:



Description:

This vi queries the front panel channel connect state.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
State	I16	This control is used to return the channel list connect state.	0 = Open 1 = Closed

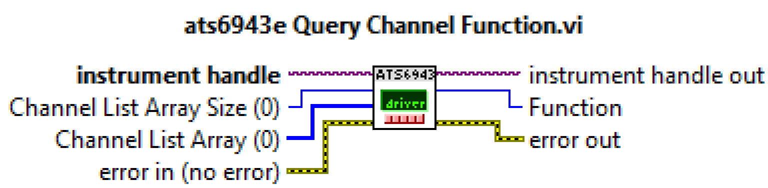
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same state.

C Function Prototype Form:

ViStatus ats6943e_queryChannelConnect (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *state);

Query Channel Function

LabVIEW Diagram:



Description:

This vi returns the channel function setting.

Refer to **Set Channel Function** for returned values.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query	1 to 32
Function	I16	This control is used to return the channel list connect state.	See description above

This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same function.

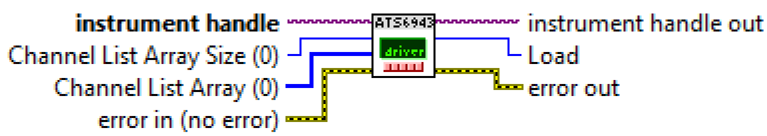
C Function Prototype Form:

ViStatus ats6943e_queryChannelFunction (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *function);

Query Channel Load State

LabVIEW Diagram:

ats6943e Query Channel Load State.vi



Description:

This vi queries the front panel channel load state.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Load	I16	This control is used to return the channel sense load state.	0 = Load off 1 = Load on 2 = Load on when output disabled

This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same load state.

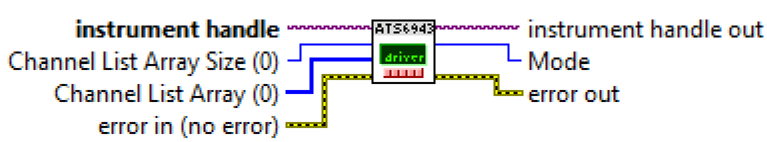
C Function Prototype Form:

ViStatus ats6943e_queryChannelLoadState (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *load);

Query Channel Mode

LabVIEW Diagram:

ats6943e Query Channel Mode.vi



Description:

This vi returns the front panel channel mode setting.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Mode	I16	This control is used to return the channel mode setting.	0 = Single Ended 1 = Differential

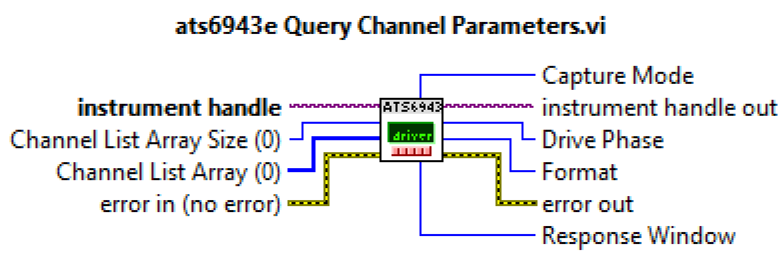
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same channel mode.

C Function Prototype Form:

ViStatus ats6943e_queryChannelMode (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *mode);

Query Channel Parameters

LabVIEW Diagram:



Description:

This vi returns the channel parameter settings.

Refer to **Set Channel Parameters** for returned values.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Capture Mode	I16	This control is used to return the channel mode setting.	See description above

Drive Phase	I16	This control is used to return the channel mode setting.	See description above
Format	I16	This control is used to return the channel mode setting.	See description above
Response Window	I16	This control is used to return the channel mode setting.	See description above

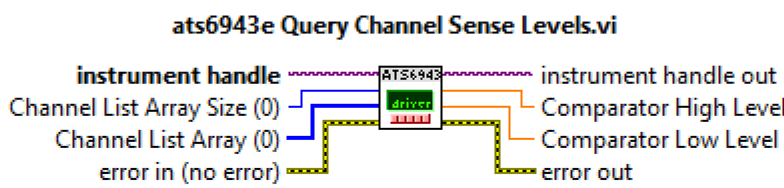
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same parameters.

C Function Prototype Form:

ViStatus ats6943e_queryChannelParameters (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt16 channelListArray[], ViInt16 *drivePhase, ViInt16 *format, ViInt16 *responseWindow, ViInt16 *captureMode);

Query Channel Sense Levels

LabVIEW Diagram:



Description:

This vi queries the front panel channel sense levels.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Comparator High Level	DBL	This control returns the high comparator level (CVH).	-2.0 to +7.0
Comparator Low Level	DBL	This control returns the low comparator level (CVL).	-2.0 to +7.0

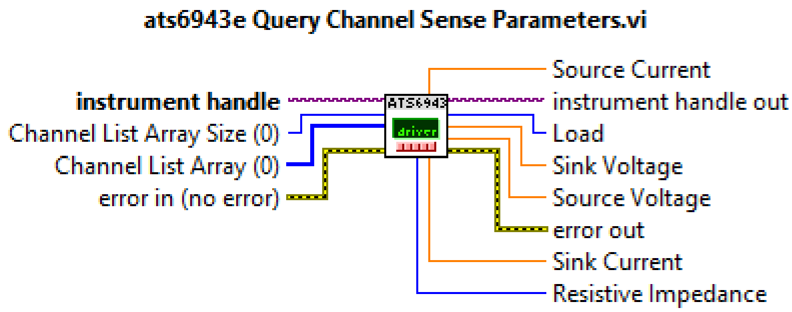
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same sense levels.

C Function Prototype Form:

ViStatus ats6943e_queryChannelSenseLevels (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 *comparatorHighLevel_CVH, ViReal64 *comparatorLowLevel_CVL);

Query Channel Sense Parameters

LabVIEW Diagram:



Description:

This vi returns the front panel channel sense parameters.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Source Current	DBL	This control returns the source current.	-2.0 to +7.0
Load	I16	This control is used to return the channel sense load configuration.	0 = Split Load 1 = Single Load
Sink Voltage	DBL	This control returns the Sink Voltage.	-2.0 to +7.0
Source Voltage	DBL	This control returns the Source Voltage.	-2.0 to +7.0
Sink Current	DBL	This control returns the Sink Current.	0 to 24 mA
Resistive Impedance	I16	This control is included for compatibility and is not used. A zero will be returned.	0

This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same sense parameters.

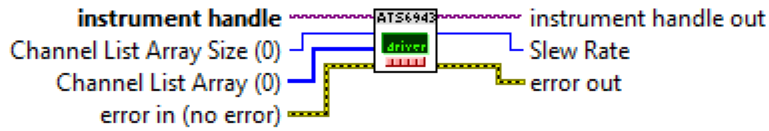
C Function Prototype Form:

ViStatus ats6943e_queryChannelSenseParameters (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *load, ViReal64 *sinkVoltage, ViReal64 *sourceVoltage, ViReal64 *sinkCurrent_mA, ViReal64 *sourceCurrent_mA, ViInt16 *resistiveImpedance);

Query Channel Slew Rate

LabVIEW Diagram:

ats6943e Query Channel Slew Rate.vi



Description:

This vi queries the output channel slew rate setting.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Slew Rate	I16	This control is used to return the channel(s) slew rate setting.	0 = Fast (1.3V/ns) 1 = Medium (1.0V/ns) 2 = Default (0.7V/ns) 3 = Slow (0.2V/ns) 4 = User 5 = Low Power

This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same slew rate.

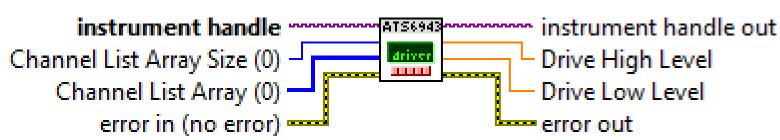
C Function Prototype Form:

ViStatus ats6943e_queryChannelSlewRate (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *slewRate);

Query Channel Source Levels

LabVIEW Diagram:

ats6943e Query Channel Source Levels.vi



Description:

This vi returns the front panel channel source levels.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Drive High Level	DBL	This control returns the high drive level (DVH).	-2.0 to +7.0
Drive Low Level	DBL	This control returns the low drive level (DVL).	-2.0 to +7.0

This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same driver levels.

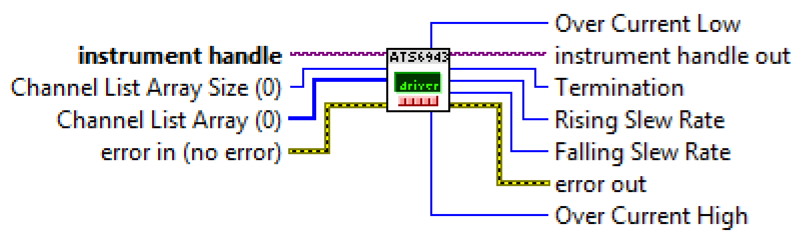
C Function Prototype Form:

ViStatus ats6943e_queryChannelSourceLevels (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 *driveHighLevel_DVH, ViReal64 *driveLowLevel_DVL);

Query Channel Source Parameters

LabVIEW Diagram:

ats6943e Query Channel Source Parameters.vi



Description:

Query the specified module data from volatile memory.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32

Name	Type	Description	Value
Over Current Low	I16	This control is included for compatibility and is not used. A zero will be returned.	0
Termination	I16	This control is used to return the channel source termination value in ohms.	35 to 66
Rising Slew Rate	I16	This control returns the rising edge slew rate.	0 to 255
Falling Slew Rate	I16	This control returns the falling edge slew rate.	0 to 255
Over Current High	I16	This control is included for compatibility and is not used. A zero will be returned.	0

For the rising and falling slew rates, the first three bits are a fine adjust setting and the next five bits are a course adjust. The fine adjust goes from -40% to +30% of coarse adjust in 10% increments.

0 = -40% of coarse adjust

7 = +30% of coarse adjust

The coarse adjust settings are:

0 = ~0.25V/ns

31 = ~1.5V/ns

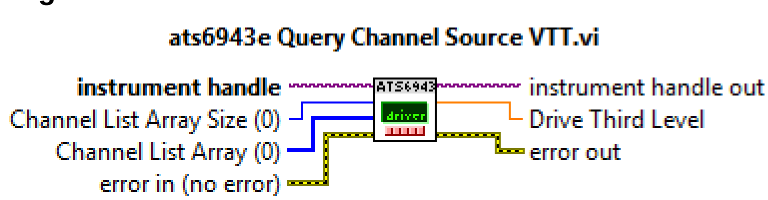
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same source parameters settings.

C Function Prototype Form:

ViStatus ats6943e_queryChannelSourceParameters (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *termination, ViInt16 *risingSlewRate, ViInt16 *fallingSlewRate, ViInt16 *overCurrentHigh, ViInt16 *overCurrentLow);

Query Channel Source VTT

LabVIEW Diagram:



Description:

This vi returns the front panel channel source VTT level.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
VTT Level	DBL	This control returns the VTT drive level.	-2.0 to +7.0

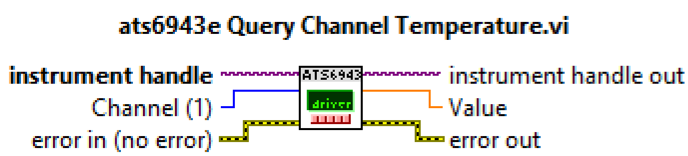
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same VTT level.

C Function Prototype Form:

ViStatus ats6943e_queryChannelSourceVtt (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 *driveThirdLevel_VTT);

Query Channel Temperature

LabVIEW Diagram:



Description:

This vi returns the channel temperature.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel	I16	This control specifies which channel temperature to query.	1 to 32
Value	DBL	This control return the channel temperature.	0.0 to 140.0

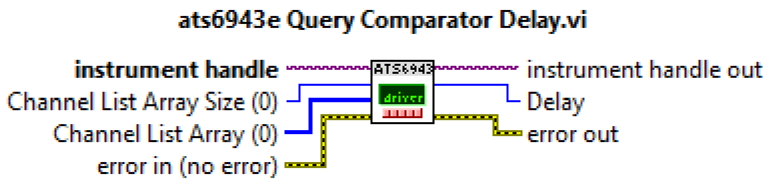
Power must be connected to query the channel temperature, see **Set Power Connect**.

C Function Prototype Form:

ViStatus ats6943e_queryChannelTemp (ViSession instrumentHandle, ViInt16 channel, ViReal64 *value);

Query Comparator Delay

LabVIEW Diagram:



Description:

This vi returns the front panel channel comparator delay.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Delay	I16	This control is used to return the channel(s) comparator delay setting.	0 to 1023

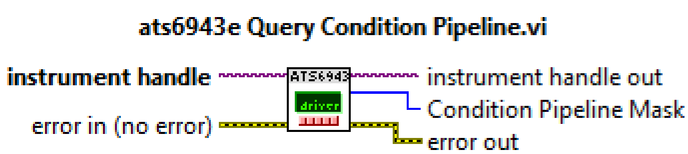
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same comparator delay value.

C Function Prototype Form:

ViStatus ats6943e_queryComparatorDelay (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *delay);

Query Condition Pipeline

LabVIEW Diagram:



Description:

This vi returns the condition pipeline setting.

Parameters:

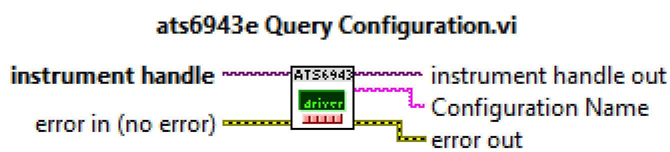
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Condition Pipeline Mask	U32	This control is used to return the condition pipeline setting.	0 to 16

C Function Prototype Form:

ViStatus ats6943e_queryConditionPipeline (ViSession instrumentHandle, ViUInt32 *conditionPipeline);

Query Configuration

LabVIEW Diagram:



Description:

This vi returns the last configuration file name and path loaded or saved.

Parameters:

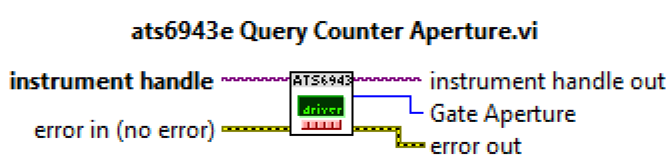
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Configuration Name	abc	This control returns the current configuration name.	ASCII string returned will be <= 384 characters.

C Function Prototype Form:

ViStatus ats6943e_queryConfiguration (ViSession instrumentHandle, ViChar configurationName[]);

Query Counter Aperture

LabVIEW Diagram:



Description:

This vi returns the gate aperture time for the frequency and timed totalize functions.

Refer to **Set Counter Aperture** for returned values.

Parameters:

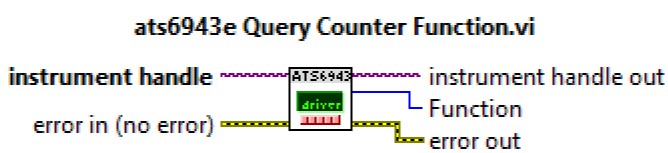
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Gate Aperture	I16	This control is used to return the gate aperture for the frequency and timed totalize functions.	See description above

C Function Prototype Form:

ViStatus ats6943e_queryCounterAperture (ViSession instrumentHandle, ViInt16 *gateAperture);

Query Counter Function

LabVIEW Diagram:



Description:

This vi returns the counter function.

Refer to **Set Counter Function** for returned values.

Parameters:

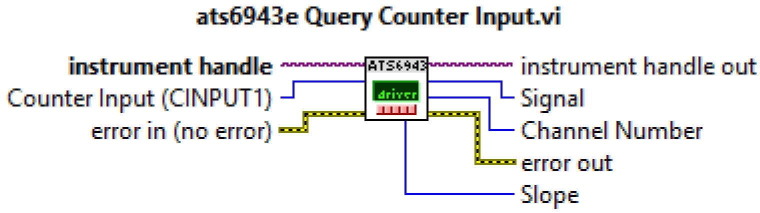
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Function	I16	This control is used to return the counter function.	See description above

C Function Prototype Form:

ViStatus ats6943e_queryCounterFunction (ViSession instrumentHandle, ViInt16 *function);

Query Counter Input

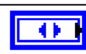
LabVIEW Diagram:



Description:

This function returns the counter/timer input signal source and slope.

Parameters:

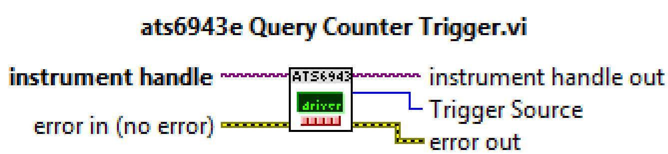
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Counter Input		This control specifies the counter input signal to query.	0 = CINPUT1 1 = CINPUT2 2 = CINPUT3
Signal	I16	This control is used to return the source of the selected counter input signal.	0 = I/O Channel 1 = AUX Channel 2 = Frequency Synthesizer 3 = CLK10 4 = 250MHz 5 = Pulse Generator
Channel Number	I16	This control is used to return which I/O or AUX channel number when the source is set to I/O Channel (0) or AUX Channel (1).	I/O Channel: 1 to 32 AUX Channel: 1 to 12
Slope	I16	This control is used return the counter input signal slope.	0 = Positive 1 = Negative

C Function Prototype Form:

ViStatus ats6943e_queryCounterInput (ViSession instrumentHandle, ViInt16 counterInput, ViInt16 *signal, ViInt16 *channelNumber, ViInt16 *slope);

Query Counter Trigger

LabVIEW Diagram:



Description:

This vi returns the counter trigger source.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Trigger Source	I16	This control is used to return the counter trigger source setting.	0 = None 1 = External 2 = Internal Continuous 3 = Internal Single

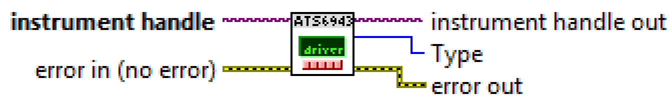
C Function Prototype Form:

ViStatus ats6943e_queryCounterTrigger (ViSession instrumentHandle, ViInt16 *triggerSource);

Query CRC Type

LabVIEW Diagram:

ats6943e Query CRC Type.vi



Description:

This vi returns the CRC type.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Type	I16	This control is used to return the CRC type.	0 = CRC16 1 = CRC32 2 = Custom

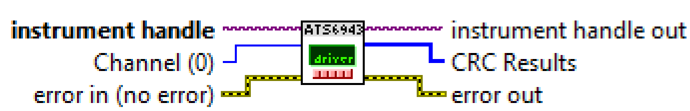
C Function Prototype Form:

ViStatus ats6943e_queryCrcType (ViSession instrumentHandle, ViInt32 *type);

Query CRC16 Results

LabVIEW Diagram:

ats6943e Query CRC16 Results.vi



Description:

This vi returns the CRC data from the 16 bit CRC memory for 1 or all 32 channels.

Parameters:

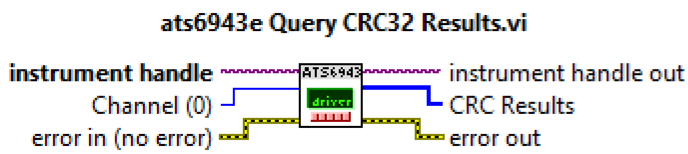
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Channel	I16	This control specifies the channel CRC to query.	0 to 32 0 returns all 32 channels.
CRC Results	[I16]	This control is used to return the CRC results.	0 to 65535

C Function Prototype Form:

ViStatus ats6943e_queryCrc (ViSession instrumentHandle, ViInt16 channel, ViInt16 CRCResults[]);

Query CRC32 Results

LabVIEW Diagram:



Description:

This vi returns the CRC32 data from the 32 bit CRC memory for 1 or all 32 channels.

Parameters:

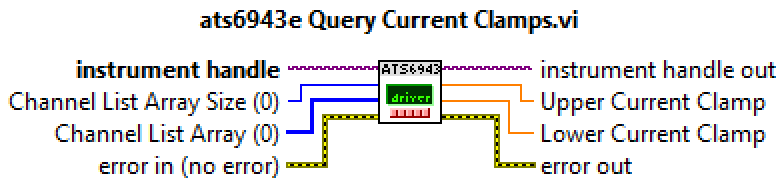
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Channel	I16	This control specifies the channel CRC to query.	0 to 32 0 returns all 32 channels.
CRC Results	[U32]	This control is used to return the CRC results.	0 to 4294967295

C Function Prototype Form:

ViStatus ats6943e_queryCrc32 (ViSession instrumentHandle, ViInt16 channel, ViInt32 CRCResults[]);

Query Current Clamps

LabVIEW Diagram:



Description:

This function returns the upper and lower current clamps for the specified channels.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Upper Current Clamp	DBL	This control is used to return the upper current clamp value (mA).	-2 * IR to 2 * IR IR = Current Range
Lower Current Clamp	DBL	This control is used to return the lower current clamp value (mA).	-2 * IR to 2 * IR IR = Current Range

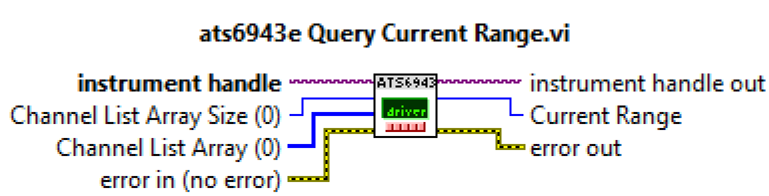
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same current clamp setting.

C Function Prototype Form:

ViStatus ats6943e_queryPmuIClamps (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 *upperCurrentClamp, ViReal64 *lowerCurrentClamp);

Query Current Range

LabVIEW Diagram:



Description:

This vi returns the current range (IR) for PMU operation for the specified channel(s). This setting determines the valid range and resolution for the current clamps and force current levels.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Current Range	I16	This control returns the IR.	0 = 5µA 1 = 50µA 2 = 500µA 3 = 5mA 4 = 50mA

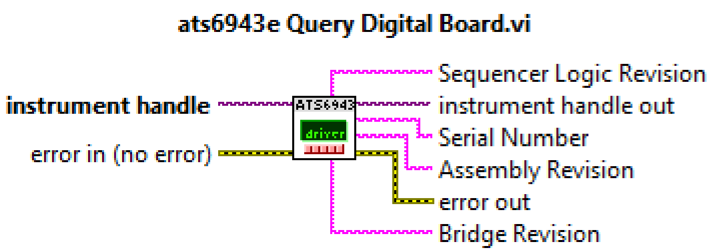
This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the same current range setting.

C Function Prototype Form:

ViStatus ats6943e_queryPmuIR (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *currentRange);

Query Digital Board

LabVIEW Diagram:



Description:

This vi returns the following Digital Board configuration information:

1. Serial Number
2. Assembly Revision
3. Bridge Logic Revision
4. Sequencer Logic Revision.

Parameters:

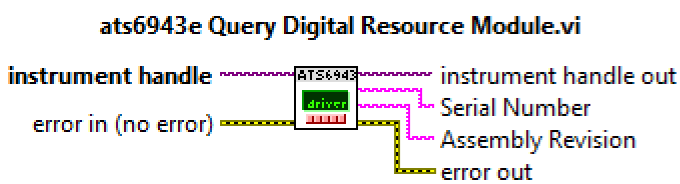
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Sequencer Logic Revision	abc	This control returns the sequencer logic revision.	ASCII string returned will be <= 10 characters.
Serial Number	abc	This control returns the serial number.	ASCII string returned will be <= 16 characters.
Assembly Revision	abc	This control returns the assemblyrevision.	ASCII string returned will be <= 4 characters.
Bridge Revision	abc	This control returns the bridge logic revision.	ASCII string returned will be <= 10 characters.

Function Prototype Form:

ViStatus ats6943e_dcbQuery (ViSession instrumentHandle, ViString *serialNumber, ViString *assemblyRevision, ViString *bridgeRevision, ViString *sequencerLogicRevision);

Query Digital Resource Module

LabVIEW Diagram:



Description:

This vi returns the following Digital Resource Module configuration information:

1. Serial Number

Assembly Revision.

Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Serial Number	abc	This control returns the serial number.	ASCII string returned will be <= 16 characters.
Assembly Revision	abc	This control returns the assemblyrevision.	ASCII string returned will be <= 4 characters.

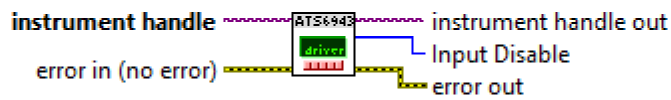
Function Prototype Form:

ViStatus ats6943e_drmQuery (ViSession instrumentHandle, ViString *serialNumber, ViString *assemblyRevision);

Query Driver Enable Control

LabVIEW Diagram:

ats6943e Query Driver Enable Control.vi



Description:

This vi returns driver enable control of the sequencer.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Input Disable	I16	This control is used to return the input disable setting.	0 = System Clock 1 = Phase Assert

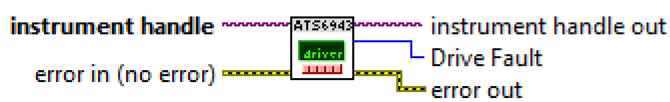
C Function Prototype Form:

ViStatus ats6943e_queryDriverEnableControl (ViSession instrumentHandle, ViInt16 *inputDisable);

Query Driver Fault State

LabVIEW Diagram:

ats6943e Query Driver Fault State.vi



Description:

This vi returns the drive fault state setting.

Key Parameters:

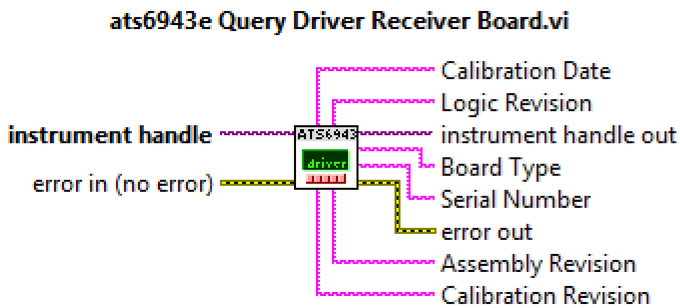
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Drive Fault	I16	This control is used to return the input disable setting.	0 = Disabled 1 = Enabled

C Function Prototype Form:

ViStatus ats6943e_queryDriveFaultState (ViSession instrumentHandle, ViInt16 *driveFault);

Query Driver Receiver Board

LabVIEW Diagram:



Description:

This vi returns the following Driver/Receiver Board configuration information:

1. Board Type
2. Serial Number
3. Assembly Revision
4. Logic Revision
5. Calibration Revision
6. Calibration Date

Key Parameters:

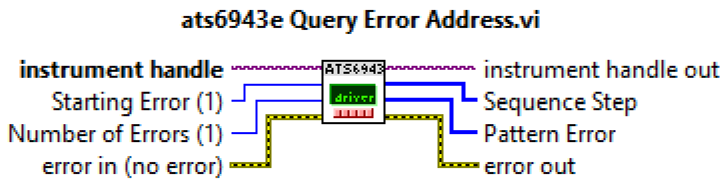
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Calibration Date	abc	This control returns the calibration date.	ASCII string returned will be <= 11 characters.
Logic Revision	abc	This control returns the logic revision.	ASCII string returned will be <= 10 characters.
Board Type	abc	This control returns the board type.	DR1: “6943DR1X” DR3: “6943DR3”
Serial Number	abc	This control returns the serial number.	ASCII string returned will be <= 16 characters.
Assembly Revision	abc	This control returns the assembly revision.	ASCII string returned will be <= 4 characters.
Calibration Revision	abc	This control returns the calibration revision.	ASCII string returned will be <= 10 characters.

C Function Prototype Form:

ViStatus ats6943e_drQuery (ViSession instrumentHandle, ViString *boardType, ViString *serialNumber, ViString *assemblyRevision, ViString *logicRevision, ViString *calibrationRevision, ViString *calibrationDate);

Query Error Address

LabVIEW Diagram:



Description:

This vi returns the error address memory contents.

The error address memory records the sequence step, and pattern address of the first 1024 errors of a sequence execution.

Key Parameters:

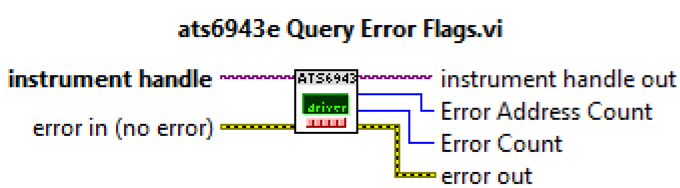
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Starting Error	I16	This control specifies the beginning error number of the error address memory to query.	1 to 1024
Number of Errors	I16	This control specifies the number of errors to query.	1 to 1024
Sequence Step	[I16]	This control is used to return the step number of the pattern error.	0 to 4095
Pattern Error	[U32]	This control is used to return the address of the pattern error.	0 to 262143

C Function Prototype Form:

ViStatus ats6943e_queryErrorAddress (ViSession instrumentHandle, ViInt16 startingError, ViInt16 numberOfErrors, ViInt16 sequenceStep[], ViInt32 patternError[]);

Query Error Flags

LabVIEW Diagram:



Description:

This vi returns the error results of the previous sequence run.

The error results consists of the following:

1. Error Address Count - The number of error address memory locations recorded in the previous sequence run.
2. Error Count - Returns the number of patterns that did not match in the previous sequence run.

Key Parameters:

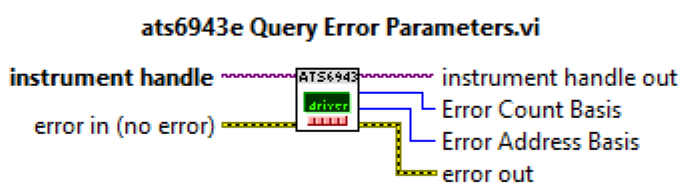
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Error Address Count	I16	This control is used to return the number of error address memory locations that have valid data from the selected sequencer.	0 to 1023
Pattern Error	I32	This control is used to return the pattern error count from the selected sequencer.	0 (no error) to 65535

C Function Prototype Form:

ViStatus ats6943e_queryErrorFlags (ViSession instrumentHandle, ViInt16 *errorAddressCount, ViInt32 *errorCount);

Query Error Parameters

LabVIEW Diagram:



Description:

This vi returns the error parameters of the sequencer.

Key Parameters:

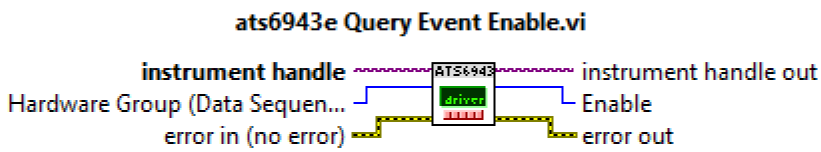
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Error Count Basis	I16	This control is used to return the error Counter basis.	0 = Local Error 1 = Qualified Local Error 2 = DTS Error 3 = Qualified DTS Error
Error Address Basis	I16	This control is used to return the Error Address basis.	0 = Local Error 1 = Qualified Local Error 2 = DTS Error 3 = Qualified DTS Error

C Function Prototype Form:

ViStatus ats6943e_queryErrorParameters (ViSession instrumentHandle, ViInt16 *errorCountBasis, ViInt16 *errorAddressBasis);

Query Event Enable

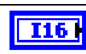
LabVIEW Diagram:



Description:

This vi queries the event enable register of the specified hardware.

Key Parameters:

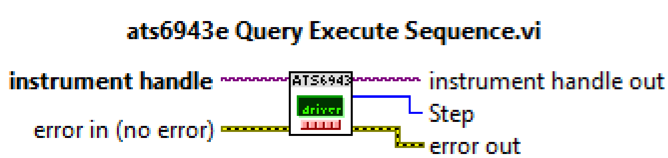
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Hardware Group		This control specifies which hardware group setting(s) to query.	0 = Data Sequencer 1 = Driver Receiver 2 = Digital Board
Enable		This control returns the enable register of the selected hardware group.	See bit definitions above.

C Function Prototype Form:

ViStatus ats6943e_queryEventEnable (ViSession instrumentHandle, ViInt16 hardwareGroup, ViInt32 *enable);

Query Execute Sequence

LabVIEW Diagram:



Description:

This vi returns the last executed sequence step.

Key Parameters:

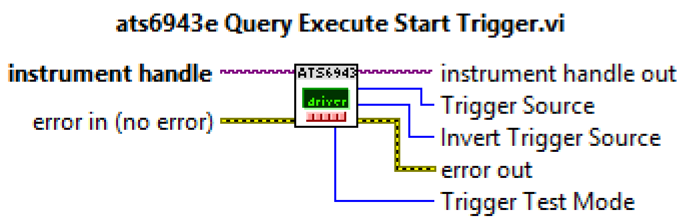
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Step	I16	This control is used to return the sequence step number last executed.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_queryExecuteSequence (ViSession instrumentHandle, ViInt16 *step);

Query Execute Start Trigger

LabVIEW Diagram:



Description:

This function returns the execute start trigger settings.

Refer to **Set Execute Start Trigger** for returned values.

Key Parameters:

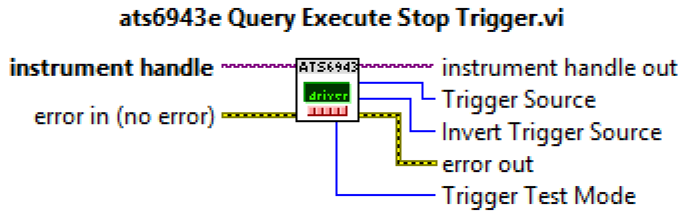
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Source	I16	This control is used to return the start trigger source.	See description above
Invert Trigger Source	I16	This control is used to return the start trigger source inverter.	See description above
Trigger Test Mode	I16	This control is used to return the start trigger test mode.	See description above

C Function Prototype Form:

ViStatus ats6943e_queryExecuteStartTrigger (ViSession instrumentHandle, ViInt16 *triggerSource, ViInt16 *invertTriggerSource, ViInt16 *triggerTestMode);

Query Execute Stop Trigger

LabVIEW Diagram:



Description:

This vi returns the execute stop trigger settings.

Refer to **Set Execute Stop Trigger** for returned values.

Key Parameters:

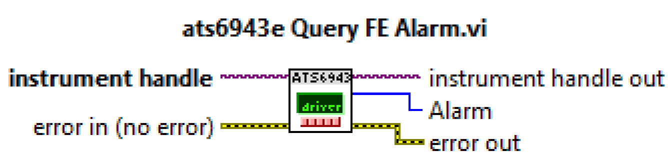
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Source	I16	This control is used to return the stop trigger source.	See description above
Invert Trigger Source	I16	This control is used to return the stop trigger source inverter.	See description above
Trigger Test Mode	I16	This control is used to return the stop trigger test mode.	See description above

C Function Prototype Form:

ViStatus ats6943e_queryExecuteStopTrigger (ViSession instrumentHandle, ViInt16 *triggerSource, ViInt16 *invertTriggerSource, ViInt16 *triggerTestMode);

Query FE Alarm

LabVIEW Diagram:



Description:

This vi returns the front-end temperature alarm bits for all 32 channels.

Key Parameters:

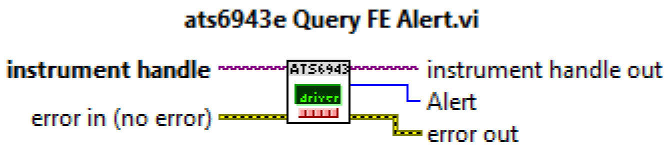
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Alarm	U32	This control returns the temperature alarm register.	Bit n maps to CHn+1 Bit high = Alarm True

C Function Prototype Form:

ViStatus ats6943e_queryFrontEndAlarm (ViSession instrumentHandle, ViInt32 *alarm);

Query FE Alert

LabVIEW Diagram:



Description:

This vi returns the front-end alert register.

Any bit set high in this register indicates an alert condition.

Over voltage alerts open up the channel connect relays.

Regulator power good and over temperature alerts will open the channel relays as well as shutting down the VCC, VEE and HV_VCC regulators.

Bit	Label	Description
0	VDD1PG	1 = VDD for CH1 to CH16 regulator fault
1	VCC1PG	1 = VCC for CH1 to CH16 regulator fault
2	VEE1PG	1 = VEE for CH1 to CH16 regulator fault
3	HVVCC1PG	1 = HVVCC for CH1 to CH16 regulator fault
4	VDD2PG	1 = VDD for CH17 to CH32 regulator fault
5	VCC2PG	1 = VCC for CH17 to CH32 regulator fault
6	VEE2PG	1 = VEE for CH17 to CH32 regulator fault
7	HVVCC2PG	1 = HVVCC for CH17 to CH32 regulator fault
10	OVH1	1 = Over voltage high CH1 to CH16
11	OVL1	1 = Over voltage low CH1 to CH16
12	OVH2	1 = Over voltage high CH17 to CH32
13	OVL2	1 = Over voltage low CH17 to CH32
16	ALARM1	1 = CH1-CH4 temperature alarm
17	ALARM2	1 = CH5-CH8 temperature alarm
18	ALARM3	1 = CH9-CH12 temperature alarm
19	ALARM4	1 = CH13-CH16 temperature alarm
20	ALARM5	1 = CH17-CH20 temperature alarm
21	ALARM6	1 = CH21-CH24 temperature alarm
22	ALARM7	1 = CH25-CH28 temperature alarm
23	ALARM8	1 = CH29-CH32 temperature alarm

Bits not listed above are unused.

Key Parameters:

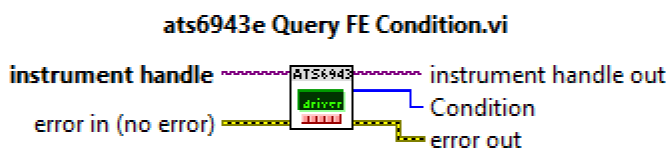
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Alert	U32	This control returns the alert register.	Bit n maps to CHn+1 Bit high = Alarm True

C Function Prototype Form:

ViStatus ats6943e_queryFrontEndAlert (ViSession instrumentHandle, ViInt32 *alert);

Query FE Condition

LabVIEW Diagram:



Description:

This vi returns the front-end condition register.

Unlike the event register, the condition register contains the current status of the hardware. The condition register is not cleared after reading.

Bit	Label	Description
0	VDD1PG	0 = VDD for CH1 to CH16 regulator off
1	VCC1PG	0 = VCC for CH1 to CH16 regulator off
2	VEE1PG	0 = VEE for CH1 to CH16 regulator off
3	HVVCC1PG	0 = HVVCC for CH1 to CH16 regulator off
4	VDD2PG	0 = VDD for CH17 to CH32 regulator off
5	VCC2PG	0 = VCC for CH17 to CH32 regulator off
6	VEE2PG	0 = VEE for CH17 to CH32 regulator off
7	HVVCC2PG	0 = HVVCC for CH17 to CH32 regulator off
10	OVH1	0 = One or more CH1 to CH16 > HVVCC1
11	OVL1	0 = One or more CH1 to CH16 < VEE1
12	OVH2	0 = One or more CH17 to CH32 > HVVCC2
13	OVL2	0 = One or more CH17 to CH32 < VEE2
14	MPSIG	MPSIG level
16	ALARM1	0 = CH1-CH4 temperature alarm true
17	ALARM2	0 = CH5-CH8 temperature alarm true
18	ALARM3	0 = CH9-CH12 temperature alarm true

Bit	Label	Description
19	ALARM4	0 = CH13-CH16 temperature alarm true
20	ALARM5	0 = CH17-CH20 temperature alarm true
21	ALARM6	0 = CH21-CH24 temperature alarm true
22	ALARM7	0 = CH25-CH28 temperature alarm true
23	ALARM8	0 = CH29-CH32 temperature alarm true

Bits not listed above are unused.

Key Parameters:

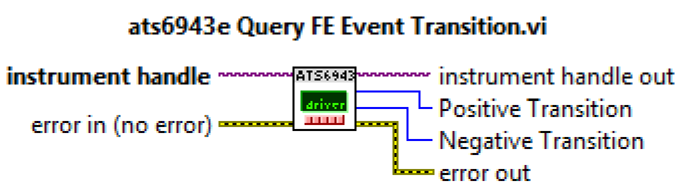
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Condition	I32	This control returns the condition register.	See table above

C Function Prototype Form:

ViStatus ats6943e_queryFrontEndCondition (ViSession instrumentHandle, ViInt16 *condition);

Query FE Event Transition

LabVIEW Diagram:



Description:

This vi returns the front-end positive and negative transition enable registers.

Bit n high of the positive transition enables the low to high transition of bit n in the condition register to set bit n in the event register high.

Bit n high of the negative transition enables the high to low transition of bit n in the condition register to set bit n in the event register high.

See **Query FE Condition** for register bit definitions.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Positive Transition	I32	This control returns the positive transition enable register.	See description above

Negative Transition	I32	This control returns the negative transition enable register.	See description above

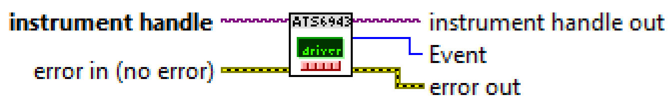
C Function Prototype Form:

ViStatus ats6943e_queryFrontEndEventTransition (ViSession instrumentHandle, ViInt32 *positiveTransition, ViInt32 *negativeTransition);

Query FE Event

LabVIEW Diagram:

ats6943e Query FE Event.vi



Description:

This vi returns the front-end event register.

A bit set in the event register indicates an enabled transition occurred in the corresponding condition register bit.

All bits are cleared automatically after reading.

See **Query FE Condition** for register bit definitions.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Event	I32	This control returns the event register.	See description above

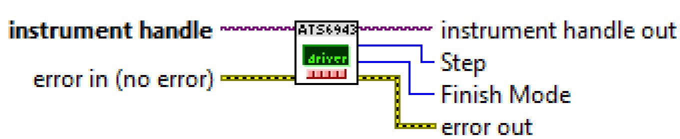
C Function Prototype Form:

ViStatus ats6943e_queryFrontEndEvent (ViSession instrumentHandle, ViInt16 *event);

Query Finish Sequence

LabVIEW Diagram:

ats6943e Query Finish Sequence.vi



Description:

This vi returns the finishing sequence step number and mode.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Step	I16	This control is used to return the finish sequence step number.	0 to 4095
Finish Mode	I16	This control is used to return the finish mode for sequence execution.	0 = Go to Standby 1 = Go to Idle

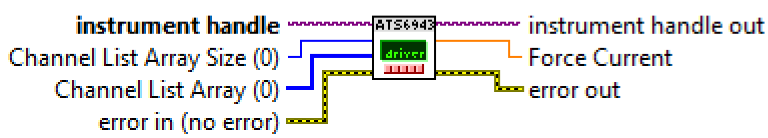
C Function Prototype Form:

ViStatus ats6943e_queryFinishSequence (ViSession instrumentHandle, ViInt16 *step, ViInt16 *finishMode);

Query Force Current

LabVIEW Diagram:

ats6943e Query Force Current.vi



Description:

This vi returns the force current level for the specified channels.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Force Current	DBL	This control returns the force current level in mA.	-50.0 to +50.0

This vi will return the warning ATS6943E_WARN_CLVALUESNEQ if all the channels do not have the force current level.

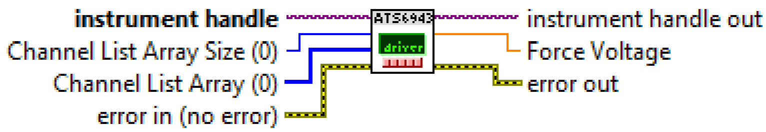
C Function Prototype Form:

ViStatus ats6943e_queryPmuFI (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 *forceCurrent);

Query Force Voltage

LabVIEW Diagram:

ats6943e Query Force Voltage.vi



Description:

This vi return the force voltage level for the specified channels.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Force Voltage	DBL	This control returns the force voltage level.	-2.0 to +7.0

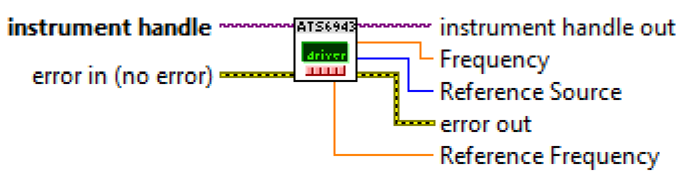
C Function Prototype Form:

ViStatus ats6943e_queryPmuFV (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 *forceVoltage);

Query Frequency Synthesizer

LabVIEW Diagram:

ats6943e Query Frequency Synthesizer.vi



Description:

This vi queries the frequency synthesizer settings.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Frequency	DBL	This control returns the synthesizer frequency (MHz).	0 to 500 0 = Disabled

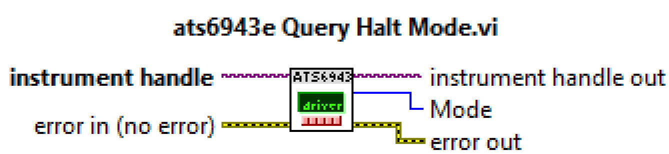
Name	Type	Description	Value
Reference Source	I16	This control is used to return the synthesizer reference clock source.	0 = Internal 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CLK10 14 = 50MHz (CLK100 / 2)
Reference Frequency	DBL	This control returns the synthesizer reference clock frequency (MHz).	5 to 100

C Function Prototype Form:

ViStatus ats6943e_queryFreqSynth (ViSession instrumentHandle, ViReal64 *frequency, ViInt16 *referenceSource, ViReal64 *referenceFrequency);

Query Halt Mode

LabVIEW Diagram:



Description:

This vi returns the halt mode.

Key Parameters:

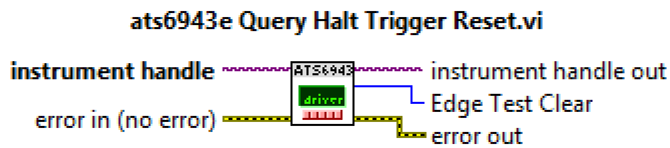
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Halt Mode	I16	This control is used to return the halt mode.	0 = Disable 1 = Pattern 2 = Step 3 = Sequence 4 = SYNC1 5 = SYNC2 6 = Pattern Fail 7 = Step Fail 8 = Sequence Fail 9 = Pattern Pass 10 = Step Pass 11 = Sequence Pass

C Function Prototype Form:

ViStatus ats6943e_queryHaltMode (ViSession instrumentHandle, ViInt16 *mode);

Query Halt Trigger Reset

LabVIEW Diagram:



Description:

This vi returns the halt trigger edge test clear condition.

Key Parameters:

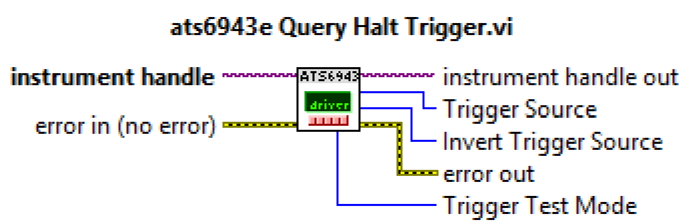
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Edge Test Clear	I16	This control is used to return the edge test clear condition.	0 = Start 1 = End of Step 2 = Trigger True

C Function Prototype Form:

ViStatus ats6943e_queryHaltTriggerReset (ViSession instrumentHandle, ViInt16 *edgeTestClear);

Query Halt Trigger

LabVIEW Diagram:



Description:

This vi returns the halt trigger settings of the selected data sequencer.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

Name	Type	Description	Value
Trigger Source	I16	This control is used to return the halt trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source	I16	This control is used to return the halt trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode	I16	This control is used to return the halt trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

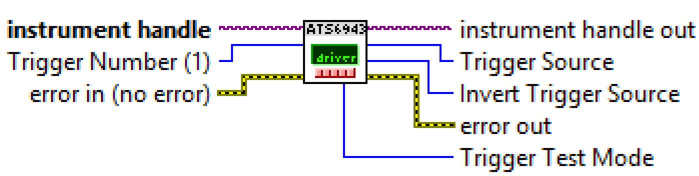
C Function Prototype Form:

ViStatus ats6943e_queryHaltTrigger (ViSession instrumentHandle, ViInt16 *triggerSource, ViInt16 *invertTriggerSource, ViInt16 *triggerTestMode);

Query Handshake Pause Trigger

LabVIEW Diagram:

ats6943e Query Handshake Pause Trigger.vi



Description:

This vi returns the specified handshake pause trigger settings.

Key Parameters:

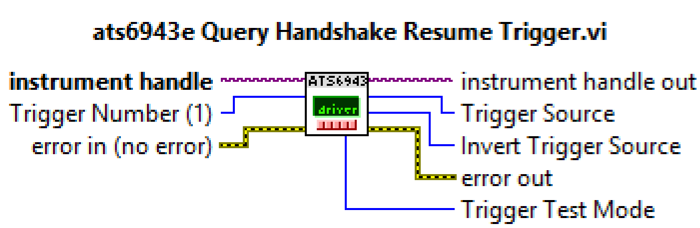
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Number	I16	This control specifies the handshake pause trigger to query.	0 to 1
Trigger Source	I16	This control is used to return the pause trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source	I16	This control is used to return the pause trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode	I16	This control is used to return the pause trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

C Function Prototype Form:

ViStatus ats6943e_queryHandshakePauseTrigger (ViSession instrumentHandle, ViInt16 triggerNumber, ViInt16 *triggerSource, ViInt16 *invertTriggerSource, ViInt16 *triggerTestMode);

Query Handshake Resume Trigger

LabVIEW Diagram:



Description:

This vi returns the specified pause resume trigger settings.

Key Parameters:

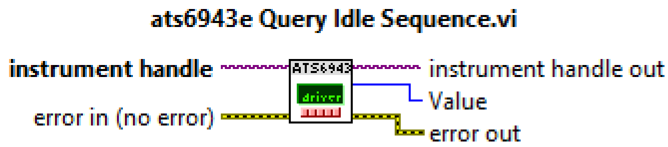
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to $2^{32}-1$
Trigger Number	I16	This control specifies the pause resume trigger to query.	0 to 1
Trigger Source	I16	This control is used to return the pause trigger resume source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source	I16	This control is used to return the pause trigger resume source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode	I16	This control is used to return the pause trigger resume test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

C Function Prototype Form:

ViStatus ats6943e_queryHandshakeResumeTrigger (ViSession instrumentHandle, ViInt16 triggerNumber, ViInt16 *triggerSource, ViInt16 *invertTriggerSource, ViInt16 *triggerTestMode);

Query Idle Sequence

LabVIEW Diagram:



Description:

This vi returns the idle sequence step number.

Key Parameters:

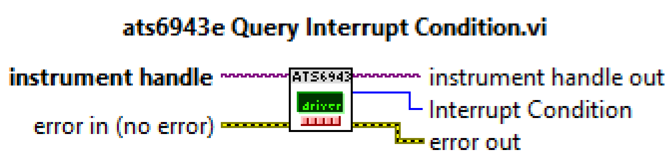
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Value	I16	This control is used to return the idle sequence step number.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_queryIdleSequence (ViSession instrumentHandle, ViInt16 *value);

Query Interrupt Condition

LabVIEW Diagram:



Description:

This vi returns the interrupt condition register.

Unlike the event register, the condition register contains the current status of the hardware. The condition register is not cleared after reading.

Bit	Label	Description
0	Sequencer Summary Bit	1 = Enabled sequencer event bit set.
1	NU	Not used.
2	Driver/Receiver Summary Bit	1 = Enabled driver/receiver event bit set.
3	NU	Not used.
4	Digital Board Summary Bit	1 = Enabled digital board event bit set.

Bits not listed above are unused.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Interrupt Condition	I16	This control returns the interrupt condition register.	See description above.

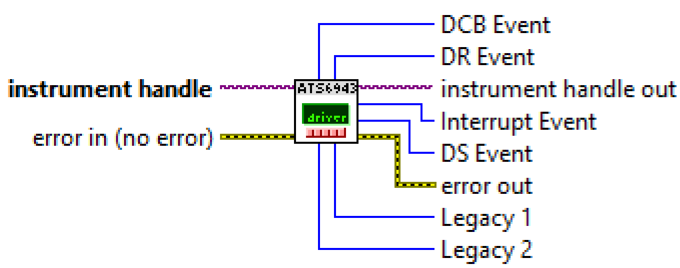
C Function Prototype Form:

ViStatus ats6943e_queryInterruptCondition (ViSession instrumentHandle, ViInt16 *interruptCondition);

Query Interrupt Event

LabVIEW Diagram:

ats6943e Query Interrupt Event.vi



Description:

This vi returns the event register(s) for the driver receiver, digital board and data sequencer.

See **Query Interrupt Condition** for the bit description of the DCB event.

See **Query FE Condition** for the bit description of the DR event.

See **Query Sequencer Condition** for the bit description of the DS event.

Key Parameters:

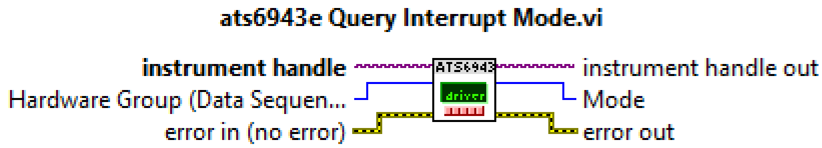
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
DCB Event	I16	This control returns the DCB event register.	See description above.
DCB Event	U32	This control returns the DCB event register.	See description above.
DCB Event	U32	This control returns the DCB event register.	See description above.

C Function Prototype Form:

ViStatus ats6943e_queryInterruptEvent (ViSession instrumentHandle, ViInt32 *interruptEvent, ViInt32 *DSEvent, ViInt32 *legacy1, ViInt32 *DREvent, ViInt32 *legacy2, ViInt32 *DCBEvent);

Query Interrupt Mode

LabVIEW Diagram:



Description:

This vi returns the interrupt mode for the specified hardware group.

Key Parameters:

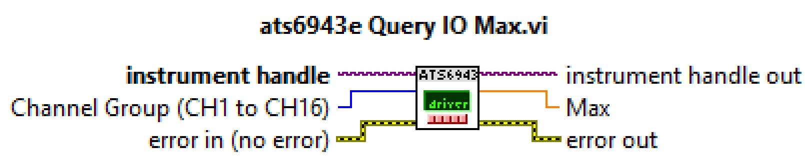
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Hardware Group		This control specifies which hardware group setting(s) to query.	0 = Data Sequencer 1 = Driver Receiver 2 = Digital Board
Mode	I16	This control is used to return which event transition to generate an interrupt on.	0 = None 1 = Event True 2 = Event False 3 = Event True or False

C Function Prototype Form:

ViStatus ats6943e_queryInterruptMode (ViSession instrumentHandle, ViInt16 hardwareGroup, ViInt16 *mode);

Query IO Max

LabVIEW Diagram:



Description:

This vi queries the front-end IO max level.

Key Parameters:

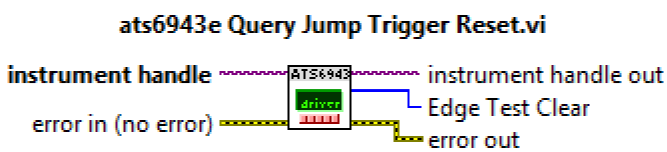
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel Group		This control specifies the IO max channel group.	0 = CH1 to CH16 1 = CH17 to CH32
Max	DBL	This control is used to return the IO max setting for the specified group.	-1.5 to 7.0

C Function Prototype Form:

ViStatus ats6943e_queryIoMax (ViSession instrumentHandle, ViInt16 channelGroup, double *max);

Query Jump Trigger Reset

LabVIEW Diagram:



Description:

This vi returns the jump trigger edge test clear setting.

Key Parameters:

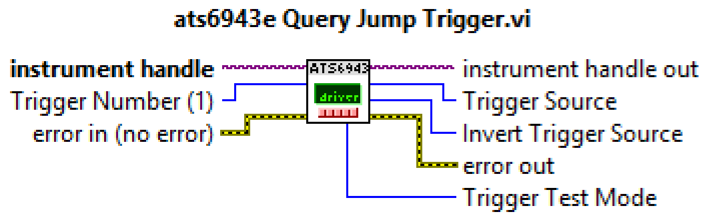
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Edge Test Clear	I16	This control is used to return the edge test clear condition.	0 = Start 1 = End of Step 2 = Trigger True

C Function Prototype Form:

ViStatus ats6943e_queryJumpTriggerReset (ViSession instrumentHandle, ViInt16 *edgeTestClear);

Query Jump Trigger

LabVIEW Diagram:



Description:

This vi returns the sequence jump trigger settings.

Key Parameters:

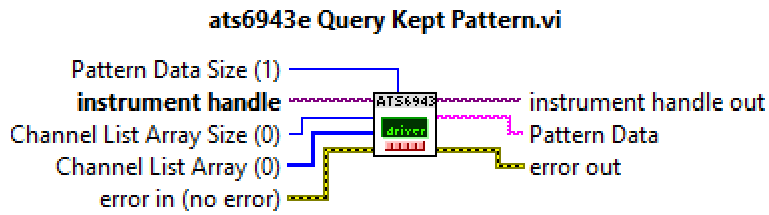
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Number	I16	This control specifies the sequence jump trigger to query.	0 to 1
Trigger Source	I16	This control is used to return the jump trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source	I16	This control is used to return the jump trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode	I16	This control is used to return the jump trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

C Function Prototype Form:

ViStatus ats6943e_queryJumpTrigger (ViSession instrumentHandle, ViInt16 triggerNumber, ViInt16 *triggerSource, ViInt16 *invertTriggerSource, ViInt16 *triggerTestMode);

Query Kept Pattern

LabVIEW Diagram:



Description:

This vi queries the kept pattern register. As patterns are output in a burst, a copy of the last pattern executed is stored in the "Kept" pattern register.

Pin State	Character Code	Description
Disabled	Z	Driver is either HiZ or VTT level
Collect CRC	C	Response level captured in CRC register
Drive High	1	Driver enabled and set to DVH level
Drive Low	0	Driver enable and set to DVL level
Expect Valid Low	L	Driver disabled, generate error if input > CVL
Expect Valid High	H	Driver disabled, generate error if input < CVH
Expect Valid	V	Driver disabled, generate error if input > CVL and input < DVH
Expect Between	B	Driver disabled, generate error if input < CVL or input > DVH
Drive Low, Expect Low	l	Driver enabled and set to DVL, generator error if input > CVL
Drive High, Expect High	h	Driver enabled and set to DVH, generator error if input < CVH
Drive Low, Expect High	/	Driver enabled and set to DVL, generator error if input < CVH
Drive High, Expect Low	\	Driver enabled and set to DVH, generator error if input > CVL

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pattern Data Size	I16	This control specifies the number of elements in the pattern data array.	1 to 32
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query	1 to 32

Name	Type	Description	Value
Pattern Data	abc	This control is used to return the pattern data.	See description above

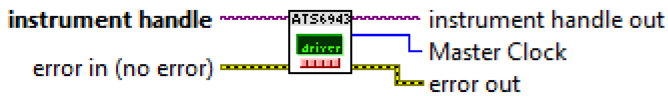
C Function Prototype Form:

ViStatus ats6943e_queryKeptPattern (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 patternDataSize, ViChar patternData[]);

Query Master Clock Source

LabVIEW Diagram:

ats6943e Query Master Clock Source.vi



Description:

This vi returns the master clock source of the sequencer.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Master Clock	I16	This control is used to return which master clock signal is selected.	0 = 500MHz 15 = Frequency Synthesizer

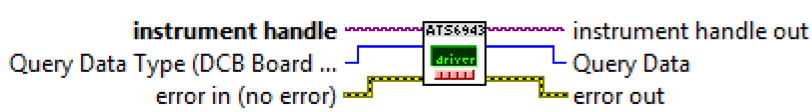
C Function Prototype Form:

ViStatus ats6943e_queryMasterClockSource (ViSession instrumentHandle, ViInt16 *masterClock);

Query Module Data

LabVIEW Diagram:

ats6943e Query Module Data.vi



Description:

This vi will return the following module data:

- DCB Board Type
- Chassis Type
- ETB Link Installed
- Power Converter

Key Parameters:

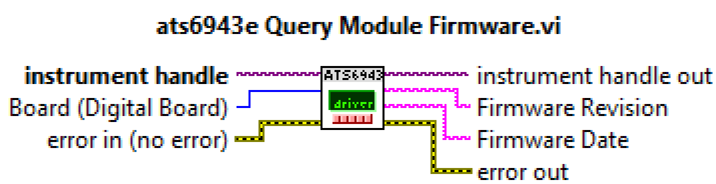
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Query Data Type	I16	This control specifies data to query.	0 = DB Type 1 = Chassis Type 2 = ETB Link Installed 3 = Power Converter
Query Data	I16	This control returns the query data.	DB Type 0 = PXIe 6943 Chassis Type 2 = PXIe ETB Link Installed 0 = Primary Link 1 = Secondary Link 2 = Terminator Link 3 = None Power Converter 0 = Not installed (legacy)

C Function Prototype Form:

ViStatus ats6943e_queryModuleData (ViSession instrumentHandle, ViInt16 queryDataType, ViInt16 *queryData);

Query Module Firmware

LabVIEW Diagram:



Description:

This vi will return the firmware revision and date for the selected board.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Board	I16	This control specifies the board to query.	0 = Digital 1 = Driver Receiver
Firmware Revision	abc	This control returns the firmware revision of the selected board.	"XX.YY". Will return "NA" if no firmware installed.
Firmware Date	abc	This control returns the firmware date of the selected board.	"DD.MM.YYYY". Will return "NA" if no firmware installed.

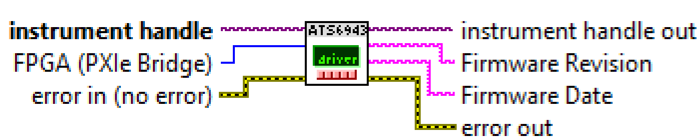
C Function Prototype Form:

ViStatus ats6943e_queryFirmware (ViSession instrumentHandle, ViInt16 board, ViString *firmwareRevision, ViString *firmwareDate);

Query Module FPGA

LabVIEW Diagram:

ats6943e Query Module FPGA.vi



Description:

This vi will return the revision and date for the selected FPGA.

Key Parameters:

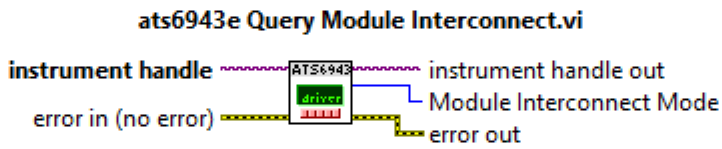
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
FPGA	I16	This control specifies the FPGA to query.	0 = PXIe Bridge 1 = Sequencer FPGA 2 = Inter Module FPGA 3 = Driver Receiver
FPGA Revision	abc	This control returns the revision of the selected FPGA.	"XX.YY". Will return "NA" if no firmware installed.
FPGA Date	abc	This control returns the date of the selected FPGA.	"DD.MM.YYYY". Will return "NA" if no firmware installed.

C Function Prototype Form:

ViStatus ats6943e_queryFpga (ViSession instrumentHandle, ViInt16 FPGA, ViString *firmwareRevision, ViString *firmwareDate);

Query Module Interconnect

LabVIEW Diagram:



Description:

This vi returns the module interconnect mode.

Key Parameters:

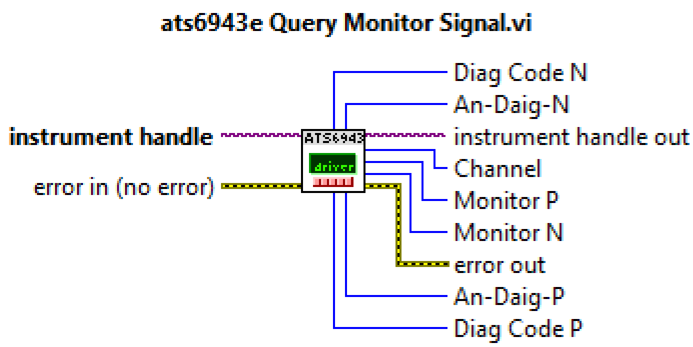
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Module Interconnect Mode	I16	This control is used to return the module interconnect mode setting.	0 = Independent 2 = Primary Synthesizer 7 = Secondary 10 = Terminator

C Function Prototype Form:

ViStatus ats6943e_queryModuleInterconnect (ViSession instrumentHandle, ViInt16 *moduleInterconnectMode);

Query Monitor Signal

LabVIEW Diagram:



Description:

This vi returns the signal source and channel of the Monitor signal.

Refer to **Set Monitor Signal** for a description of the value settings.

Key Parameters:

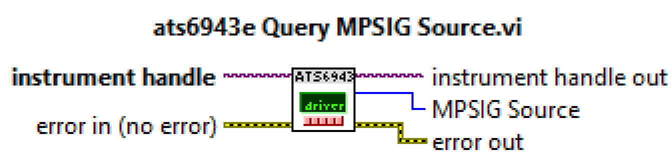
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel	I16	This control returns which channels monitor value is set.	0 to 32 (0 = monitor disabled)
Monitor P	I16	This control returns which signal is selected for Mon-P.	0 to 3
Monitor N	I16	This control returns which signal is selected for Mon-N.	0 to 3
An-Diag-P	I16	This control returns which signal is selected for An-Diag-P and is only valid when Mon-P selection is set to An-Diag-P.	0 to 7
An-Diag-N	I16	This control returns which signal is selected for An-Diag-N and is only valid when Mon-N selection is set to An-Diag-N.	0 to 7
Diag Code P	I16	This control returns the diagnostic code for the An-Diag-P selection.	0 to 15
Diag Code N	I16	This control returns the diagnostic code for the An-Diag-N selection.	0 to 15

C Function Prototype Form:

ViStatus ats6943e_queryMonitorSignal (ViSession instrumentHandle, ViInt16 *channel, ViInt16 *monitorP, ViInt16 *monitorN, ViInt16 *anDaigP, ViInt16 *anDaigN, ViInt16 *diagCodeP, ViInt16 *diagCodeN);

Query MPSIG Source

LabVIEW Diagram:



Description:

This vi programs the MPSIG source.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
MPSIG Source	I16	This control returns the MPSIG source bits. Bit set high adds the signal to the OR logic of MPSIG.	Bit 0 = Sequence Active Bit 1 = Paused Bit 2 = Halted Bit 3 = Burst Error Bit 4 = NU

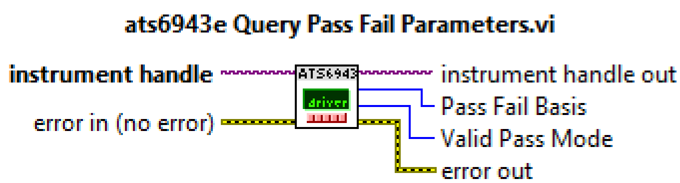
Name	Type	Description	Value
			Bit 5 = Drive Fault Bit 6 = Watchdog Timeout Bit 7 = Sequence Timeout Bit 8 = Pattern Error Bit 9 = Sync Error

C Function Prototype Form:

ViStatus ats6943e_queryMpsigSource (ViSession instrumentHandle, ViInt16 *MPSIGSource);

Query Pass Fail Parameters

LabVIEW Diagram:



Description:

This vi returns the pass/fail parameters of sequencer.

Key Parameters:

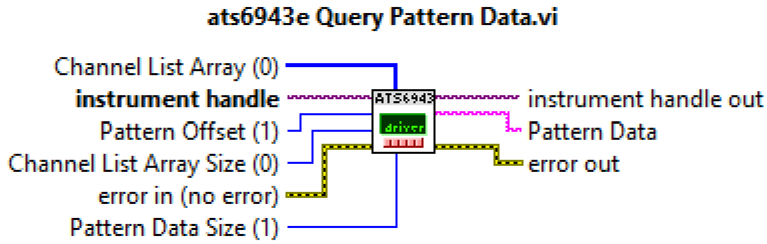
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pass Fail Basis	I16	This control is used to return the pass/fail basis.	0 = Local Errors 1 = Qualified Local Errors 2 = DTS Errors 3 = Qualified DTS Errors
Valid Pass Mode	I16	This control is used to return the valid pass enable mode.	0 = Disabled 1 = Enabled

C Function Prototype Form:

ViStatus ats6943e_queryPassFailParameters (ViSession instrumentHandle, ViInt16 *passFailBasis, ViInt16 *validPassMode);

Query Pattern Data

LabVIEW Diagram:



Description:

This vi queries a single pattern in a pattern set.

Each pattern in a pattern set contains data for 32 pins.

The pattern data is expressed as an ASCII code described below:

Pin State	Character Code	Description
Disabled	Z	Driver is either HiZ or VTT level.
Collect CRC	C	Response level captured in CRC register.
Drive High	1	Driver enabled and set to DVH level.
Drive Low	0	Driver enable and set to DVL level.
Expect Valid Low	L	Driver disabled, generate error if input > CV.L
Expect Valid High	H	Driver disabled, generate error if input < CVH.
Expect Valid	V	Driver disabled, generate error if input > CVL and input < DVH.
Expect Between	B	Driver disabled, generate error if input < CVL or input > DVH.
Drive Low, Expect Low	l	Driver enabled and set to DVL, generator error if input > CVL.
Drive High, Expect High	h	Driver enabled and set to DVH, generator error if input < CVH.
Drive Low, Expect High	/	Driver enabled and set to DVL, generator error if input < CVH.
Drive High, Expect Low	\	Driver enabled and set to DVH, generator error if input > CVL.

An optional pin list can be specified to define which pins to query and the order with respect to the pattern data. For example, the following pin list/pattern data arrays returns pin 7 driving low, pin 1 expect low and pin 14 expect high.

Array Index	Pin List Array	Pattern Data Array
0	7	‘0’
1	1	‘L’
2	14	‘H’

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pattern Data Size	I16	This control specifies the number of elements in the pattern data array.	1 to 32
Pattern Offset	I32	This control specifies the pattern offset to query.	1 to 262144
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query	1 to 32
Pattern Data	abc	This control is used to return the pattern data.	See description above.

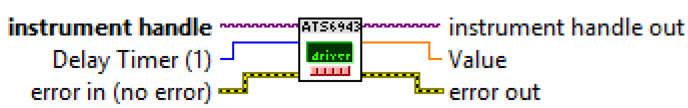
C Function Prototype Form:

ViStatus ats6943e_queryPatternData (ViSession instrumentHandle, ViInt32 patternOffset, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 patternDataSize, ViChar patternData[]);

Query Pattern Delay Timer

LabVIEW Diagram:

ats6943e Query Pattern Delay Timer.vi



Description:

This vi returns the pattern delay timer value.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Delay Timer	I16	This control specifies the pattern delay timer to query.	1 = Pattern Delay Timer 1 2 = Pattern Delay Timer 2
Value	DBL	This control is used to return the pattern delay timer value.	20ns to 42.94967297s

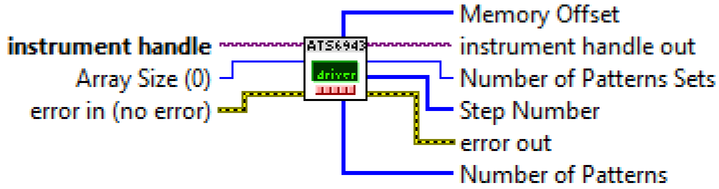
C Function Prototype Form:

ViStatus ats6943e_queryPatternDelayTimer (ViSession instrumentHandle, ViInt16 delayTimer, ViReal64 *value);

Query Pattern Set List

LabVIEW Diagram:

ats6943e Query Pattern Set List.vi



Description:

This vi queries all the pattern sets currently in the pattern memory.

The list is returned in three arrays and one variable. The variable indicates the number pattern sets. The three arrays contain the step number, number of patterns and memory offset for each pattern set.

For example, if after calling this vi the following data was returned,

Number of Pattern Sets = 5;

Step Number = [0, 1, 3, 4, 5]

Number of Patterns = [1, 33, 21, 14, 1]

Memory Offset = [0, 4, 40, 64, 80]

Step 0 pattern set with 1 pattern starting at offset 0.

Step 1 pattern set with 33 patterns starting at offset 4.

Step 3 pattern set with 21 patterns starting at offset 40.

Step 4 pattern set with 14 patterns starting at offset 64.

Step 5 pattern set with 1 pattern starting at offset 80.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Array Size	I16	The control specifies the number of elements in the "Step Number", "Number of Patterns" and "Memory Offset" control arrays.	1 = Pattern Delay Timer 1 2 = Pattern Delay Timer 2
Number of Pattern Sets	I16	This control is used to return the number of patterns sets in the pattern memory.	0 to 4096
Step Number	[I16]	This control is used to return the step number of each pattern set.	0 to 4095
Number of Patterns	[I32]	This control is used to return the number of patterns in each pattern set.	1 to 262144
Memory Offset	[I32]	This control is used to return the memory offset of each pattern set.	0 to 262143

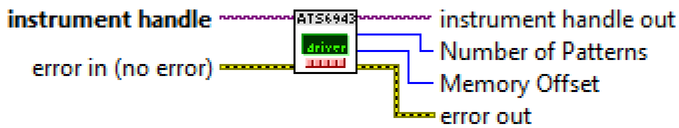
C Function Prototype Form:

ViStatus ats6943e_queryPatternSetList (ViSession instrumentHandle, ViInt16 *numberOfPatternsSets, ViInt16 arraySize, ViInt16 stepNumber[], ViInt32 numberOfPatterns[], ViInt32 memoryOffset[]);

Query Pattern Set

LabVIEW Diagram:

ats6943e Query Pattern Set.vi



Description:

This vi queries the pattern set of the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Number of Patterns	I32	This control is used to return the number of patterns in the pattern set.	1 to 262144
Memory Offset	I32	This control is used to return the memory offset of the pattern set.	0 to 262143

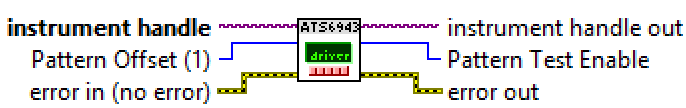
C Function Prototype Form:

ViStatus ats6943e_queryPatternSet (ViSession instrumentHandle, ViInt32 *numberOfPatterns, ViInt32 *memoryOffset);

Query Pattern Test Enable

LabVIEW Diagram:

ats6943e Query Pattern Test Enable.vi



Description:

This vi returns the pattern test enable setting for the specified offset.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

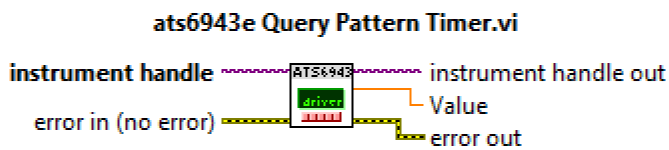
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pattern Offset	I32	This control specifies the pattern offset to query.	1 to 262144
Pattern Test Enable	I16	This control is used to return the sequence test enable setting for the sequence step pattern at the specified offset.	0 = None 1 = CONDEN 2 = BERREN 3 = Both

C Function Prototype Form:

ViStatus ats6943e_queryPatternTestEnable (ViSession instrumentHandle, ViInt16 *patternTestEnable, ViInt32 patternOffset);

Query Pattern Timer

LabVIEW Diagram:



Description:

This vi returns the pattern timeout value.

Key Parameters:

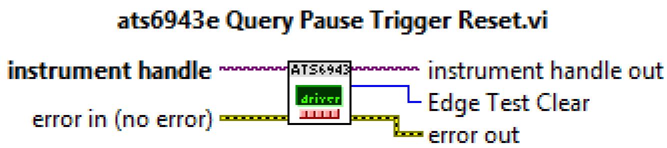
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Value	DBL	This control is used to return the pattern timeout value.	20ns to 42.94967297s

C Function Prototype Form:

ViStatus ats6943e_queryPatternTimer (ViSession instrumentHandle, ViReal64 *value);

Query Pause Trigger Reset

LabVIEW Diagram:



Description:

This vi returns the pause trigger edge test clear condition.

Key Parameters:

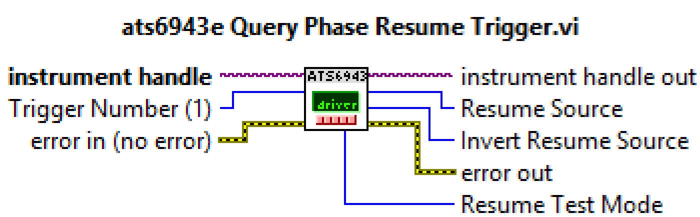
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Edge Test Clear	I16	This control is used to return the edge test clear condition.	0 = Start 1 = End of Step 2 = Trigger True

C Function Prototype Form:

ViStatus ats6943e_queryPauseTriggerReset (ViSession instrumentHandle, ViInt16 *edgeTestClear);

Query Phase Resume Trigger

LabVIEW Diagram:



Description:

This vi returns the phase resume trigger settings.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Number	I16	This control specifies the phase resume trigger to query.	1 to 4
Resume Source	I16	This control is used to return the phase resume source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3

Name	Type	Description	Value
			4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source	I16	This control is used to return the pause trigger resume source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode	I16	This control is used to return the pause trigger resume test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

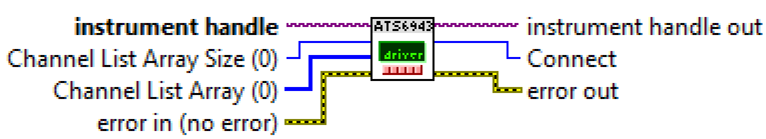
C Function Prototype Form:

ViStatus ats6943e_queryPhaseResumeTrigger (ViSession instrumentHandle, ViInt16 triggerNumber, ViInt16 *resumeSource, ViInt16 *invertResumeSource, ViInt16 *resumeTestMode);

Query PMU Connect

LabVIEW Diagram:

ats6943e Query PMU Connect.vi

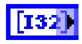


Description:

This vi queries the specified channels and returns a flag indicating the PMU connect state.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

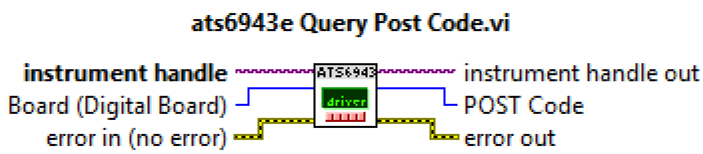
Channel List Array Size		The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array		Array contains the channel numbers to query.	1 to 32
Connect		This control is used to return the PMU connect state.	-2 = PMU mode not set. -1 = One or more connected. 0 = None connected. 1 = All connected.

C Function Prototype Form:

ViStatus ats6943e_queryPmuConnect (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *connect);

Query Post Code

LabVIEW Diagram:



Description:

This vi will return the power on self-test (POST) code of the selected board.

Digital Board POST codes:

Bit	Description
0	IIC bus 1 failed to initialize. IIC Bus 1 controls the frequency synthesizers, voltage/current monitors and the temperature monitor.
1	Frequency synthesizer I/O failure.
2	Temperature monitor I/O failure.
3	Voltage/Current monitor I/O failure initializing.
4	IIC bus 2 failed to initialize. IIC Bus 2 controls the V4 FPGA temperature monitor.
5	V4 temperature Monitor I/O failure.
6	SPI Flash I/O failed.
7	V4 FPGA not programmed.
8	Voltage/Current monitor I/O failure during update.
9	Front-end +12V voltage or current failure while reset: Current > 1.0A Voltage < 11.0V
10	Front-end +3.3V voltage or current failure while reset: Current > 0.3A Voltage < 3.0V
11	S6 FPGA not programmed.
12	Voltage/Current monitor I/O error during update.
13	Front-end +12V voltage or current failure after reset: Current > 2.0A

Bit	Description
	Voltage < 11.0V
14	Front-end +3.3V voltage or current failure after reset: Current > 4.1A Voltage < 3.0V
15	NVM flash I/O failure.
16	Frequency synthesizer I/O failure.
17	500MHz Clock failure.
18	V4 Input Delay Control failure.
19	Pattern memory bank 1 request timeout.
20	Pattern memory bank 2 request timeout.
21	Pattern memory bank 3 request timeout.
22	Pattern memory bank 4 request timeout.
23	Probe memory request timeout.
24	Driver Board I/O failure.

Driver Receiver POST codes:

Bit	Description
0	IIC bus failed to initialize. IIC Bus controls the voltage/current monitors and the VEE/HV-VCC DAC.
1	ADC SPI bus failed to initialize.
2	Flash SPI bus failed to initialize.
3	Voltage/Current monitor I/O failure initializing.
4	Voltage/Current monitor I/O error during update.
5	+3.3V voltage or current failure: Current > 1.0A Voltage < 3.2V
6	+5V voltage or current failure: Current > 0.5A Voltage < 4.75V
7	Vdd regulator failure.
8	CH1-CH4 chip failure.
9	CH5-CH8 chip failure.
10	CH9-CH12 chip failure.
11	CH13-CH16 chip failure.
12	CH17-CH20 chip failure.
13	CH21-CH24 chip failure.
14	CH25-CH28 chip failure.
15	CH29-CH32 chip failure.
16	Calibration flash I/O failure.
17	NVM flash I/O failure.
18	FPGA not loaded or firmware not running.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Board	I16	This control specifies the board POST to query.	0 = Digital Board 1 = Driver Receiver
POST Code	I32	This control returns the POST code of the selected board.	See description above

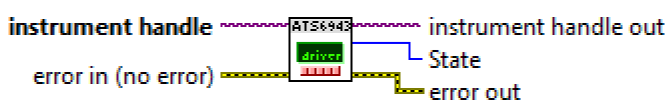
C Function Prototype Form:

ViStatus ats6943e_queryPostCode (ViSession instrumentHandle, ViInt16 board, ViInt32 *POSTCode);

Query Power Connect

LabVIEW Diagram:

ats6943e Query Power Connect.vi



Description:

This vi queries the front panel power connect state.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
State	I16	This control is used to return the front panel external power connect state.	0 = Open 1 = Close

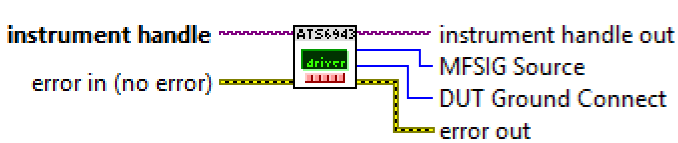
C Function Prototype Form:

ViStatus ats6943e_queryPowerConnect (ViSession instrumentHandle, ViInt16 *state);

Query Power Settings

LabVIEW Diagram:

ats6943e Query Power Settings.vi



Description:

This vi queries the front panel power connector settings.

Key Parameters:

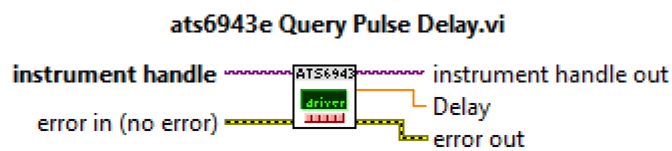
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
MFSIG Source	I16	This control is used to return the MFSIG source.	0 = Disabled 1 = MPSIG
DUT Ground Connect	I16	This control is used to return the DUT Ground Connect state.	0 = Signal Ground 1 = Front Panel

C Function Prototype Form:

ViStatus ats6943e_queryPowerSettings (ViSession instrumentHandle, ViInt16 *MFSIGSource, ViInt16 *DUTGroundConnect);

Query Pulse Delay

LabVIEW Diagram:



Description:

This vi returns the pulse generator delay.

Key Parameters:

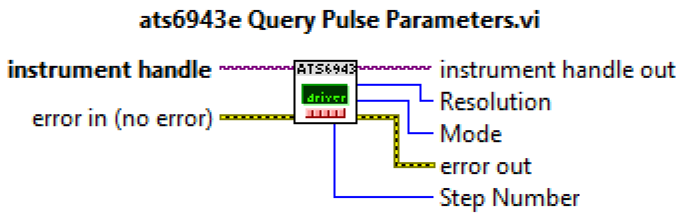
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pulse Delay	DBL	This control is used to return the pulse generator delay.	0.0 to 42.949672960

C Function Prototype Form:

ViStatus ats6943e_queryPulseDelay (ViSession instrumentHandle, ViReal64 *delay);

Query Pulse Parameters

LabVIEW Diagram:



Description:

This vi returns the pulse generator parameters.

Key Parameters:

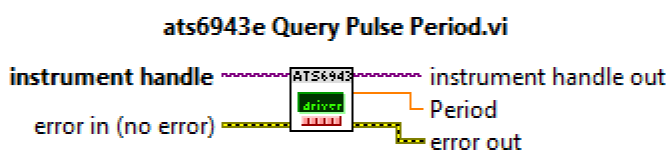
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Resolution	I16	This control is used to return the pulse period, width and delay resolution.	1 = 10ns
Mode	I16	This control is used to return the pulse generator mode.	0 = Continuous 1 = Continuous Start 2 = Single Start 3 = Single Step
Step Number	I16	This control is used to return the pulse generator mode.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_queryPulseParameters (ViSession instrumentHandle, ViInt16 *resolution, ViInt16 *mode, ViInt16 *stepNumber);

Query Pulse Period

LabVIEW Diagram:



Description:

This vi returns the pulse generator period.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

Name	Type	Description	Value
Pulse Period	DBL	This control is used to return the pulse generator period.	20.0e-9 to 42.949672970

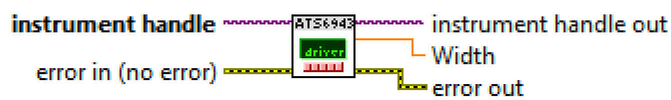
C Function Prototype Form:

ViStatus ats6943e_queryPulsePeriod (ViSession instrumentHandle, ViReal64 *period);

Query Pulse Width

LabVIEW Diagram:

ats6943e Query Pulse Width.vi



Description:

This vi returns the pulse generator width.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pulse Delay	DBL	This control is used to return the pulse generator width.	0.0 to 42.949672960

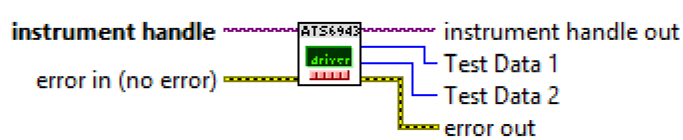
C Function Prototype Form:

ViStatus ats6943e_queryPulseWidth (ViSession instrumentHandle, ViReal64 *width);

Query Ram Test Results

LabVIEW Diagram:

ats6943e Query Ram Test Results.vi



Description:

This vi returns the pattern results from the last execution of the **Ram Test** vi.

RAM Test Results:

Returned RAM Test Error	Test 1 Data	Test 1 Data
ATS6943E_RAM_TEST_DBIT	Expected Data	Actual Data
ATS6943E_RAM_TEST_SA1	Failing memory address	Actual Data
ATS6943E_RAM_TEST_SA0	Failing memory address	Actual Data
ATS6943E_RAM_TEST_PAT5	Failing memory address	Actual Data
ATS6943E_RAM_TEST_PATA	Failing memory address	Actual Data

Key Parameters:

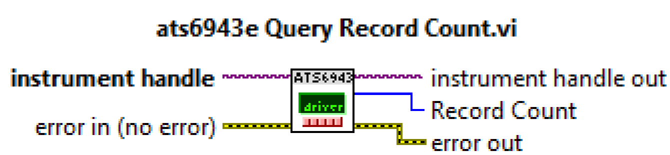
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Test Data 1	U32	This control is used to return additional results from the last RAM test run.	See description above.
Test Data 2	U32	This control is used to return additional results from the last RAM test run.	See description above.

C Function Prototype Form:

ViStatus ats6943e_queryRamTest (ViSession instrumentHandle, ViUInt32 *testData1, ViUInt32 *testData2);

Query Record Count

LabVIEW Diagram:



Description:

This vi returns the number of record memory results for the current execution sequence.

For indexed recording it’s the depth of the record memory recorded into.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
POST Code	I32	This control is used to return the record count.	0 to 262143

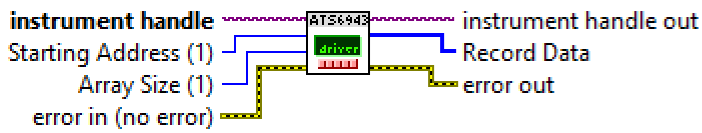
C Function Prototype Form:

ViStatus ats6943e_queryRecordCount (ViSession instrumentHandle, ViInt32 *recordCount);

Query Record Data

LabVIEW Diagram:

ats6943e Query Record Data.vi



Description:

This vi returns the record memory contents.

See **Record Memory** section in chapter 6 for a description of the record memory contents and settings.

The "Starting Address" plus the "Array Size" must be less than or equal to 262144.

Output

Key Parameters:

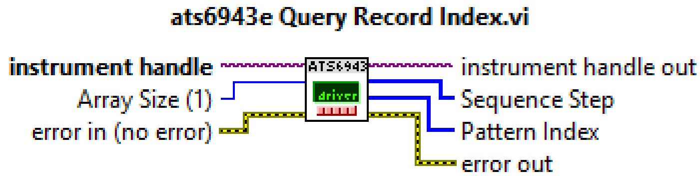
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Starting Address	I32	This control specifies the beginning address of the record memory to query.	0 to 262143
Array Size	I32	This control is used to return the record count.	1 to 262144
Record Data	[I32]	This control is used to return the contents of the record memory.	See description above.

C Function Prototype Form:

ViStatus ats6943e_queryRecordData (ViSession instrumentHandle, ViInt32 startingAddress, ViInt32 arraySize, ViInt32 recordData[]);

Query Record Index

LabVIEW Diagram:



Description:

This vi returns the record index memory contents.

The record index memory stores the sequence step and pattern index of the first 1024 steps of a sequence execution.

When the record type is set to indexed, the sequence results are stored sequentially in the record memory starting at offset 0. The record index memory contains the step number and record memory index for every step and is used re-align the recorded data with the pattern data.

Key Parameters:

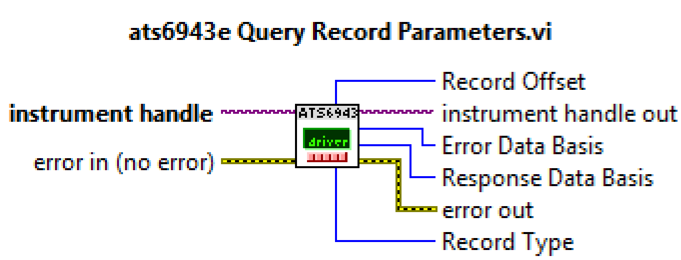
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Array Size	I16	This control specifies the number of elements to query.	1 to 1024
Sequence Step	[I16]	This control is used to return the sequence step data.	See description above.
Pattern Index	[I32]	This control is used to return the pattern index data.	See description above.

C Function Prototype Form:

ViStatus ats6943e_queryRecordIndex (ViSession instrumentHandle, ViInt16 arraySize, ViInt16 sequenceStep[], ViInt32 patternIndex[]);

Query Record Parameters

LabVIEW Diagram:



Description:

This vi returns the record parameters of the sequencer.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Record Offset	I16	This control returns the record offset.	0 to 63
Error Data Basis	I16	This control is used to return the error data basis.	0 = Dual Threshold 1 = Single Threshold
Response Data Basis	I16	This control is used to return the response data basis.	0 = Good 0 1 = Good 1
Record Type	I16	This control is used to return the response data basis.	0 = Normal 1 = Indexed

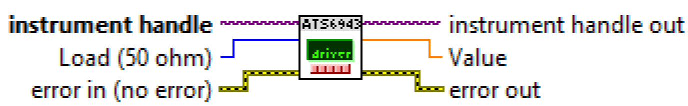
C Function Prototype Form:

ViStatus ats6943e_queryRecordParameters (ViSession instrumentHandle, ViInt16 *errorDataBasis, ViInt16 *responseDataBasis, ViInt16 *recordType, ViInt16 *recordOffset);

Query Reference Load

LabVIEW Diagram:

ats6943e Query Reference Load.vi



Description:

This vi returns the actual load resistor value measured from the EXTFORCE pin to GND. This reference is used for current load Source/Sink calibration.

Key Parameters:

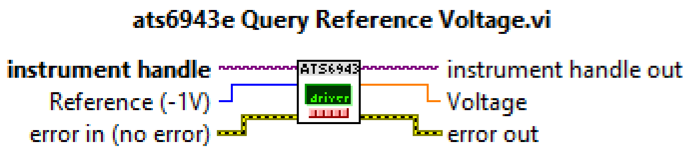
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Load	I16	This control specifies the reference load to query.	0 = 50 ohm 1 = 10K ohm
Value	DBL	This control returns the load resistance calibration value.	Load value entered.

C Function Prototype Form:

ViStatus ats6943e_queryRefLoad (ViSession instrumentHandle, ViInt16 load, ViReal64 *value);

Query Reference Voltage

LabVIEW Diagram:



Description:

This vi returns the reference voltage value set by user.

Key Parameters:

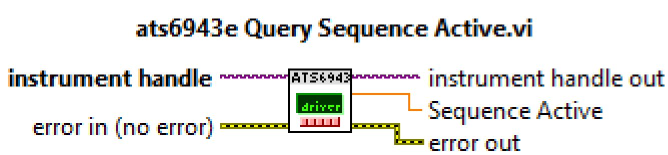
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Reference	I16	This control specifies the reference voltage to query.	0 = 5.0 1 = 3.33 2 = 1.66 3 = -1.0
Value	DBL	This control returns the reference voltage value programmed by the user.	Voltage value entered.

C Function Prototype Form:

ViStatus ats6943e_queryRefVoltage (ViSession instrumentHandle, ViInt16 reference, ViReal64 *voltage);

Query Sequence Active

LabVIEW Diagram:



Description:

This vi returns the execution time of the previous sequence run.

The sequence active time has a 10ns resolution +/- 10ns.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

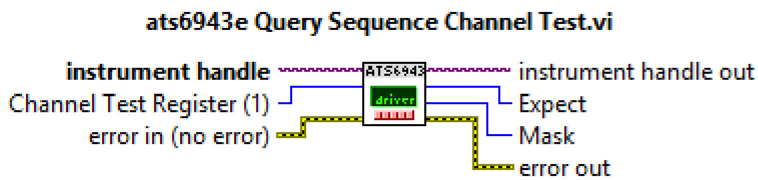
Sequence Active	DBL	Measure result.	0 to 42.94967295

C Function Prototype Form:

ViStatus ats6943e_querySequenceActive (ViSession instrumentHandle, ViReal64 *sequenceActive);

Query Sequence Channel Test

LabVIEW Diagram:



Description:

This vi queries one of the four sequence channel test registers of the selected data sequencer.

Key Parameters:

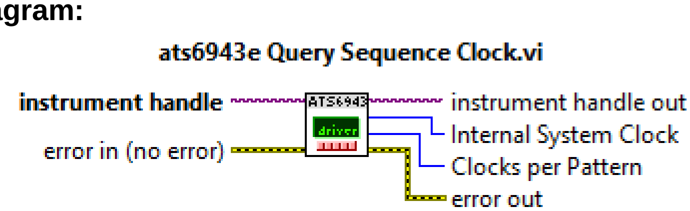
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel Test Register	I16	This control specifies the reference voltage to query.	0 = 5.0 1 = 3.33 2 = 1.66 3 = -1.0
Expect	U32	This control is used to return the channel test expect value.	0 to 42949677296
Mask	U32	This control is used to return the channel test mask value.	0 to 42949677296

C Function Prototype Form:

ViStatus ats6943e_querySequenceChannelTest (ViSession instrumentHandle, ViInt16 channelTestRegister, ViUInt32 *expect, ViUInt32 *mask);

Query Sequence Clock

LabVIEW Diagram:



Description:

This vi queries the clock data for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

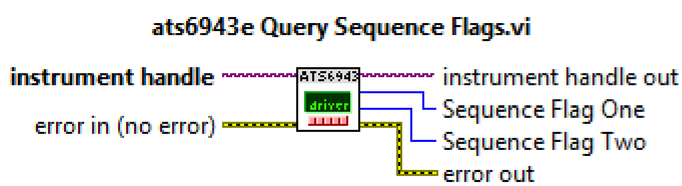
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Internal System Clock	U32	This control is used to return the internal system clock period (T0CLK).	18 to 65551
Clocks Per Pattern	I16	This control is used to return the number of system clocks per pattern clock.	1 to 256

C Function Prototype Form:

ViStatus ats6943e_querySequenceClock (ViSession instrumentHandle, ViInt32 *internalSystemClock_T0CLK, ViInt16 *clocksPerPattern);

Query Sequence Flags

LabVIEW Diagram:



Description:

This vi returns the sequence flag levels for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

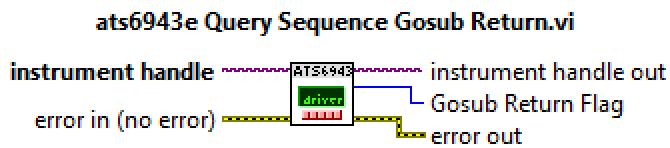
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Sequence Flag One	I16	This control is used to return the level of sequence flag one when this sequence step is executed.	0 = Low 1 = High
Sequence Flag Two	I16	This control is used to return the level of sequence flag two when this sequence step is executed.	0 = Low 1 = High

C Function Prototype Form:

ViStatus ats6943e_querySequenceFlags (ViSession instrumentHandle, ViInt16 *sequenceFlagOne, ViInt16 *sequenceFlagTwo);

Query Sequence Gosub Return

LabVIEW Diagram:



Description:

This vi returns the gosub return flag level for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

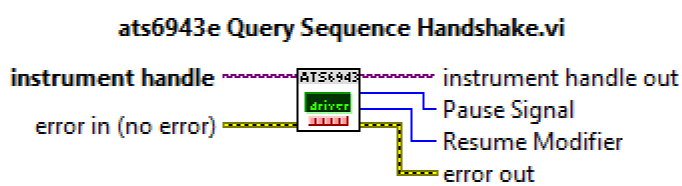
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Gosub Return Flag	I16	This control is used to query the gosub return flag for this sequence step.	0 = False 1 = True

C Function Prototype Form:

ViStatus ats6943e_querySequenceGosubReturn (ViSession instrumentHandle, ViInt16 *gosubReturnFlag);

Query Sequence Handshake

LabVIEW Diagram:



Description:

This vi queries the handshake signal and mode for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pause Signal	I16	This control is used to return the pause signal for the selected sequence step.	0 = None 2 = Pause Trigger 1 True 3 = Pause Trigger 1 Not True

Name	Type	Description	Value
			4 = Pause Trigger 2 True 5 = Pause Trigger 2 Not True 6 = Phase 1 Assert 7 = Phase 1 Return 8 = Phase 2 Assert 9 = Phase 2 Return 10 = Phase 3 Assert 11 = Phase 3 Return 12 = Phase 4 Assert 13 = Phase 4 Return
Resume Modifier	I16	This control is used to return the resume modifier for the selected sequence step.	0 = None 1 = Pattern Delay Timer 1 2 = Pattern Delay Timer 2 3 = Pattern Timeout

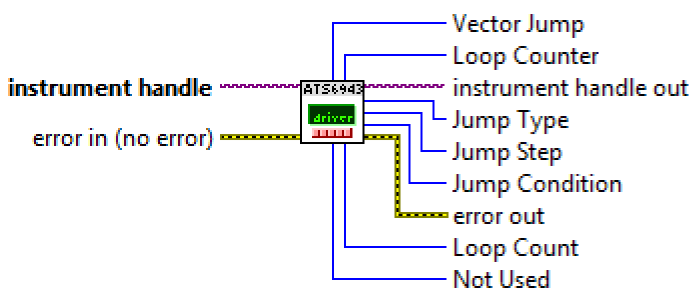
C Function Prototype Form:

ViStatus ats6943e_querySequenceHandshake (ViSession instrumentHandle, ViInt16 *pauseSignal, ViInt16 *resumeModifier);

Query Sequence Jump

LabVIEW Diagram:

ats6943e Query Sequence Jump.vi



Description:

This vi returns the jump parameters for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Vector Jump	I16	This control is used to return the vector jump flag for this sequence step.	0 = False 1 = True
Loop Counter	I16	This control is used to return the loop counter used if a loop count is specified.	0 to 15

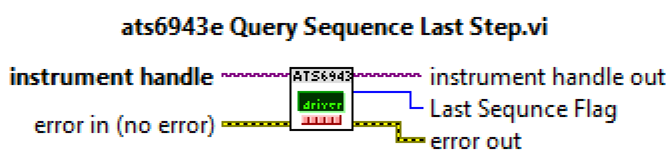
Name	Type	Description	Value
Jump Type	I16	This control is used to return the jump type for the selected sequence step.	0 = None 1 = Normal 2 = Gosub
Jump Step	I16	This control is used to return the sequence step number to jump to.	0 to 4095
Jump Condition	I16	This control is used to return the jump condition for the selected sequence step.	1 = Always 2 = Step Not Pass 3 = Step Not Fail 4 = Step Fail 5 = Step Pass 6 = Sequence Fail 7 = Sequence Pass 8 = Jump Trigger 1 9 = Not Jump Trigger 1 10 = Jump Trigger 2 11 = Not Jump Trigger 2 12 = Jump Trigger 3 13 = Not Jump Trigger 3 14 = Jump Trigger 4 15 = Not Jump Trigger 4
Loop Count	U32	This control is used to return a loop count for the jump step.	0 to 65536 (0 disables jump)
Not Used	I16	This control is not used and is include for legacy support.	

C Function Prototype Form:

ViStatus ats6943e_querySequenceJump (ViSession instrumentHandle, ViInt16 *jumpType, ViInt16 *jumpStep, ViInt16 *jumpCondition, ViInt32 *loopCount, ViInt16 *loopCounter, ViInt16 *notUsed, ViInt16 *vectorJump);

Query Sequence Last Step

LabVIEW Diagram:



Description:

This vi returns the last step flag level for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

Last Step Flag	I16	This control is used to return the last step flag for this sequence step.	0 = False 1 = True

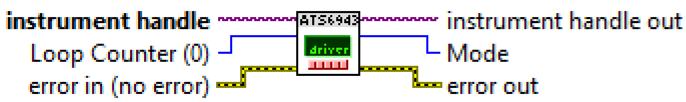
C Function Prototype Form:

ViStatus ats6943e_querySequenceLastStep (ViSession instrumentHandle, ViInt16 *lastSequnceFlag);

Query Sequence Loop Mode

LabVIEW Diagram:

ats6943e Query Sequence Loop Mode.vi



Description:

This vi returns the loop counter mode.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Loop Counter	I16	This control specifies the loop counter to query. 16 selects all counters.	0 to 16
Mode	I16	This control is used to return the loop counter mode.	0 = Reload 1 = Disable If Loop Counter set to 16 then Bit 0 = Counter 0 Mode Bit 1 = Counter 1 Mode . . Bit 15 = Counter 15 Mode

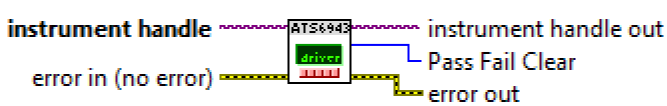
C Function Prototype Form:

ViStatus ats6943e_querySequenceLoopMode (ViSession instrumentHandle, ViInt16 loopCounter, ViInt16 *mode);

Query Sequence Pass Fail Clear

LabVIEW Diagram:

ats6943e Query Sequence Pass Fail Clear.vi



Description:

This vi returns the sequence step pass fail clear mode for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pass Fail Clear	I16	This control is used to return the pass fail clear mode for the selected sequence step.	0 = Default 1 = Mask

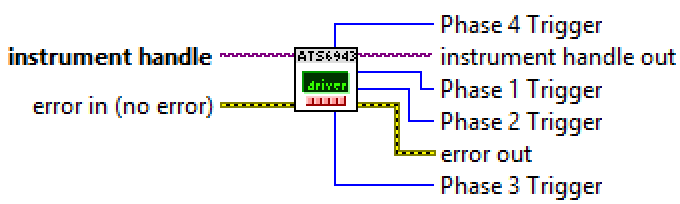
C Function Prototype Form:

ViStatus ats6943e_querySequencePassFailClear (ViSession instrumentHandle, ViInt16 *passFailClear);

Query Sequence Phase Trigger

LabVIEW Diagram:

ats6943e Query Sequence Phase Trigger.vi



Description:

This vi returns the phase triggers for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

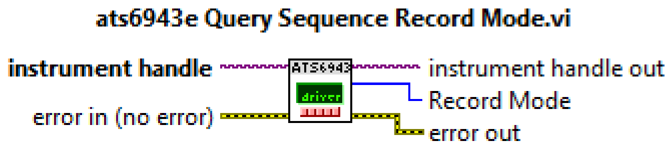
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Phase 1 Trigger	I16	This control is used to return the group 1 phase timing trigger.	0 = System Clock 1 = Pattern Clock
Phase 2 Trigger	I16	This control is used to return the group 2 phase timing trigger.	0 = System Clock 1 = Pattern Clock
Phase 3 Trigger	I16	This control is used to return the group 3 phase timing trigger.	0 = System Clock 1 = Pattern Clock
Phase 4 Trigger	I16	This control is used to return the group 4 phase timing trigger.	0 = System Clock 1 = Pattern Clock

C Function Prototype Form:

ViStatus ats6943e_querySequencePhaseTrigger (ViSession instrumentHandle, ViInt16 *phase1Trigger, ViInt16 *phase2Trigger, ViInt16 *phase3Trigger, ViInt16 *phase4Trigger);

Query Sequence Record Mode

LabVIEW Diagram:



Description:

This vi returns the record mode for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

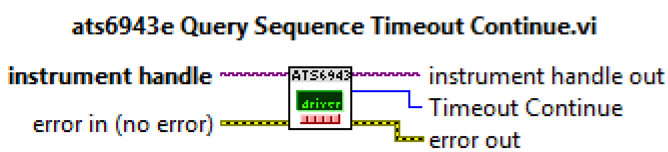
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Record Mode	I16	This control is used to return the record mode for the selected sequence step.	0 = None 1 = Record Count 2 = Record Error 3 = Record Response

C Function Prototype Form:

ViStatus ats6943e_querySequenceRecordMode (ViSession instrumentHandle, ViInt16 *recordMode);

Query Sequence Timeout Continue

LabVIEW Diagram:



Description:

This vi returns the timeout continue flag level for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Timeout Continue	I16	This control is used to return the timeout continue flag for this sequence step.	0 = False 1 = True

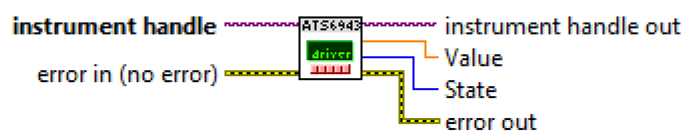
C Function Prototype Form:

ViStatus ats6943e_querySequenceTimeoutContinue (ViSession instrumentHandle, ViInt16 *timeoutContinue);

Query Sequence Timer

LabVIEW Diagram:

ats6943e Query Sequence Timer.vi



Description:

This vi returns the sequence timeout value and state.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
State	I16	This control is used to return the sequence timeout state.	0 = Disabled 1 = Enabled
Value	DBL	This control is used to return the sequence timeout value.	0 to 42.949672970

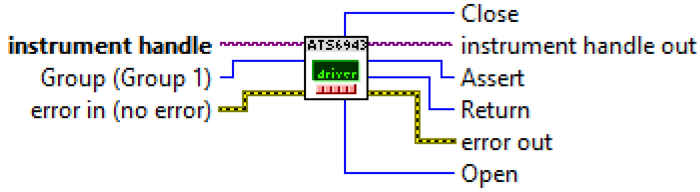
C Function Prototype Form:

ViStatus ats6943e_querySequenceTimer (ViSession instrumentHandle, ViReal64 *value, ViInt16 *state);

Query Sequence Timing Data

LabVIEW Diagram:

ats6943e Query Sequence Timing Data.vi



Description:

This vi returns the phase and window settings for the selected sequence step and group for non-indexed timing mode.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Group	I16	This control specifies which timing group to query.	0 = Group 1 1 = Group 2 2 = Group 3 3 = Group 4
Assert	I32	This control is used to query the phase assert time.	0 to 65535
Return	I32	This control is used to query the phase return time.	0 to 65535
Open	I32	This control is used to query the window open time.	0 to 65535
Close	I32	This control is used to query the window close time.	0 to 65535

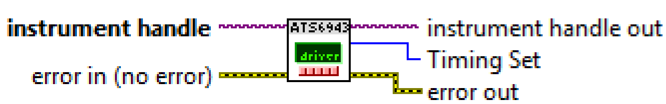
C Function Prototype Form:

ViStatus ats6943e_querySequenceTimingData (ViSession instrumentHandle, ViInt16 group, ViInt32 *assert, ViInt32 *return, ViInt32 *open, ViInt32 *close);

Query Sequence Timing Set

LabVIEW Diagram:

ats6943e Query Sequence Timing Set.vi



Description:

This vi queries the timing set number for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Timing Set	I16	This control is used to return the timing set number.	0 to 255

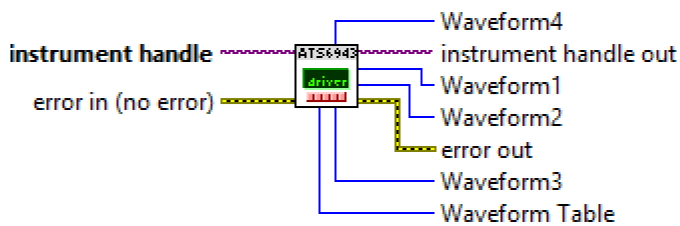
C Function Prototype Form:

ViStatus ats6943e_querySequenceTimingSet (ViSession instrumentHandle, ViInt16 *timingSet);

Query Sequence Waveform

LabVIEW Diagram:

ats6943e Query Sequence Waveform.vi



Description:

This vi returns the waveform enables as well as the waveform table for the selected sequence step.

Waveform1 replaces Phase 4.

Waveform2 replaces Window 4.

Waveform3 replaces Phase 3.

Waveform4 replaces Window 3.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Waveform 1	I16	This control is used to return the enable state of Waveform1.	0 = Disabled 1 = Enabled
Waveform 2	I16	This control is used to return the enable state of Waveform2.	0 = Disabled 1 = Enabled
Waveform 3	I16	This control is used to return the enable state of Waveform3.	0 = Disabled 1 = Enabled

Waveform 4	I16	This control is used to return the enable state of Waveform4.	0 = Disabled 1 = Enabled
Waveform Table	I16	This control returns the waveform table.	1 to 16

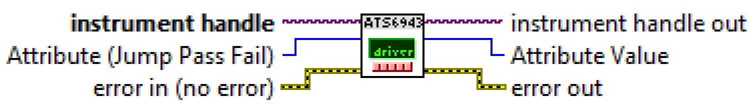
C Function Prototype Form:

ViStatus ats6943e_querySequenceWaveform (ViSession instrumentHandle, ViInt16 *waveform1, ViInt16 *waveform2, ViInt16 *waveform3, ViInt16 *waveform4, ViInt16 *waveformTable);

Query Sequencer Attribute

LabVIEW Diagram:

ats6943e Query Sequencer Attrubute.vi



Description:

This vi returns the sequencer attribute values.

Attribute Values:

Attribute	Value
Jump Pass Fail	0 = Normal 1 = Legacy
Phase 3 Mode	0 = Normal 1 = Jump Trigger 1
Window 3 Mode	0 = Normal 1 = Jump Triger 2
Window 3 Delay	0 to 15
CRC Preload	0 to hex FFFFFFFF
CRC Feedback	0 to hex FFFFFFFF

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Attribute	I16	This control specifies the sequencer attribute to query.	0 = Jump Pass Fail 1 = Phase 3 Mode 2 = Window 3 Mode 3 = Window 3 Delay 4 = CRC Preload 5 = CRC Feedback
Attribute Value	U32	This control returns the waveform table.	See description above.

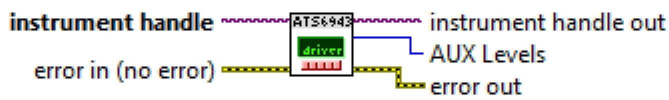
C Function Prototype Form:

ViStatus ats6943e_querySequencerAttribute (ViSession instrumentHandle, ViInt16 attribute, ViUInt32 *attributeValue);

Query Sequencer Aux

LabVIEW Diagram:

ats6943e Query Sequencer Aux.vi



Description:

This vi returns the sequencer AUX level bits.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Aux Levels	I16	This control returns the sequencer AUX level bits.	Bit 0 = AUX1 Good 0 Bit 1 = AUX1 Good 1 Bit 2 = AUX2 Good 1 Bit 3 = AUX3 Good 1 Bit 4 = AUX4 Good 1 Bit 5 = AUX5 Good 1 Bit 6 = AUX6 Good 1 Bit 7 = AUX7 Good 1 Bit 8 = AUX8 Good 1 Bit 9 = AUX9 Good 1 Bit 10 = AUX10 Good 1 Bit 11 = AUX11 Good 1 Bit 12 = AUX12 Good 1

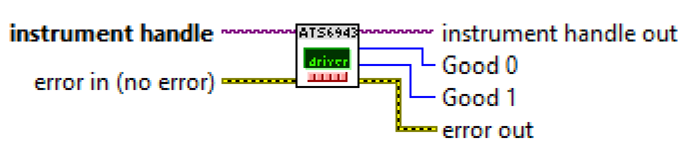
C Function Prototype Form:

ViStatus ats6943e_querySequencerAux (ViSession instrumentHandle, ViInt16 *AUXLevels);

Query Sequencer Channels

LabVIEW Diagram:

ats6943e Query Sequencer Channels.vi



Description:

This vi returns the static readback of the front-end channels.

Good 0 bit set high indicates that the channel is < CVL.

Good 1 bit set high indicates that the channel is > CVH.

Key Parameters:

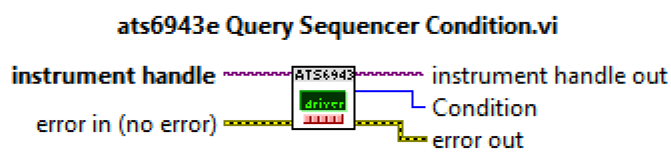
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Good 0	U32	This control returns the sequencer good 0 bits.	Bit 0 = Good 0 CH1 . . Bit 31 = Good 0 CH32
Good 1	U32	This control returns the sequencer good 1 bits.	Bit 0 = Good 1 CH1 . . Bit 31 = Good 1 CH32

C Function Prototype Form:

ViStatus ats6943e_querySequencerChannels (ViSession instrumentHandle, ViUInt32 *good0, ViUInt32 *good1);

Query Sequencer Condition

LabVIEW Diagram:



Description:

This vi returns the sequencer condition register.

Unlike the event register, the condition register contains the current status of the hardware. The condition register is cleared prior to sequence execution.

Bit	Label	Description
0	IACT	1 = Idle sequence occurred
1	SACT	1 = Primary sequence occurred
2	EHALT	1 = External halt occurred
3	BERR	1 = Burst error occurred
4	JUMP	1 = Jump occurred
5	NU	Not used.
6	WDTO	1 = Watchdog timeout occurred

Bit	Label	Description
7	STO	1 = Sequence timeout
8	PFIFO	1 = Pipeline FIFO error occurred
9	SYNC	1 = Inter module sync error occurred
10	PWG	1 = Phase/Window glitch occurred
11	WCF	1 = Window capture fault occurred
12	PTO	1 = Pattern timeout occurred
13	PAUSE	1 = Pause occurred
14	ESTOP	1 = External stop occurred
15	FSERR	1 = Frequency synthesizer error
16	MSUB	1 = Multiple subroutine calls
17	RNS	1 = Return without subroutine
18	SEND	1 = Sequence end in subroutine
19	ISC	1 = Idle sequence complete
20	ESC	1 = External halt occurred
21	ECG	1 = External halt occurred
22	ECTO	1 = External halt occurred
23	DFLT	1 = Driver Fault occurred
24	RAOV	1 = Record address overflow
25	RIOV	1 = Record index overflow
26	NU	Not used.
27	NU	Not used.
28	FCDR	1 = Counter data ready
29	TDR	1 = Timer Data ready
30	NU	Not used.
31	ESF	1 = External start fault

Bits not listed above are unused.

Key Parameters:

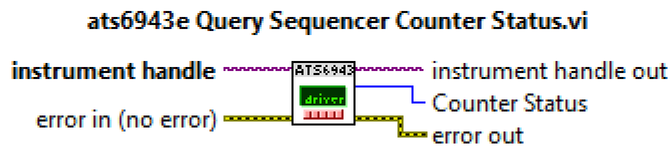
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Condition	U32	This control returns the condition register.	See table above.

C Function Prototype Form:

ViStatus ats6943e_querySequencerCondition (ViSession instrumentHandle, ViInt32 *condition);

Query Sequencer Counter Status

LabVIEW Diagram:



Description:

This control returns the counter status bits.

Counter status bit set high indicates the counter is active.

Key Parameters:

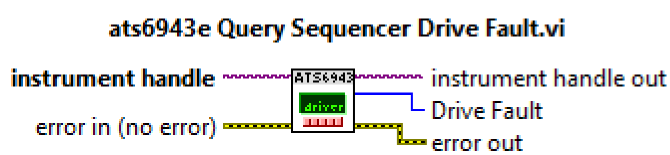
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Counter Status	U32	This control returns the counter status bits.	Bit 0 = Counter 0 Active . . Bit 15 = Counter 15 Active

C Function Prototype Form:

ViStatus ats6943e_querySequencerCounterStatus (ViSession instrumentHandle, ViInt32 *counterStatus);

Query Sequencer Drive Fault

LabVIEW Diagram:



Description:

This vi returns the sequencer drive fault bits.

Key Parameters:

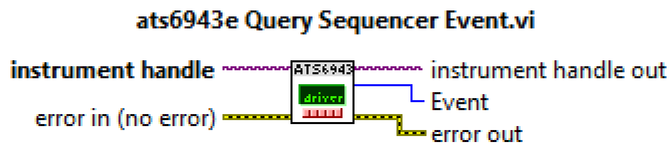
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Dirve Fault	U32	This control returns the sequencer drive fault bits.	Bit 0 = Drive Fault CH1 . . Bit 31 = Drive Fault CH32

C Function Prototype Form:

ViStatus ats6943e_querySequencerDriveFault (ViSession instrumentHandle, ViUInt32 *driveFault);

Query Sequencer Event

LabVIEW Diagram:



Description:

This vi returns the sequencer event register.

A bit set in the event register indicates a positive transition occurred in the corresponding condition register bit.

All bits are cleared automatically after reading.

See **Query Sequencer Condition** for register bit definitions.

Key Parameters:

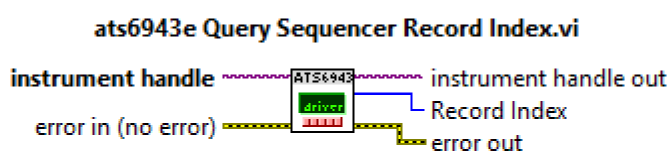
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Event	U32	This control returns the event register.	See description above.

C Function Prototype Form:

ViStatus ats6943e_queryFrontEndEvent (ViSession instrumentHandle, ViInt16 *event);

Query Sequencer Record Index

LabVIEW Diagram:



Description:

This vi returns the sequencer record index count.

Key Parameters:

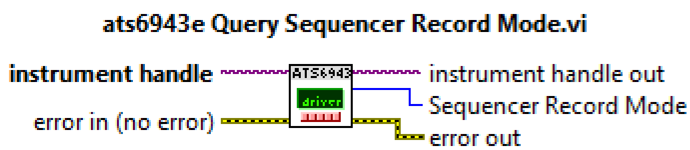
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Record Index	I16	This vi returns the sequencer record index count.	0 to 1023

C Function Prototype Form:

ViStatus ats6943e_querySequencerRecordIndex (ViSession instrumentHandle, ViInt16 *recordIndex);

Query Sequencer Record Mode

LabVIEW Diagram:



Description:

This vi returns the sequencer record mode setting.

Key Parameters:

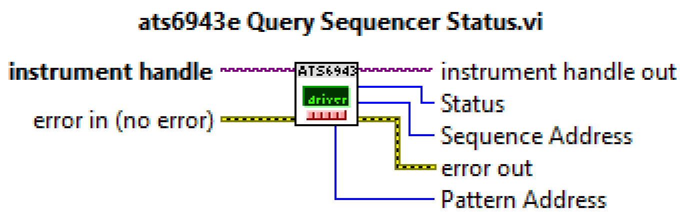
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Sequencer Record Mode	I16	This control is used to return the sequencer record mode setting.	0 = Record Disabled 1 = Record Non-Error

C Function Prototype Form:

ViStatus ats6943e_querySequencerRecordMode (ViSession instrumentHandle, ViInt16 *sequencerRecordMode);

Query Sequencer Status

LabVIEW Diagram:



Description:

This vi returns the sequencer status.

Key Parameters:

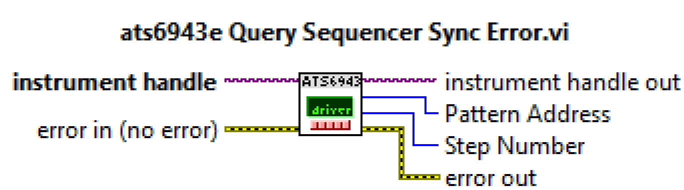
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Status	I16	This control returns the sequencer status bits.	Bit 0 = Sequencer is paused Bit 2 = 500MHz clock OK Bit 3 = ETB Delay Locked Bit 4 = Internal Stop Bit 5 = External Stop Bit 6 = Internal Start Bit 7 = External Start Bit 8 = Sequencer is halted Bit 9 = Single Step Bit 10 = Idle Active Bit 11 = Sequence Active Bit 12 = Drivers Enabled Bit 13 = External Halt
Sequence Address	I16	This control returns the current sequencer step address.	0 to 4095
Pattern Address	I32	This control returns the current pattern address.	0 to 262143

C Function Prototype Form:

ViStatus ats6943e_querySequencerStatus (ViSession instrumentHandle, ViInt16 *status, ViInt16 *sequenceAddress, ViInt32 *patternAddress);

Query Sequencer Sync Error

LabVIEW Diagram:



Description:

This vi returns the sequencer sync error data.

When a DTS sync error event is generated (Bit 9 in sequencer event register), the pattern address and step number are recorded.

Key Parameters:

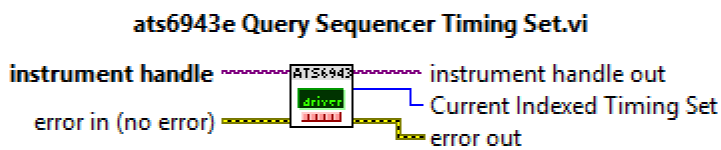
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pattern Address	I32	This control returns the sequencer status bits.	0 to 262143
Step Number	I16	This control returns the current sequencer step address.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_querySequencerSyncError (ViSession instrumentHandle, ViInt32 *patternAddress, ViInt16 *stepNumber);

Query Sequencer Timing Set

LabVIEW Diagram:



Description:

This vi returns the current indexed timing set number.

Key Parameters:

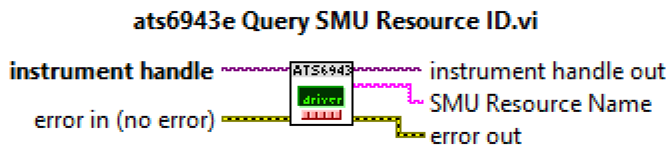
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Timing Set	I16	This control returns the current indexed timing set.	0 to 255

C Function Prototype Form:

ViStatus ats6943e_querySequencerTimingSet (ViSession instrumentHandle, ViInt16 *currentIndexedTimingSet);

Query SMU Resource ID

LabVIEW Diagram:



Description:

This vi returns the SMU resource name used for calibrating the voltage reference, PMU and Active Load.

Key Parameters:

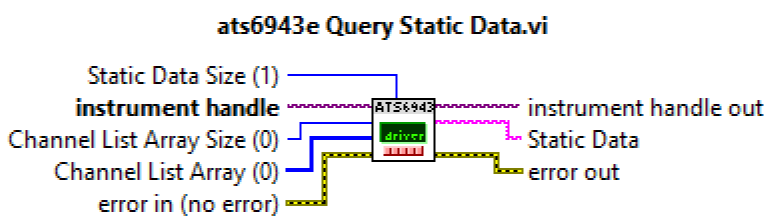
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
Calibration Date	abc	This control contains the SMU instrument resource name.	ASCII string returned will be <= 64 characters.

C Function Prototype Form:

ViStatus ats6943e_querySmuResourceId (ViSession instrumentHandle, ViString *SMUResourceName);

Query Static Data

LabVIEW Diagram:



Description:

This vi returns the static pattern.

Static data is expressed as an ASCII code described below:

Pin Action	Static Data Code
Disable channel	'Z'
Drive Low	'0'
Drive High	'1'

The channel list specifies which pins to read and the order with respect to the static data.

There is a one to one correspondence between the data in channel list array and the static data array. The static code in index n of the static data array contains the static output for the channel specified in index n of the channel list array, i.e., for every channel in the channel list array n = 0 to (channel list size - 1)

Static Code @ StaticData[n] = Static Output for Channel @ ChannelArray[n]

Key Parameters:

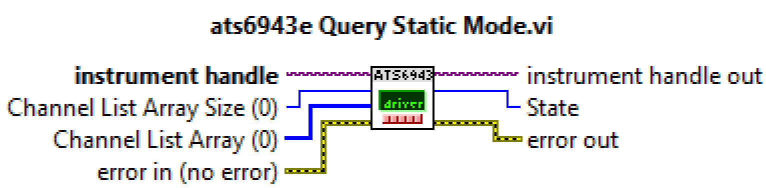
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Static Data Size	I16	This control specifies the number of pin codes in the static data array.	1 to 32 Should be >= Channel List Array Size.
Static Data	abc	This control contains the static output data	ASCII string returned will be = Static Data Size characters.

C Function Prototype Form:

ViStatus ats6943e_queryStaticData (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 staticDataSize, ViChar staticData[]);

Query Static Mode

LabVIEW Diagram:



Description:

This vi returns the static mode enable for the specified channels.

Key Parameters:

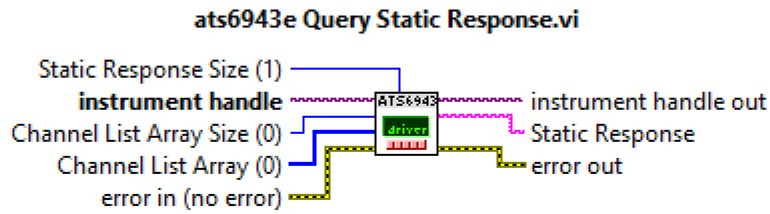
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
State	I16	This control is used to return the static mode state.	0 = Off 1 = On

C Function Prototype Form:

ViStatus ats6943e_queryStaticMode (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 *state);

Query Static Response

LabVIEW Diagram:



Description:

This vi returns the static response data.

Static response data is expressed as an ASCII code described below:

Pin State	Static Response Code
Between	‘B’
Low	‘L’
High	‘H’
Unknown	‘?’

There is a one to one correspondence between the data in static response array and the data in the channel list array. The static code in index n of static response array will be the response from the channel specified in index n of the channel list array, i.e., for every channel in the channel list array n = 0 to (channel list size - 1)

Channel @ ChannelArray[n] = Static Code @ StaticResponse[n]

If the static response array does not contain as many elements as the channel list array, then the static response will be set to '?'. For example if there are 6 channels specified in the channel list and only two static data codes in the pattern array then the following will be programmed;

Channel @ ChannelArray[0] = Static Code @ StaticResponse[0]

Channel @ ChannelArray[1] = Static Code @ StaticResponse[1]

Channel @ ChannelArray[2] = '?'

Channel @ ChannelArray[3] = '?'

Channel @ ChannelArray[4] = '?'

Channel @ ChannelArray[5] = '?' The channel list specifies which pins to read and the order with respect to the static response.

Key Parameters:

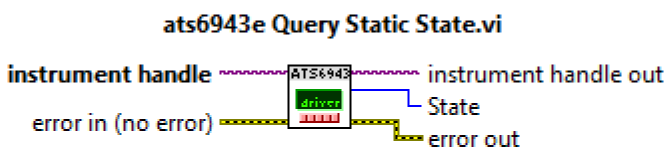
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Static Response Size	I16	This control specifies the number of pin codes in the static data array.	1 to 32 Should be >= Channel List Array Size.
Static Response	abc	This control contains the static response data	ASCII string returned will be = Static Data Size characters.

C Function Prototype Form:

ViStatus ats6943e_queryStaticResponse (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 staticResponseSize, ViChar staticResponse[]);

Query Static State

LabVIEW Diagram:



Description:

This vi returns the static state.

Key Parameters:

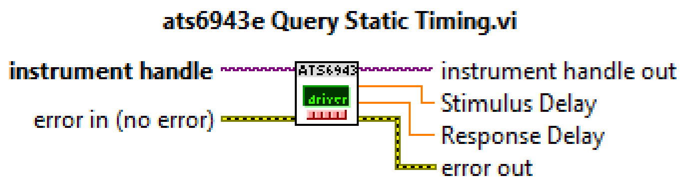
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
State	I16	This control is used to return the static state.	0 = Disabled 1 = Enabled

C Function Prototype Form:

ViStatus ats6943e_queryStaticState (ViSession instrumentHandle, ViInt16 *state);

Query Static Timing

LabVIEW Diagram:



Description:

This vi returns the static timing for the specified sequencer.

Key Parameters:

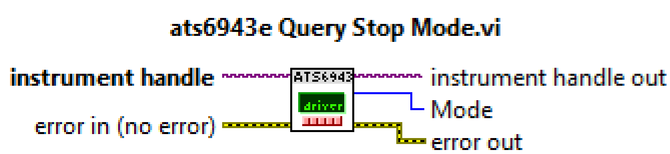
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Stimulus Delay	DBL	This control is included for legacy support.	0.0
Response Delay	DBL	This control is used to return the static response delay.	0.0 to 6.5ms

C Function Prototype Form:

ViStatus ats6943e_queryStaticTiming (ViSession instrumentHandle, ViReal64 *stimulusDelay, ViReal64 *responseDelay);

Query Stop Mode

LabVIEW Diagram:



Description:

This vi returns the stop mode.

Key Parameters:

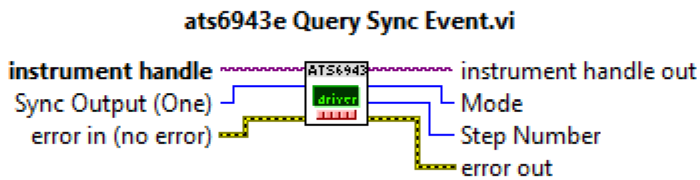
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Mode	I16	This control is used to return the stop mode.	0 = Disabled 1 = Stop after next pattern 2 = Stop looping and continue 3 = stop at end of sequence

C Function Prototype Form:

ViStatus ats6943e_queryStopMode (ViSession instrumentHandle, ViInt16 *mode);

Query Sync Event

LabVIEW Diagram:



Description:

This vi returns the sync output event.

Key Parameters:

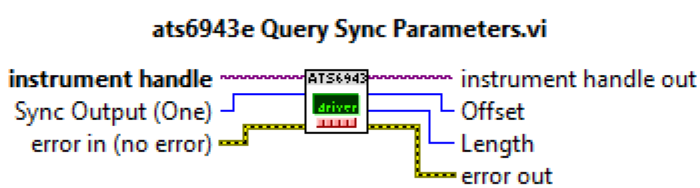
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Sync Output	◂▸	This control specifies which sync output to query.	0 = SYNC1 1 = SYNC2
Mode	I16	This control is used to return the sync output mode.	0 = Start 1 = Single Step 2 = Continuous Step
Step Number	I16	This control is used to return the sync step number.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_querySyncEvent (ViSession instrumentHandle, ViInt16 syncOutput, ViInt16 *mode, ViInt16 *stepNumber);

Query Sync Parameters

LabVIEW Diagram:



Description:

This vi returns the sync output parameters.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Sync Output	◄►	This control specifies which sync output to query.	0 = SYNC1 1 = SYNC2
Offset	I32	This control is used to return the sync pattern offset.	0 to 1048575
Length	I16	This control is used to return the sync step number.	0 to 4095

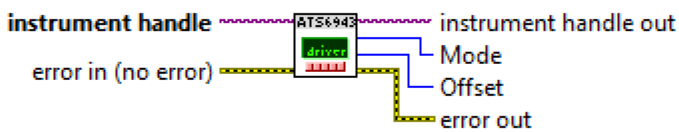
C Function Prototype Form:

ViStatus ats6943e_querySyncParameters (ViSession instrumentHandle, ViInt16 syncOutput, ViInt32 *offset, ViInt16 *length);

Query System Clock Parameters

LabVIEW Diagram:

ats6943e Query System Clock Parameters.vi



Description:

This vi returns the external system clock parameters of the sequencer.

Key Parameters:

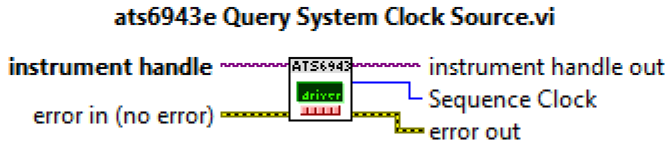
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Mode	I16	This control is used to return the system clock mode.	0 = None 1 = Rising Edge 2 = Falling Edge 3 = Both Edges 4 = Divide by 2 Rising Edge 5 = Divide by 2 Falling Edge
Offset	I16	This control returns the system clock offset.	0 to 65535

C Function Prototype Form:

ViStatus ats6943e_querySystemClockParameters (ViSession instrumentHandle, ViInt16 *mode, ViInt16 *offset);

Query System Clock Source

LabVIEW Diagram:



Description:

This vi returns the system clock source of the sequencer.

Refer to **Set System Clock Source** for returned values.

Key Parameters:

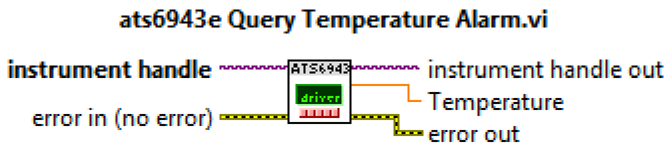
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Sequence Clock	I16	This control is used to return the system clock source.	See description above

C Function Prototype Form:

ViStatus ats6943e_querySystemClockSource (ViSession instrumentHandle, ViInt16 *sequenceClock);

Query Temperature Alarm

LabVIEW Diagram:



Description:

This vi queries the temperature alarm trip point.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Temperature	DBL	This control returns the temperature at which the alarm will trip.	70.0 to 130.0

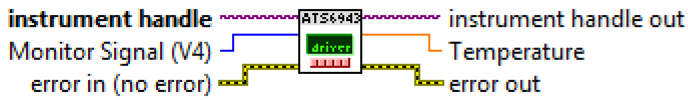
C Function Prototype Form:

ViStatus ats6943e_queryTemperatureAlarm (ViSession instrumentHandle, ViReal64 *temperature_C);

Query Temperature Monitor

LabVIEW Diagram:

ats6943e Query Temperature Monitor.vi



Description:

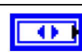
This vi returns the temperature of the specified monitor signal.

There are three digital board signals.

The digital board monitor signal are:

1. U15 near the IO heat sink.
2. U16 near the Artix FPGA
3. U35 near the V4 FPGA

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Monitor Signal		This control returns the temperature at which the alarm will trip.	0 = IO Heatsink 1 = Artix (A7) 2 = V4
Temperature		This control return the temperature of the selected signal.	-128.0 to 128.0

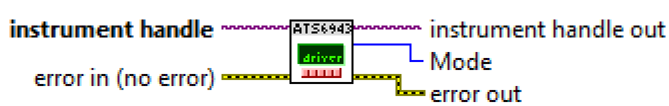
C Function Prototype Form:

ViStatus ats6943e_queryTemperatureMonitor (ViSession instrumentHandle, ViInt16 monitorSignal, ViReal64 *temperature);

Query Timing Mode

LabVIEW Diagram:

ats6943e Query Timing Mode.vi



Description:

This vi returns the timing mode of sequencer.

Key Parameters:

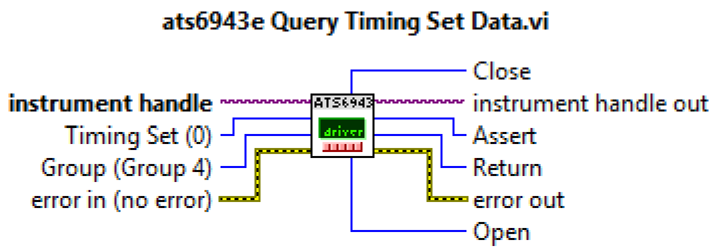
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Mode	I16	This control is used to return which timing mode is set.	0 = Per Step Multi 1 = Per Step Single 2 = Indexed

C Function Prototype Form:

ViStatus ats6943e_queryTimingMode (ViSession instrumentHandle, ViInt16 *mode);

Query Timing Set Data

LabVIEW Diagram:



Description:

This vi returns the stimulus and capture settings for the specified timing set and group.

Key Parameters:

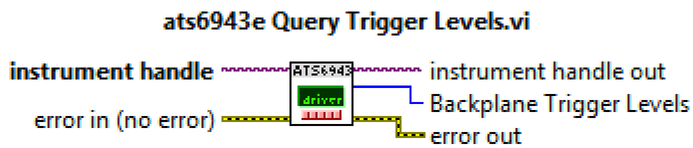
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Timing Set	I16	This control specifies the timing set number to query.	0 to 255
Group	I16	This control specifies which timing group to query.	0 = Group 1 1 = Group 2 2 = Group 3 3 = Group 4
Assert	I32	This control is used to query the phase assert time.	0 to 65535
Return	I32	This control is used to query the phase return time.	0 to 65535
Open	I32	This control is used to query the window open time.	0 to 65535
Close	I32	This control is used to query the window close time.	0 to 65535

C Function Prototype Form:

ViStatus ats6943e_queryTimingSetData (ViSession instrumentHandle, ViInt16 timingSet, ViInt16 group, ViInt32 *assert, ViInt32 *return, ViInt32 *open, ViInt32 *close);

Query Trigger Levels

LabVIEW Diagram:



Description:

This vi returns the trigger level bits of the PXIe 6943.

Key Parameters:

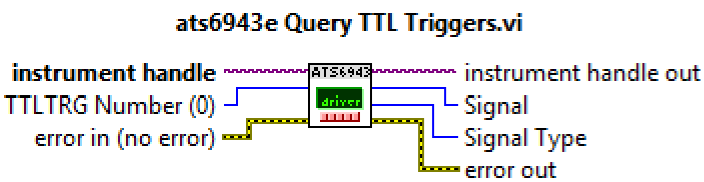
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Backplane Trigger Levels	I16	This control returns the backplane trigger bits.	Bit 0 = 1, TTLTRG0 Low . . Bit 7 = 1, TTLTRG7 Low

C Function Prototype Form:

ViStatus ats6943e_queryTriggerLevels (ViSession instrumentHandle, ViInt16 *backplaneTriggerLevels);

Query TTL Triggers

LabVIEW Diagram:



Description:

This vi returns the TTLTRG signal source and routing.

Refer to **Set TTL Triggers** for returned values.

Key Parameters:

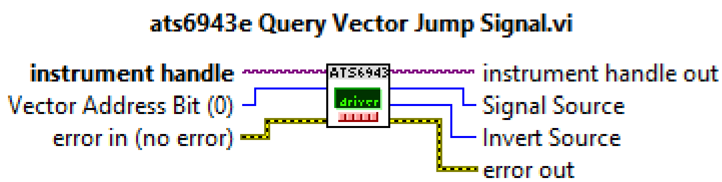
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger	I16	This control specifies which TTL trigger to query.	0 to 7
Signal	I16	This control is used to return the selected TTLTRG signal source.	See description above.
Signal Type	I16	This control is used return the signal type.	See description above.

C Function Prototype Form:

ViStatus ats6943e_queryTtlTriggers (ViSession instrumentHandle, ViInt16 TTLTRGNumber, ViInt16 *signal, ViInt16 *signalType);

Query Vector Jump Signal

LabVIEW Diagram:



Description:

This vi queries one of the four vector jump signals of the selected data sequencer.

Refer to **Set Vector Jump Signal** for returned values.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Vector Address Bit	I16	This control specifies the phase resume trigger to query.	0 to 3
Signal Source	I16	This control is used to return the vector jump signal source.	See description above.
Invert Source	I16	This control is used to return the vector jump signal source inverter.	See description above.

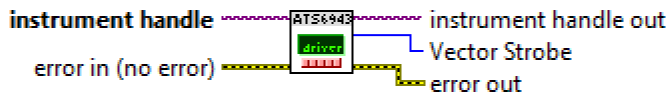
C Function Prototype Form:

ViStatus ats6943e_queryVectorJumpSignal (ViSession instrumentHandle, ViInt16 vectorAddressBit_VAn, ViInt16 *signalSource, ViInt16 *invertSource);

Query Vector Jump Strobe

LabVIEW Diagram:

ats6943e Query Vector Jump Strobe.vi



Description:

This vi returns the vector strobe setting.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Vector Strobe	I16	This control is used to return the vector strobe signal.	0 = Window 1 1 = Window 2 2 = Window 3 3 = Window 4

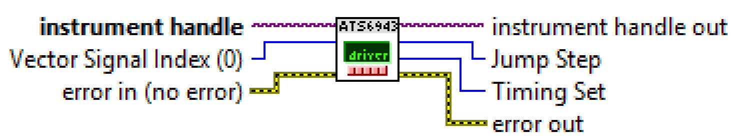
C Function Prototype Form:

ViStatus ats6943e_queryVectorJumpStrobe (ViSession instrumentHandle, ViInt16 vectorStrobe);

Query Vector Jump Table

LabVIEW Diagram:

ats6943e Query Vector Jump Table.vi



Description:

This vi returns one of the sixteen vector jump table entries for the selected data sequencer.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Vector Signal Index	I16	This control specifies the phase resume trigger to query.	0 to 15
Jump Step	I16	This control is used to return the phase resume source.	0 to 4095

Name	Type	Description	Value
Timing Set	I16	This control is used to return the pause trigger resume source inverter.	0 to 255

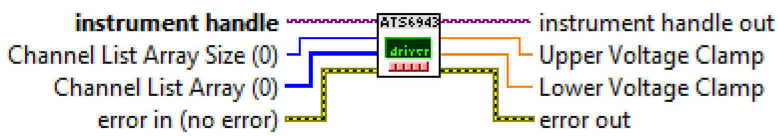
C Function Prototype Form:

ViStatus ats6943e_queryVectorJumpTable (ViSession instrumentHandle, ViInt16 vectorSignalIndex, ViInt16 *jumpStep, ViInt16 *timingSet);

Query Voltage Clamps

LabVIEW Diagram:

ats6943e Query Voltage Clamps.vi



Description:

This function returns the upper and lower voltage clamps for the specified channels.

Key Parameters:

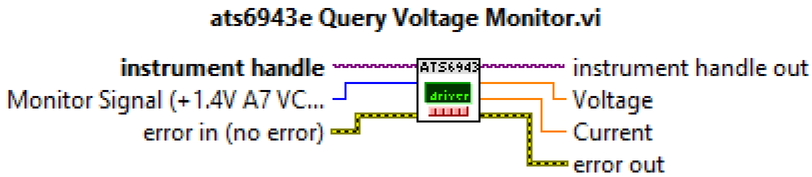
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to query.	1 to 32
Upper Voltage Clamp	DBL	This control is used to return the upper voltage clamp value.	-2.5 to 7.5
Lower Voltage Clamp	DBL	This control is used to return the lower voltage clamp value.	-2.5 to 7.5

C Function Prototype Form:

ViStatus ats6943e_queryPmuVClamps (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 *upperVoltageClamp, ViReal64 *lowerVoltageClamp);

Query Voltage Monitor

LabVIEW Diagram:



Description:

This vi returns the voltage and current of the specified monitor signal.

Key Parameters:

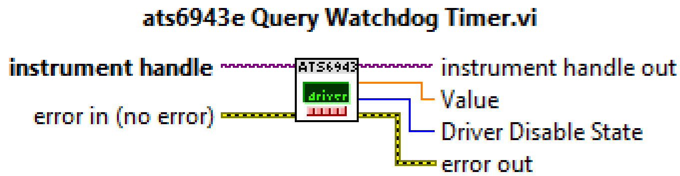
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Monitor Signal		This control specifies the voltage monitor to query.	0 = DB A7 1.0V VCCINT 1 = DB A7 1.8V VCCAUZ 2 = DB A7 3.3V IO 3 = DB A7 1.4V VCCLDO 4 = DB A7 1.0V MGTA VCC 5 = DB A7 1.2V MGTA VTT 6 = DB V4 2.5V IO 7 = DB V4 1.2V VCC 8 = DB 3.3V PXI DB 9 = DB 12V PXI DB 10 = DB 3.3V PXI DR 11 = DB 12V PXI DR 13 = DR 3.3V 14 = DR 5.0V 15 = DR VCC1 16 = DR VCC2 17 = DR HV-VCC1 18 = DR HV-VCC2
Voltage		This control returns the voltage of the selected signal.	0 to 28
Current		This control returns the current of the selected signal.	-32 to +32

C Function Prototype Form:

ViStatus ats6943e_queryVoltageMonitor (ViSession instrumentHandle, ViInt16 monitorSignal, ViReal64 *voltage, ViReal64 *current);

Query Watchdog Timer

LabVIEW Diagram:



Description:

This vi returns the watchdog timeout value and state.

Key Parameters:

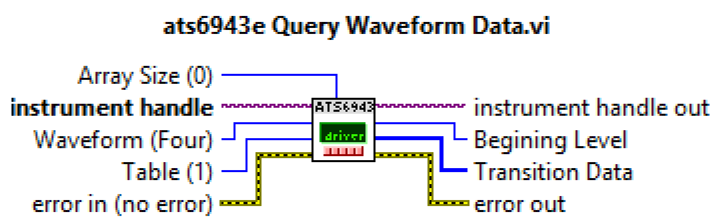
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Value	DBL	This control is used to return the watchdog timeout value.	40ns to 4000s
Driver Disable State	I16	This control is used to return the watchdog timer driver disable state.	0 = Off 1 = On

C Function Prototype Form:

ViStatus ats6943e_queryWatchdogTimer (ViSession instrumentHandle, ViReal64 *value, ViInt16 *driverDisableState);

Query Waveform Data

LabVIEW Diagram:



Description:

This vi queries the specified waveform table data.

The transition data contains the state change timing in increments of ½ MCLK period.

For example if the MCLK is set to 500MHz, beginning level is 0 and the transition data is [5, 10, 20], then the waveform will start low, go high at 5ns, go low at 10ns and finally go high at 20ns.

Key Parameters:

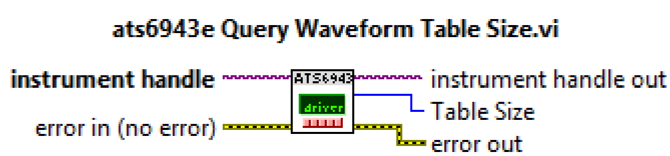
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Waveform	◄►	This control specifies which waveform to query.	0 = Waveform 1 1 = Waveform 2 2 = Waveform 3 3 = Waveform 4
Table	I16	This control specifies the table number to query.	1 to 16
Array Size	I32	This control is used to indicate the number of elements in the transition data array.	0 to 16384
Beginning Level	I16	This control is used to return the beginning level of the waveform data.	0 to 1
Transition Data	[I32]	This control is used to return the transition bits of the specified waveform table.	See description above

C Function Prototype Form:

ViStatus ats6943e_queryWaveformData (ViSession instrumentHandle, ViInt16 waveform, ViInt16 table, ViInt16 *beginingLevel, ViInt32 arraySize, ViInt32 transitionData[]);

Query Waveform Table Size

LabVIEW Diagram:



Description:

This vi returns the waveform table size.

Key Parameters:

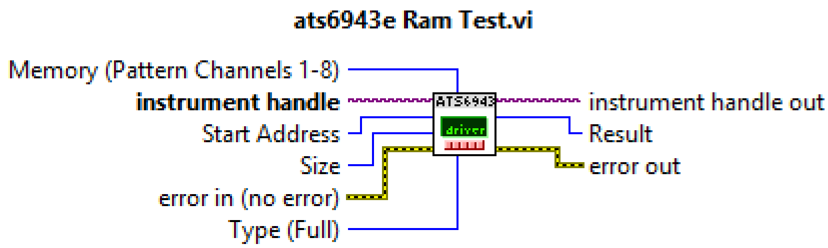
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Table Size	I16	This control is used to return the waveform table size.	0 = 16 Tables of 1024 bits 1 = 8 tables of 2048 bits 2 = 4 tables of 4096 bits 3 = 2 tables of 8192 bits 4 = 1 table of 16384 bits

C Function Prototype Form:

ViStatus ats6943e_queryWaveformTableSize (ViSession instrumentHandle, ViInt16 *tableSize);

Ram Test

LabVIEW Diagram:



Description:

This vi performs a RAM test algorithm at the specified address for the specified size.

The RAM test algorithm is performed in 4 steps:

1. Write/verify a walking 1 pattern at the start address to check for stuck or shorted data bits.
2. Write/verify all zeros to the entire space to check for stuck at one memory bits.
3. Write/verify all ones to the entire space to check for stuck at zero memory bits.
4. Perform a march RAM test using AA and 55 patterns to check for stuck or shorted address bits.

The pattern, probe/flag and record memory are tested at speed to verify the access timing.

The user can specify either quick or full. Quick tests only the address boundaries (0, 2, 4, 8, 16...) during the march pattern test while Full tests every address.

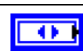
Additional failure data is available from **Query Ram Test Results**.

RAM memory size is listed below:

Memory	Size
Pattern Record Probe/Flag	262144
Sequence	32768
Timing Set	8192

Memory	Size
Persistence Waveform Error Address	2048
Record Index	1024
Timing Set Index	4096

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Memory		This control specifies which data sequencer memory to test.	0 = Pattern CH1 to CH8 1 = Pattern CH9 to CH16 2 = Pattern CH17 to CH24 3 = Pattern CH25 to CH32 4 = Record 5 = Probe/Flag 6 = Sequence 7 = Timing Set 8 = Persistence 9 = Waveform 10 = Record Index 11 = Error Address 12 = Timing Set Index
Start Address	I32	This control specifies the start address of the RAM to be tested.	0 to (Size – 1) See description above.
Size	I32	This control specifies the size of the RAM to be tested.	1 to (Size – Start Address) See description above.
Type		This control specifies either a quick or a full pattern test.	0 = Quick 1 = Full
Result	I16	This control returns the RAM test result.	0 = No Errors 1 = Data Bit Failure 2 = Stuck at 1 Failure 3 = Stuck at 0 Failure 4 = Pattern 55 Failure 5 = Pattern AA Failure

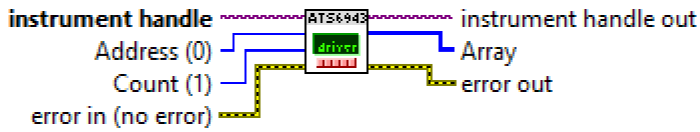
C Function Prototype Form:

ViStatus ats6943e_ramTest (ViSession instrumentHandle, ViInt32 startAddress, ViInt32 size, ViInt16 memory, ViInt16 type, ViInt16 *result);

Read 16 Bit Block

LabVIEW Diagram:

ats6943e Read 16 Bit Block.vi



Description:

This vi returns the contents of a block of 16 bit BAR0 register addresses.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Address	I32	This control specifies the address to read.	0 to hex 1001000
Count	I32	This control specifies the number of locations to read.	1 to (Size – Start Address) See description above.
Array	[I16]	This control contains the data read at the address.	0 to 65535

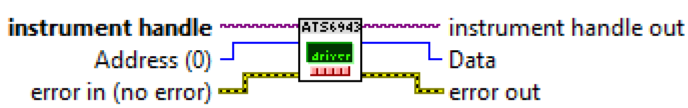
C Function Prototype Form:

ViStatus ats6943e_read_block16_data (ViSession instrumentHandle, ViInt32 address, ViInt32 count, ViInt16 array[]);

Read 16 Bit Register

LabVIEW Diagram:

ats6943e Read 16 Bit Register.vi



Description:

This vi returns the contents of a single 16 bit BAR0 register address.

Key Parameters:

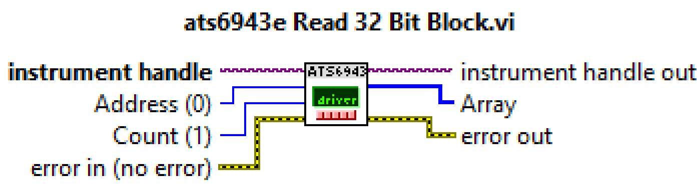
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Address	I32	This control specifies the address to read.	0 to hex 1001000
Data	I16	This control contains the data read at the address.	0 to 65535

C Function Prototype Form:

ViStatus ats6943e_read16_data (ViSession instrumentHandle, ViInt32 address, ViInt16 *data);

Read 32 Bit Block

LabVIEW Diagram:



Description:

This vi returns the contents of a block of 32 bit BAR0 register address.

Key Parameters:

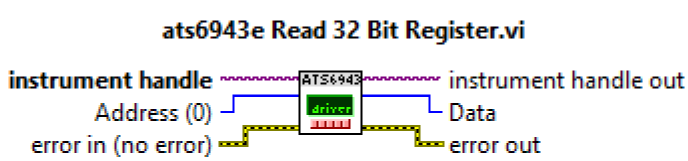
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Address	I32	This control specifies the address to read.	0 to hex 1001000
Count	I32	This control specifies the number of locations to read.	1 to (Size – Start Address) See description above.
Array	[I32]	This control contains the data read at the address.	0 to 4294967295

C Function Prototype Form:

ViStatus ats6943e_read_block32_data (ViSession instrumentHandle, ViInt32 address, ViInt32 count, ViInt32 array[]);

Read 32 Bit Register

LabVIEW Diagram:



Description:

This vi returns the contents of a single 32 bit BAR0 register address.

Key Parameters:

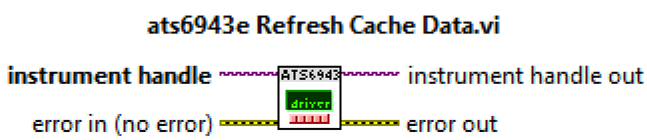
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Address	I32	This control specifies the address to read.	0 to hex 1001000
Data	I32	This control contains the data read at the address.	0 to 4294967295

C Function Prototype Form:

ViStatus ats6943e_read32_data (ViSession instrumentHandle, ViInt32 address, ViInt32 *data);

Refresh Cache Data

LabVIEW Diagram:



Description:

This vi reads the current PXIe 6943 hardware into the cache memory.

Key Parameters:

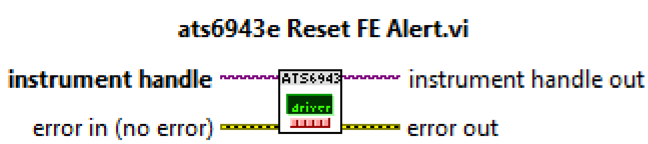
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_refreshCacheData (ViSession instrumentHandle);

Reset FE Alert

LabVIEW Diagram:



Description:

This vi resets the front-end alert register.

Resetting the alert register will reconnect any channel whose connect setting was closed but will not turn on the VEE and HV_VCC regulators.

Key Parameters:

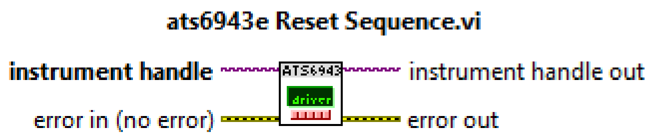
Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_resetFrontEndAlert (ViSession instrumentHandle);

Reset Sequence

LabVIEW Diagram:



Description:

This vi performs a sequence reset.

DTS Operation:

Primary if "Sequence Reset" assigned to a common PXI trigger.

All coupled modules if "Sequence Reset" not assigned to a common PXI Bus trigger.

Key Parameters:

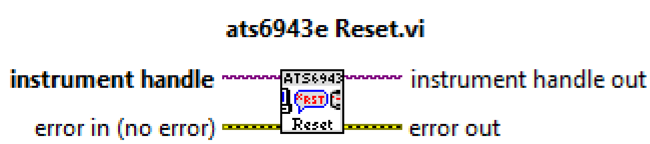
Name	Type	Description	Value
Instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_resetSequence (ViSession instrumentHandle);

Reset

LabVIEW Diagram:



Description:

This vi places the PXIe 6943 into a power-up reset state.

The reset function will perform the following actions:

1. Stop execution of both sequencer A and sequencer B.
2. Delete all defined structures (segments and sequences).
3. Open all relays.
4. Reset all references to defaults.

The reset function takes less than a second to complete.

DTS Operation:

All coupled modules.

Key Parameters:

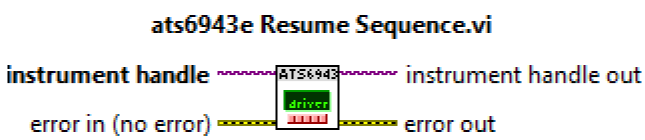
Name	Type	Description	Value
Instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_reset (ViSession instrumentHandle);

Resume Sequence

LabVIEW Diagram:



Description:

This vi resumes an execution sequence from a break or single step operation.

DTS Operation:

Primary only.

Key Parameters:

Name	Type	Description	Value
Instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

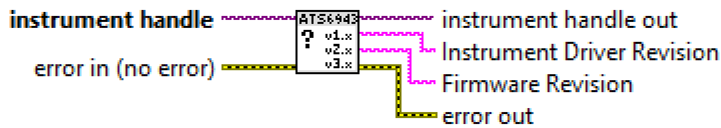
C Function Prototype Form:

ViStatus ats6943e_resumeSequence (ViSession instrumentHandle);

Revision Query

LabVIEW Diagram:

ats6943e Revision Query.vi



Description:

This vi returns the instrument driver and firmware revision of the instrument being used.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Instrument Driver Revision	abc	This control returns the driver revision.	ASCII string returned will be = 6 characters.
Firmware Revision	abc	This control returns the instrument firmware revision.	ASCII string returned will be = 12 characters.

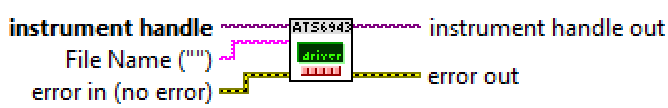
C Function Prototype Form:

ViStatus ats6943e_revision_query (ViSession instrumentHandle, ViChar instrumentDriverRevision[], ViChar firmwareRevision[]);

Save Configuration

LabVIEW Diagram:

ats6943e Save Configuration.vi



Description:

This vi saves the current PXIe 6943 configuration to a file set specified by the file name.

The file set consists of three parts:

1. <file name>.cfg contains all the PXIe 6943 device settings including the timing, sequencer and front-end data.
2. <file name>_A_PMEM.dat contains the pattern memory for sequencer.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
File Name	abc	This control specifies the file to save to.	Must be <= 364

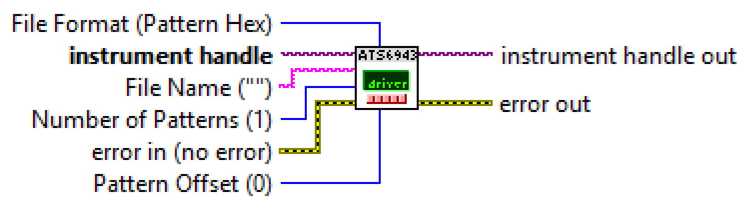
C Function Prototype Form:

ViStatus ats6943e_saveConfiguration (ViSession instrumentHandle, ViChar fileName[]);

Save Pattern Memory

LabVIEW Diagram:

ats6943e Save Pattern Memory.vi



Description:

This vi reads the PXIe 6943 pattern memory and writes it to the file name in the specified format.

The formats include:

- Pattern data as ASCII Hex
- Pattern data as ASCII String
- Pattern data as Binary
- Pattern data and flags as ASCII Hex
- Pattern data and flags as ASCII String
- Pattern data and flags as Binary

A header is written to the file that identifies the number of patterns and the format.

The format of the header is:

[ATS6943 PAT DUMP <dd> <nnnnnn>]

where:

<dd> is the format;

00 = Pattern data as ASCII Hex

01 = Pattern data as ASCII String

02 = Pattern data as Binary

03 = Pattern data and flags as ASCII Hex

04 = Pattern data and flags as ASCII String

05 = Pattern data and flags as Binary

<nnnnnn> is the number of patterns.

Key Parameters:

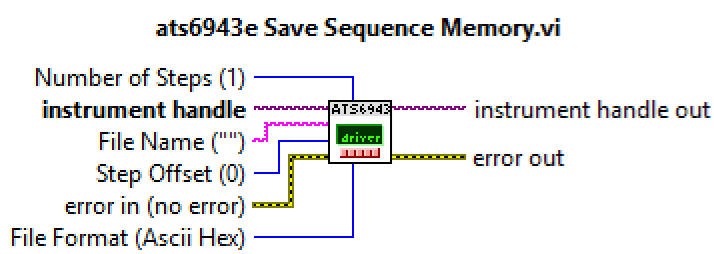
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
File Name	abc	This control specifies the file to save to.	Must be <= 364
Number of Patterns	I32	This control specifies the number of patterns to read.	1 to 262144
File Format	I16	This control specifies the save format.	0 = Hex Patterns 1 = Binary Patterns 2 = String Patterns 3 = Hex Patterns and Flags 4 = Binary Patterns and Flags 5 = String Patterns and Flags
Pattern Offset	I32	This control specifies the pattern offset where the first pattern will be read.	0 to 262243

C Function Prototype Form:

ViStatus ats6943e_savePatternMemory (ViSession instrumentHandle, ViChar fileName[], ViInt32 numberOfPatterns, ViInt16 fileFormat, ViInt32 patternOffset);

Save Sequence Memory

LabVIEW Diagram:



Description:

This vi reads the PXIe 6943 sequence memory and writes it to the file name in the specified format.

The formats include:

- ASCII Hex
- Binary

A header is written to the file that identifies the number of steps and the format.

The format of the header is:

[ATS6943E SEQ DUMP <dd> <nnnn>]

where:

<dd> is the format;

0 = ASCII Hex.

1 = Binary

<nnnn> is the number of sequence steps.

Key Parameters:

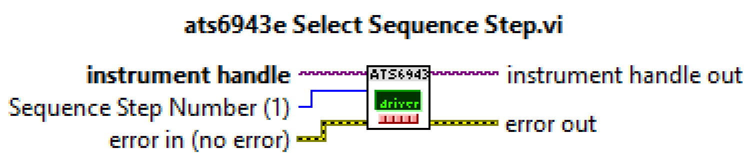
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
File Name	abc	This control specifies the file to save to.	Must be <= 364
Number of Steps	I16	This control specifies the number of steps to read.	1 to 4096
File Format	I16	This control specifies the save format.	0 = Hex 1 = Binary
Step Offset	I16	This control specifies the step offset where the first step will be read.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_saveSequenceMemory (ViSession instrumentHandle, ViChar fileName[], ViInt16 stepOffset, ViInt16 numberOfSteps, ViInt16 fileFormat);

Select Sequence Step

LabVIEW Diagram:



Description:

This vi selects a sequencer step number for subsequent programming or query functions. The step number remain active until this vi is called with a different selection.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

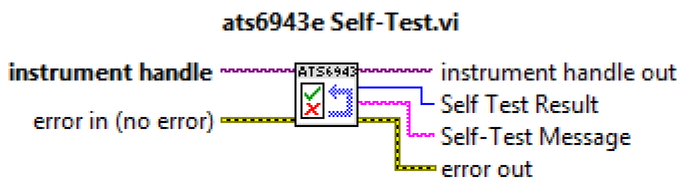
Sequence Step Number		This control specifies the sequence step number to select. All the functions in this class program or query the selected step number.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_selectSequenceStep (ViSession instrumentHandle, ViInt16 sequenceStepNumber);

Self-Test

LabVIEW Diagram:



Description:

This vi causes the PXIe 6943 to perform its self-test function and return the result.

<results> Bit Number	Message
0	Counter measuring 500MHz / 2 timeout
1	500MHz / 2 out of range
2	Counter measuring FS at 250MHz timeout
3	FS at 250MHz out of range
4	Counter measuring PXICLK10 timeout
5	PXICLK10 out of range
6	Counter measuring PG at 25MHz timeout
7	PG at 25MHz out of range
8	DB POST error
9	DR POST error
10	Sequence RAM
11	Timing Set RAM
12	Persistence RAM
13	Waveform RAM
14	Record Index RAM
15	Error Address RAM
16	Pattern 1-8 RAM
17	Pattern 9-16 RAM
18	Pattern 17-24 RAM

<results> Bit Number	Message
19	Pattern 25-32 RAM
20	Record RAM
21	Probe/Flag RAM

Key Parameters:

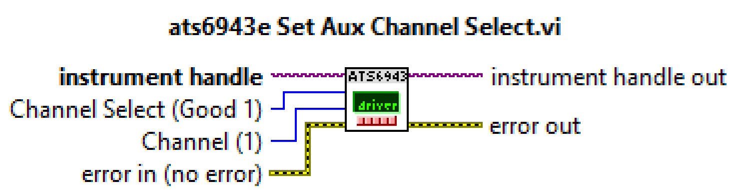
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Self-Test Result	I16	This control returns the result of the self-test command.	0 = No Error 1 = Error, see message.
Self-Test Message	abc	This control returns a message "Self-Test Passed" if the self-test command returns a zero and "Self-Test Failed;<results>" if a non-zero is returned.	See description above.

C Function Prototype Form:

ViStatus ats6943e_self_test (ViSession instrumentHandle, ViInt16 *selfTestResult, ViChar selfTestMessage[]);

Set Aux Channel Select

LabVIEW Diagram:



Description:

This vi programs the channel of the specified channel select signal.

When the AUX source is set to "Channel Select Good 1" or "Channel Select Good 0" any of the 32 channel inputs can be set as the source.

Key Parameters:

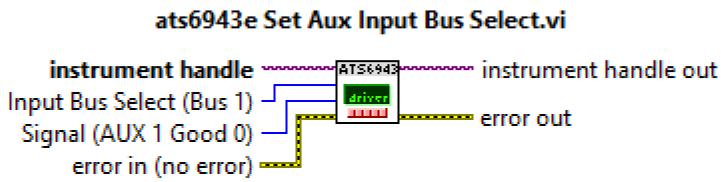
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel Select	◂▸	This control specifies which channel select to set.	0 = Good 1 1 = Good 0
Channel	I16	This control sets the channel select signal.	1 to 32

C Function Prototype Form:

ViStatus ats6943e_setAuxChannelSelect (ViSession instrumentHandle, ViInt16 channelSelect, ViInt16 channel);

Set Aux Input Bus Select

LabVIEW Diagram:

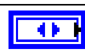
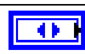


Description:

This vi programs the signal source of the specified bus select signal.

When the AUX source is set to "Input Bus 1-4", the bus select signal will be output on the specified AUX pin.

Key Parameters:

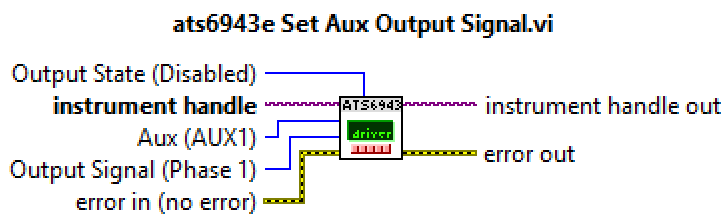
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Input Bus Select		This control specifies which input bus to program.	0 = Bus 0 1 = Bus 1 2 = Bus 2 3 = Bus 3
Signal		This control specifies the signal assigned to the input bus.	0 = AUX0 Good 0 1 = AUX1 Good 1 2 = AUX2 Good 1 3 = AUX3 Good 1 4 = AUX4 Good 1 5 = AUX5 Good 1 6 = AUX6 Good 1 7 = AUX7 Good 1 8 = AUX8 Good 1 9 = AUX9 Good 1 10 = AUX10 Good 1 11 = AUX11 Good 1 12 = AUX12 Good 1 13 = Channel Test 1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7

C Function Prototype Form:

ViStatus ats6943e_setAuxInputBusSelect (ViSession instrumentHandle, ViInt16 inputBusSelect, ViInt16 signal);

Set Aux Output Signal

LabVIEW Diagram:



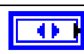
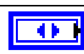
Description:

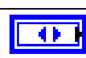
This vi programs the output signal and output state of the specified auxiliary signal.

Each data sequencer has twelve auxiliary signals that can be programmed to output a variety of signals.

If the auxiliary signal is assigned as an input then set the output state to disabled.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Aux		This control specifies which AUX signal to program.	0 = AUX1 . . 11 = AUX12
Output Signal		This control specifies the signal assigned to the AUX signal.	0 = Phase 1 1 = Phase 2 2 = Phase 3 3 = Phase 4 4 = Window 1 5 = Window 2 6 = Window 3 7 = Window 4 8 = Waveform 1 9 = Waveform 2 10 = Waveform 3 11 = Waveform 4 12 = Sync 1 13 = Sync 2 14 = Idle Active 15 = Sequence Active 16 = Channel Select Good 1 17 = Channel Select Good 0 18 = Waveform 5 19 = Waveform 6 20 = Input Bus Select 1

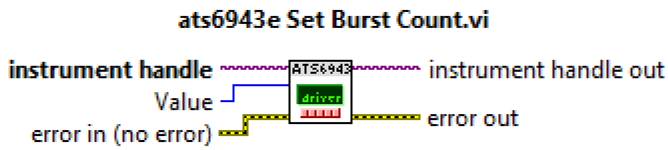
Name	Type	Description	Value
			21 = Input Bus Select 2 22 = Input Bus Select 3 23 = Input Bus Select 4 24 = Sequence Flag 1 25 = Sequence Flag 2 26 = T0CLK 27 = Pattern Clock 28 = SEQ_CLK In 29 = Jump In 30 = Raw Error 31 = SEQCLKD In 32 = T0CLK Out 33 = SEQ_CLK Out 34 = Jump Out 35 = SEQCLKD Out 36 = Pulse Generator 37 = Record Active 38 = FS Reference Clock 39 = FS 40 = Jump Strobe 41 = Int Error 42 = Ext Error 43 = HIGH 44 = PASS 45 = FAIL 46 = CONDEN 47 = BERREN 48 = Load Sequence Register 49 = Load Loop Count 50 = Counter Active 51 = CPP Done 52 = Last Word 53 = Burst Count Done 54 = Loop Count Done 55 = Gosub Active 56 = Counted Loop 57 = Subroutine Return 58 = Return Flag 59 = Last Sequence 60 = Jump Test 1 61 = Jump Test 2 62 = Jump Test 3 63 = Jump Test 4
Output State		This control specifies the output state of the aux signal.	0 = Disabled 1 = Enabled 2 = Inverted

C Function Prototype Form:

ViStatus ats6943e_setAuxOutputSignal (ViSession instrumentHandle, ViInt16 aux, ViInt16 outputSignal, ViInt16 outputState);

Set Burst Count

LabVIEW Diagram:



Description:

This vi programs the sequence burst count.

DTS Operation:

All coupled modules.

Key Parameters:

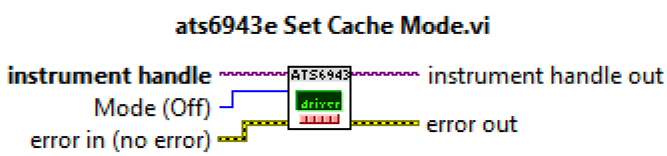
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Value	I32	This control specifies the burst count.	0 to 1048576 0 = continuous

C Function Prototype Form:

ViStatus ats6943e_setBurstCount (ViSession instrumentHandle, ViInt32 value);

Set Cache Mode

LabVIEW Diagram:



Description:

This vi enables/disables the cache mode for the current session.

The cache mode significantly improves the overall programming time of the API. The PXIe 6943 pattern, sequence and timing memories are stored in a local structure. The PXIe 6943 memories are only updated prior to sequence execution or by the **Update Cache Data** vi.

Cached data is local to the specified session and is not shared with other instances. It is up to the user to manage the cache data when enabled.

Key Parameters:

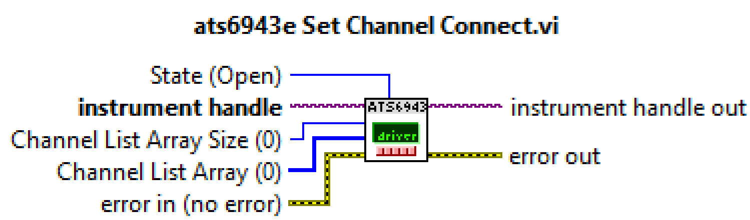
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Mode		This control specifies the cache mode count.	0 = Disabled 1 = Enabled

C Function Prototype Form:

ViStatus ats6943e_setCacheMode (ViSession instrumentHandle, ViInt16 mode);

Set Channel Connect

LabVIEW Diagram:



Description:

This vi programs the front panel channel connect state.

There are two possible states:

- Open: Driver/Receiver logic isolated from the I/O pin.
- Close: Driver/Receiver logic connected to the I/O pin.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
State		This control specifies the channel list connect state.	0 = Open 1 = Close

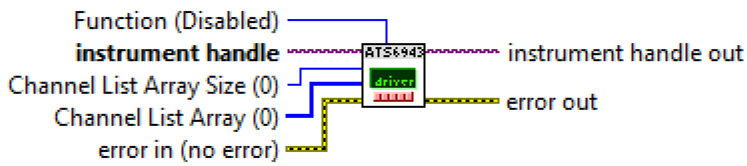
C Function Prototype Form:

ViStatus ats6943e_setChannelConnect (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 state);

Set Channel Function

LabVIEW Diagram:

ats6943e Set Channel Function.vi



Description:

This vi programs the front panel channel function setting.

The channel function can be set to:

- Disabled (power on default)
- Dynamic HiZ
- Dynamic VTT
- PMU Force Voltage
- PMU Force Current

The Disable setting places the channel in a low power mode and is not functional for dynamic or PMU functions.

The Dynamic settings are pattern driven from the digital sequencer. The driver disabled mode can be set to HiZ (two level output) or VTT (three level output).

The PMU settings are not controlled by the sequencer and are set by the **Set Force Current** and **Set Force Load** vi's.

Key Parameters:

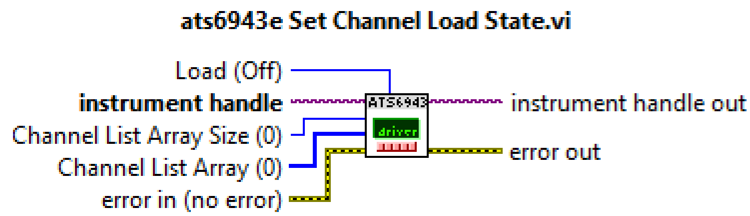
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Function	◂▸	This control specifies the channel function setting.	0 = Disable 1 = Dynamic HiZ 2 = Dynamic VTT 3 = PMU Force Voltage 4 = PMU Force Current

C Function Prototype Form:

ViStatus ats6943e_setChannelFunction (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 function);

Set Channel Load State

LabVIEW Diagram:



Description:

This vi programs the front panel channel load state.

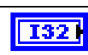
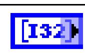
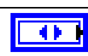
The active load can only be used if the channel function is set to "DYNAMIC_HIZ"

The active load states are:

- "Off" - Active load is disabled.
- "On" - Active load is enable.
- "HiZ" - Active load is enabled when driver is disabled.

The load type and parameters are set in the **Set Channel Sense Parameters** vi.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size		The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array		Array contains the channel numbers to program.	1 to 32
Load		This control specifies the channel active load state.	0 = Off 1 = On 2 = HiZ

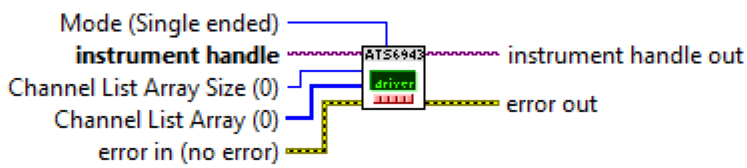
C Function Prototype Form:

ViStatus ats6943e_setChannelLoadState (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 load);

Set Channel Mode

LabVIEW Diagram:

ats6943e Set Channel Mode.vi



Description:

This vi programs the channel mode, which sets the comparator path that determines the good 1 and good 0 levels.

The comparator path can be set to:

- Single ended
- Differential

The differential path selects the comparator that uses adjacent odd and even channels. The single ended path uses dual comparators with the CVL and CVH thresholds.

For single ended:

if CH > CVH then good 1 is set high.

if CH < CVH then good 1 is set low.

if CH > CVL then good 0 is set low.

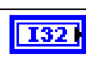
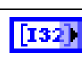
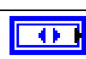
if CH < CVL then good 0 is set high.

For differential:

if CHn > CHn+1 then good 1 is set high and good 0 is set low.

if CHn < CHn+1 then good 1 is set low and good 0 is set high.

Key Parameters:

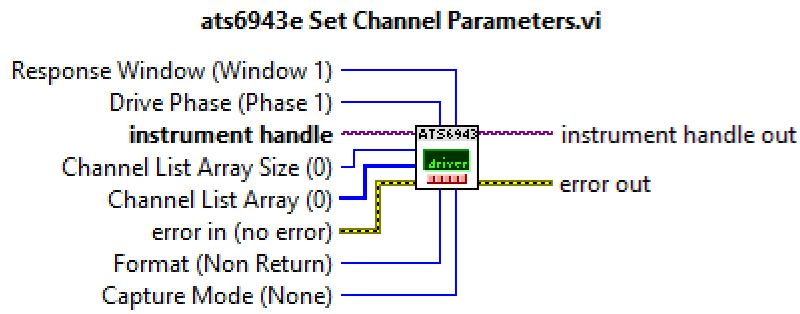
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size		The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array		Array contains the channel numbers to program.	1 to 32
Mode		This control specifies the channel mode setting.	0 = Single Ended 1 = Differential

C Function Prototype Form:

ViStatus ats6943e_setChannelMode (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 mode);

Set Channel Parameters

LabVIEW Diagram:



Description:

This vi programs the channel parameters for the specified channels.

The channel parameters consist of the following:

1. Drive Phase.
2. Response Window.
3. Format.
4. Capture Mode.

Drive Phase/Response Window:

Based on the selected timing mode there is either one phase /window or four phases/windows that can be selected:

- Per Step Multi 4
- Per Step Single 1
- Indexed 4

Each phase has an Assert time and a Return time. If the channel has a valid drive level programmed in the pattern memory, then it will be driven at the "Assert" time. The drive level will perform the specified format at the "Return" time.

Each window has an Open time and a Close time. If the channel has a valid expect level programmed in the pattern memory, then it will be tested based on the response mode using the "Open" and or "Close" times. See ‘Capture Mode’ below.

Format Settings:

Non Return Output driver remains in previous state at the start of the pattern. Phase Assert causes output driver to go to the level specified by the Pattern Code instruction in Pattern Memory. Phase Return has no action.

Return Off	Output driver remains in previous state at the start of the pattern. Phase Assert causes output Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return disables the output driver (HiZ or VTT based on channel function setting).
Return Zero	Output driver remains in previous state at the start of the pattern. Phase Assert causes output Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return causes output driver to go to the DVL level.
Return One	Output driver remains in previous state at the start of the pattern. Phase Assert causes output Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return causes output driver to go to the DVH level.
Return Comp	Output driver remains in previous state at the start of the pattern. Phase Assert causes output Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return causes output driver to go to the complemented level determined by the Pattern Code instruction in Pattern Memory.
Comp Surround	Output driver goes to complemented level determined by the Pattern Code instruction in Pattern Memory at the start of the pattern. At the Assert, Output driver goes to level determined by the Pattern Code instruction in Pattern Memory. Phase Return causes output driver to go to the complemented level determined by the Pattern Code instruction in Pattern Memory. Note: For this format to work effectively, the assert must be at least 15 ns (depends on the swing and slew-rate programmed).
Force Low	Output driver goes to DVL level immediately.
Force High	Output driver goes to high level immediately.
Force Off	Output driver disables immediately (HiZ or VTT).
Force /Phase	Phase Assert causes output driver to go to the DVL level and Phase Return causes the output driver to go to the DVH level.
Force Phase	Phase Assert causes output driver to go to the DVH level and Phase Return causes the output driver to go to the DVL level.

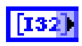
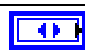
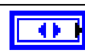
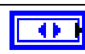
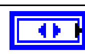
Capture Mode:

There are four capture modes available:

None	Errors, Pass Valid, Capture Faults, CRC and Drive Faults will not generated for these channels even if the pattern memory has valid expect codes.
Open Edge	Sample the comparator(s) at the open time to determine valid expect.
Close Edge	Sample the comparator(s) at the close time to determine valid expect.

Window — Sample the comparator(s) from the open time to the close time to determine valid expect. Input level must be stable the entire open to close time for a valid expect

Key Parameters:

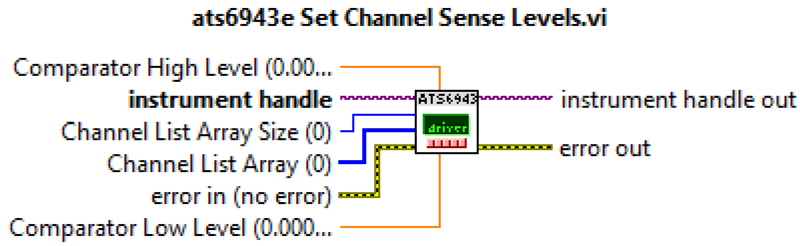
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size		The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array		Array contains the channel numbers to program.	1 to 32
Drive Phase		This control specifies the drive phase timing set signal.	0 = Phase 1 1 = Phase 2 2 = Phase 3 3 = Phase 4
Response Window		This control is used to select the response window signal for the specified channels.	0 = Window 1 1 = Window 2 2 = Window 3 3 = Window 4
Format		This control is used to select the driver format for the specified channels.	0 = Non Return 2 = Return Off 4 = Return Zero 5 = Return One 6 = Return Complement 7 = Compliment Surround 8 = Force Zero 9 = Force One 10 = Force Off 12 = Force Inverted Phase 13 = Force Phase
Capture Mode		This control is used to select the capture mode for the specified channels.	0 = None 1 = Open Edge 2 = Close Edge 4 = Window

C Function Prototype Form:

ViStatus ats6943e_setChannelParameters (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 drivePhase, ViInt16 format, ViInt16 responseWindow, ViInt16 captureMode);

Set Channel Sense Levels

LabVIEW Diagram:



Description:

This vi programs the front panel channel sense levels.

The front panel channel sense levels consist of:

1. Comparator High Level (CVH)
2. Comparator Low Level (CVL)

The comparator max levels are limited by the IOmax setting see **Set IO Max** vi.

Key Parameters:

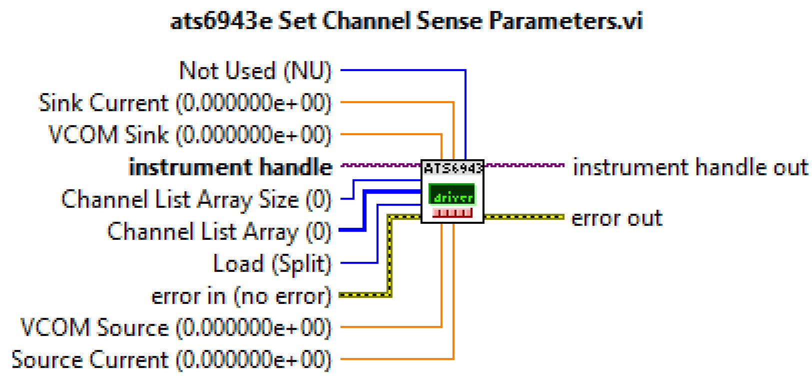
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Comparator High Level	DBL	This control specifies the high comparator level (CVH).	-2.0 to +7.0
Comparator Low Level	DBL	This control specifies the low comparator level (CVL).	-2.0 to +7.0

C Function Prototype Form:

ViStatus ats6943e_setChannelSenseLevels (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 comparatorHighLevel_CVH, ViReal64 comparatorLowLevel_CVL);

Set Channel Sense Parameters

LabVIEW Diagram:



Description:

This vi programs the front panel channel sense parameters.

The front panel channel sense parameters consist of:

1. Load Type
   - Split
   - Single
2. VCOM High Voltage
3. VCOM Low Voltage
4. Sink Current
5. Source Current

When the "Load" is set to "Split", then the sink current will be active if the input voltage is higher than VCOM High. The source current will be active if the input voltage is lower than VCOM Low.

When the "Load" is set to "Single", then the sink current will be active if the input voltage is higher than VCOM High. The source current will be active if the input voltage is lower than VCOM High. VCOM Low is not used.

The channel load state is set in the **Set Channel Load State** vi.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Load		This control specifies the channel sense load configuration.	0 = Split 1 = Single
VCOM Sink	DBL	This control specifies the VCOM Sink Voltage.	-2.0 to +7.0

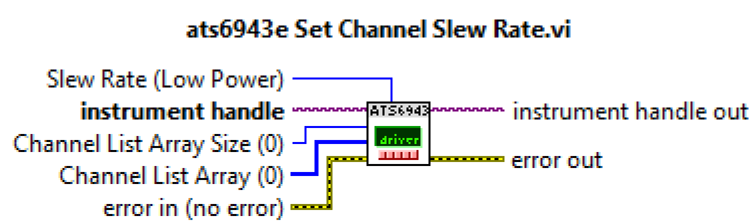
Name	Type	Description	Value
		Note: 1. If the "Load" is set to single, then this parameter is the single VCOM. 2. If the "Load" is set to split, then this parameter defines the voltage at which the sink load activates (Commutating voltage). An input voltage higher than this value enables the sink load. An input voltage less that this value disables the sink load. 3. VCOM Sink must be >= VCOM Source.	
Sink Current	DBL	This control specifies the sink current in mA.	0 to 24
VCOM Source	DBL	This control specifies the VCOM Sink Voltage. **Note**: 1. If the "Load" is set to single, then this parameter is ignored. 2. If the "Load" is set to split, then this parameter defines the voltage at which the source load activates (Commutating voltage). An input voltage higher than this value disables the source load. An input voltage less that this value enables the sink load.	-2.0 to +7.0
Source Current	DBL	This control specifies the source current in mA.	0 to 24
Not Used	I16	This control is included for legacy support and is ignored.	

C Function Prototype Form:

ViStatus ats6943e_setChannelSenseParameters (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 load, ViReal64 VCOMSink, ViReal64 VCOMSource, ViReal64 sinkCurrent_mA, ViReal64 sourceCurrent_mA, ViInt16 notUsed);

Set Channel Slew Rate

LabVIEW Diagram:



Description:

This vi programs the output channel slew rate setting.

The DR3 front-end has programmable slew rates. All other front-end types have a fixed slew rate and a setting other than "Default" or "Disabled" will return the error ATS6943E_ERROR_FRONTEND, "Installed front-end board does not support this function."

Five preset slew rates are defined:

- Fast (~1.3V/ns)
- Medium (~1.0V/ns)
- Default (~0.7V/ns)
- Slow (~0.2V/ns)
- Low Power (< 0.1 V/ns)

A sixth setting, User, can be used if a custom slew rate is programmed using the **Set Channel Source Parameters** function. This setting has no effect on the slew registers.

Key Parameters:

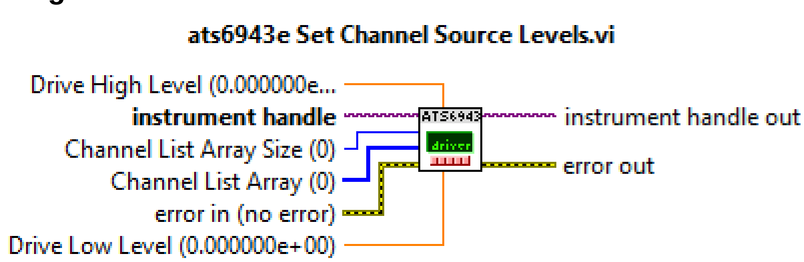
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Slew Rate		This control specifies the channel(s) slew rate setting.	0 = Fast 1 = Medium 2 = Default 3 = Slow 4 = User 5 = Low Power

C Function Prototype Form:

ViStatus ats6943e_setChannelSlewRate (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 slewRate);

Set Channel Source Levels

LabVIEW Diagram:



Description:

This vi programs the front panel channel source levels.

The front panel channel source levels consist of:

1. Drive High Level (DVH)
2. Drive Low Level (DVL)

The comparator max levels are limited by the IOmax setting see **Set IO Max** vi.

The DVL to DVH swing must be greater or equal to 0.5V

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Comparator High Level	DBL	This control specifies the high drive level (CVH).	-1.5 to IOmax
Comparator Low Level	DBL	This control specifies the low drive level (CVL).	-2.0 to IOmax – 1.0

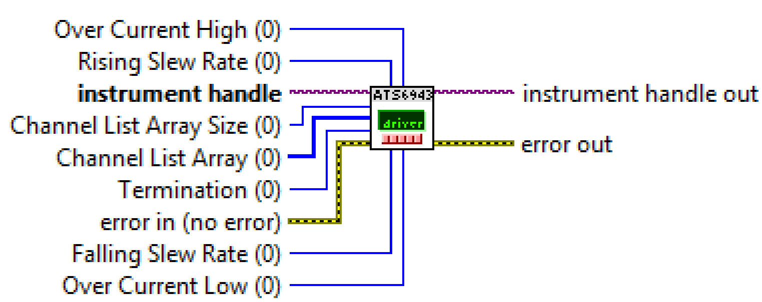
C Function Prototype Form:

ViStatus ats6943e_setChannelSourceLevels (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 driveHighLevel_DVH, ViReal64 driveLowLevel_DVL);

Set Channel Source Parameters

LabVIEW Diagram:

ats6943e Set Channel Source Parameters.vi



Description:

This vi programs the front panel channel source parameters.

The front panel channel source parameters consist of:

1. Termination value

2. Rising Slew Rate
3. Falling Slew Rate
4. Over Current High Flag (Source)
5. Over Current Low Flag (Sink)

For the rising and falling slew rates, the first three bits are a fine adjust setting and the next five bits are a course adjust. The fine adjust goes from -40% to +30% of coarse adjust in 10% increments.

0 = -40% of coarse adjust

7 = +30% of coarse adjust

The coarse adjust settings are:

0 = ~0.25V/ns

31 = ~1.5V/ns

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Rising Slew Rate	I16	This control specifies the rising edge slew rate of the specified channels.	0 to 255 0 = Do not program
Falling Slew Rate	I16	This control specifies the falling edge slew rate of the specified channels.	0 to 255 0 = Do not program
Termination	I16	This control specifies the channel list termination.	0 = 50ohms 1 = 35ohms 35 to 66
Over Current High	I16	This control is included for compatibility and is not used.	
Over Current Low	I16	This control is included for compatibility and is not used.	

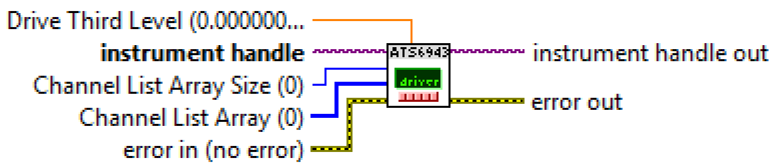
C Function Prototype Form:

ViStatus ats6943e_setChannelSourceParameters (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 termination, ViInt16 risingSlewRate, ViInt16 fallingSlewRate, ViInt16 overCurrentHigh, ViInt16 overCurrentLow);

Set Channel Source VTT

LabVIEW Diagram:

ats6943e Set Channel Source VTT.vi



Description:

This vi programs the front panel channel source third level.

The front panel channel source third level is VTT

The comparator max levels are limited by the IOmax setting see **Set IO Max** vi.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Drive Third Level	DBL	This control specifies the third drive level (VTT) on the front-end board.	-2.0 to IOmax

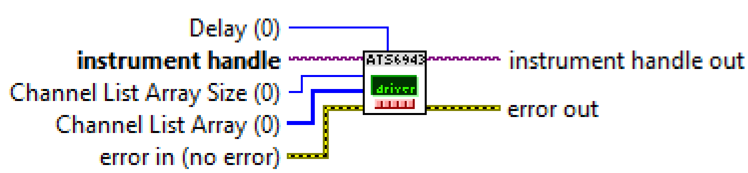
C Function Prototype Form:

ViStatus ats6943e_setChannelSourceVtt (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 driveThirdLevel_VTT);

Set Comparator Delay

LabVIEW Diagram:

ats6943e Set Comparator Delay.vi



Description:

This vi programs the front panel channel comparator delay.

The comparator delay can be set up to ~10ns with 10ps resolution.

Key Parameters:

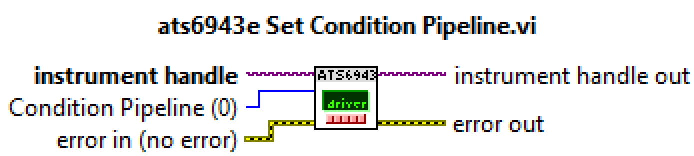
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Delay	I16	This control specifies the comparator delay (10ps per count).	0 to 1023

C Function Prototype Form:

ViStatus ats6943e_setComparatorDelay (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 delay);

Set Condition Pipeline

LabVIEW Diagram:



Description:

This vi programs the pass/fail pipeline for jumping and halting.

The sequence jump and halt logic can operate on a pass/fail condition. The pass/fail condition is the combination of all the valid channel comparator results that have an expect code programmed and "condition enable flag" (CONDEN) set true for the current pattern (CONDEN only required if the pass/fail basis is set to qualified)

The pass/fail condition is inserted into a pipeline with a programmable depth up to 16.

For qualified pass/fail basis, the pattern test enable must also be set to either 'Condition' or 'Both' to generate a valid Pass/Fail signal **Set Pattern Test Enable** vi

Key Parameters:

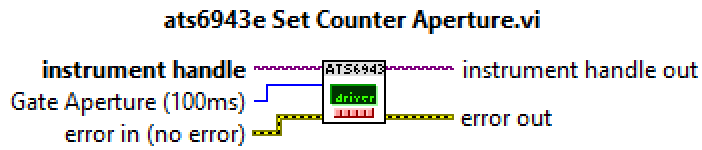
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Condition Pipeline	I16	This control specifies pipeline setting.	0 to 16

C Function Prototype Form:

ViStatus ats6943e_setConditionPipeline (ViSession instrumentHandle, ViUInt32 conditionPipeline);

Set Counter Aperture

LabVIEW Diagram:



Description:

This vi programs the gate aperture time.

The aperture is used in the following counter functions:

- Frequency
- Period
- Timed Totalize

Key Parameters:

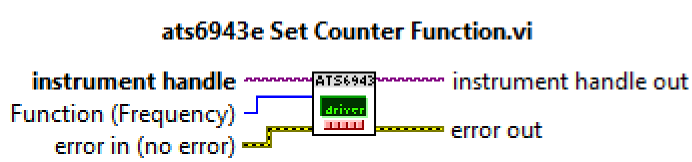
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Gate Aperture		This control specifies the channel(s) slew rate setting.	0 = 1µs 1 = 10µs 2 = 100µs 3 = 1ms 4 = 10ms 5 = 100ms 6 = 1s 7 = 10s

C Function Prototype Form:

ViStatus ats6943e_setCounterAperture (ViSession instrumentHandle, ViInt16 gateAperture);

Set Counter Function

LabVIEW Diagram:



Description:

This vi programs the counter function.

There are seven functions:

1. Frequency (CINPUT1)
2. Period (CINPUT1)
3. Time Interval (CINPUT1 to CINPUT2)
4. Totalize (CINPUT1 by CINPUT3)
5. Timed Totalize (CINPUT1)
6. Positive Pulse (CINPUT1)
7. Negative Pulse (CINPUT1)

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Function		This control specifies the counter function.	0 = Frequency 1 = Period 2 = Time Interval 3 = Totalize 4 = Timed Totalize 5 = Positive Pulse 6 = Negative Pulse

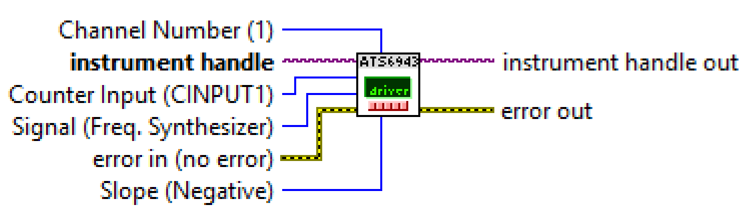
C Function Prototype Form:

ViStatus ats6943e_setCounterFunction (ViSession instrumentHandle, ViInt16 function);

Set Counter Input

LabVIEW Diagram:

ats6943e Set Counter Input.vi



Description:

This vi programs the counter/timer input signal source and slope.

The counter has two inputs available and are described below:

CINPUT1 Used as the input for frequency and timed totalize measurements and as the start signal for time interval measurements (period, positive pulse, negative pulse).

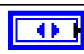
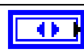
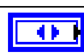
CINPUT2	Used as the stop signal for time interval measurements.
CINPUT3	Used to: • Start the aperture windows in frequency, period and timed totalize measurements on the rising edge. • Enable totalize measurement and as the stop signal for time interval measurements.

The input signal source can be set to the following:

- Any I/O Channel
- Any AUX Channel
- Sequencer Frequency Synthesizer
- CLK10
- 250MHz
- Pulse Generator Output

The input signal can be set to either the positive or negative slope as the active edge.

Key Parameters:

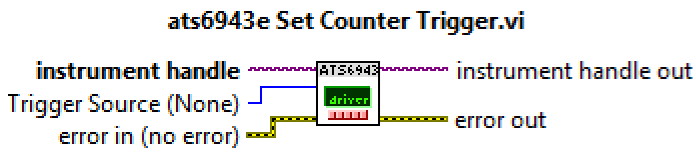
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Counter Input		This control specifies the counter input signal to program.	0 = CINPUT1 1 = CINPUT2 2 = CINPUT3
Signal		This control specifies the source of the selected counter input signal.	0 = I/O Channel 1 = AUX Channel 2 = Frequency Synthesizer 3 = CLK10 4 = 250MHz 5 = Pulse Generator
Channel Number	I16	This control specifies which I/O or AUX channel number when the source is set to I/O Channel (0) or AUX Channel (1).	I/O Channel: 1 to 32 AUX Channel: 1 to 12
Slope		This control specifies the counter input signal slope.	0 = Positive 1 = Negative

C Function Prototype Form:

ViStatus ats6943e_setCounterInput (ViSession instrumentHandle, ViInt16 counterInput, ViInt16 signal, ViInt16 channelNumber, ViInt16 slope);

Set Counter Trigger

LabVIEW Diagram:



Description:

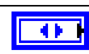
This vi programs the counter trigger source.

The counter trigger is used to start a frequency or timed totalize measurement and is set to either:

- None
- External CINPUT3
- Internal Continuous
- Internal Single

If the source is set to internal Continuous, then the counter measurement is triggered continuously.

Key Parameters:

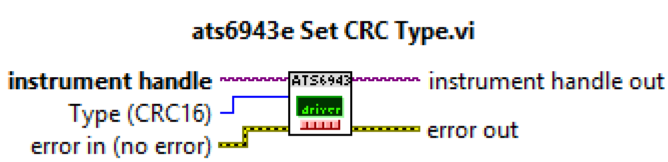
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Source		This control specifies the counter trigger selection.	0 = None 1 = External 2 = Internal Continuous 3 = Internal Single

C Function Prototype Form:

ViStatus ats6943e_setCounterTrigger (ViSession instrumentHandle, ViInt16 triggerSource);

Set CRC Type

LabVIEW Diagram:



Description:

This vi programs the CRC type.

The CRC type can be set to:

- CRC16 (default)
- CRC32
- Custom

See **Set Sequencer Attribute** to specify the custom CRC polynomial.

The CRC16 polynomial is:

G(x) = X^16 + X^12 + X^9 + X^7 + 1

The CRC32 polynomial is:

G(x) = X^32 + X^26 + X^23 + X^23 + X^22 + X^16 + X^12 + X^11 + X^10 + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Type		This control specifies the CRC type.	0 = CRC16 1 = CRC32 2 = Custom 3 = Internal Single

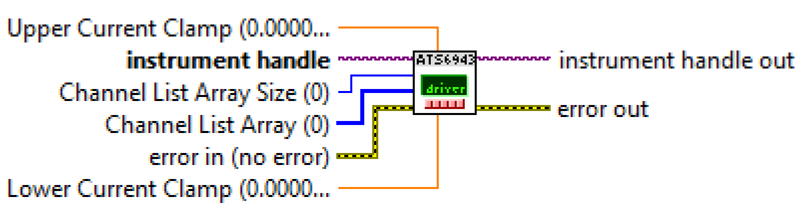
C Function Prototype Form:

ViStatus ats6943e_setCrcType (ViSession instrumentHandle, ViInt16 type);

Set Current Clamps

LabVIEW Diagram:

ats6943e Set Current Clamps.vi



Description:

This vi programs the upper and lower current clamps for the specified channels.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32

Upper Current Clamp	DBL	This control specifies the upper current clamp value (mA).	-2 * IR to 2 * IR IR = Current Range
Lower Current Clamp	DBL	This control specifies the lower current clamp value (mA).	-2 * IR to 2 * IR IR = Current Range

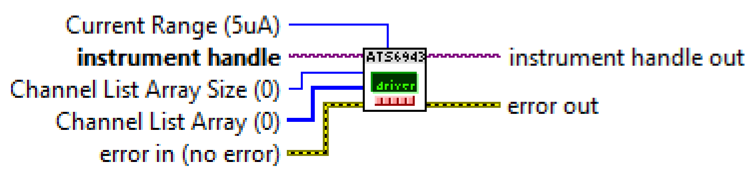
C Function Prototype Form:

ViStatus ats6943e_setPmuIClamps (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 upperCurrentClamp_mA, ViReal64 lowerCurrentClamp_mA);

Set Current Range

LabVIEW Diagram:

ats6943e Set Current Range.vi



Description:

This vi programs the current range (IR) of the specified channels for PMU force current (FI) and measure current (MI) operation.

The five current ranges are:

- +/-5uA
- +/-50uA
- +/-500uA
- +/-5mA
- +/-50mA

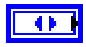
If the channel(s) are not set to PMU FV or PMU FI function, then the IR setting will be saved.

If the channel(s) are set to PMU FV or PMU FI function, then changing the IR involves the following steps:

1. Dis-connect PMU if active.
2. Program new IR.
3. Re-connect PMU

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

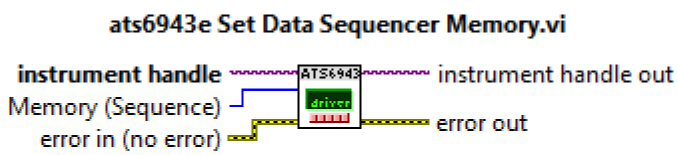
Name	Type	Description	Value
Channel List Array Size		The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array		Array contains the channel numbers to program.	1 to 32
Current Range		This control specifies the IR.	0 = 5µA 1 = 50µA 2 = 500µA 3 = 5mA 4 = 50mA

C Function Prototype Form:

ViStatus ats6943e_setPmuIR (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 currentRange);

Set Data Sequencer Memory

LabVIEW Diagram:

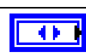


Description:

This vi sets the memory select register for BAR0 register access.

Shared memories (pattern, record and probe/flag) are requested from the sequencer. If the memory is not granted, an error will be returned.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Memory		This control specifies the memory to select.	0 = Pattern CH1 to CH8 1 = Pattern CH9 to CH16 2 = Pattern CH17 to CH24 3 = Pattern CH25 to CH32 4 = Record 5 = Probe/Flag 6 = Sequence 7 = Timing Set 8 = Persistence 9 = Waveform 10 = Record Index 11 = Error Address

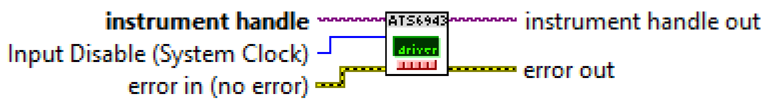
C Function Prototype Form:

ViStatus ats6943e_setDataSequencerMemory (ViSession instrumentHandle, ViInt16 memory);

Set Driver Enable Control

LabVIEW Diagram:

ats6943e Set Driver Enable Control.vi



Description:

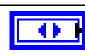
This vi programs driver enable control of the dynamic IO channels.

When a channel transitions from an output pattern code to an input pattern code, the enable can be set to disable at the beginning of the pattern (System Clock) or when the phase assert signal is true.

DTS Operation:

- All coupled modules.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Input Disable		This control specifies when the driver enable for an input channel goes false.	0 = System Clock 1 = Phase Assert

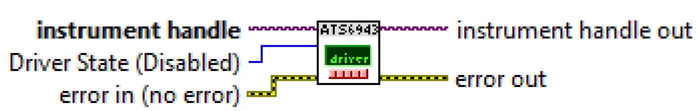
C Function Prototype Form:

ViStatus ats6943e_setDriverEnableControl (ViSession instrumentHandle, ViInt16 inputDisable);

Set Driver Enable

LabVIEW Diagram:

ats6943e Set Driver Enable.vi



Description:

This vi programs the enable state of the AUX and dynamic channel output drivers. This setting does not affect channels configured as PMU.

DTS Operation:

Disable	Primary if "Driver Disable" assigned to a common TTL trigger. All coupled modules if "Driver Disable" not assigned to a common TTL trigger.
Enable	All coupled modules.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Driver State		This control specifies the driver enable state.	0 = Disabled 1 = Enabled

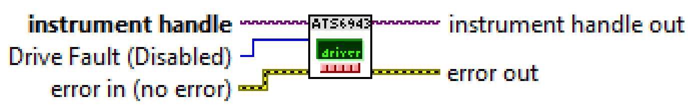
C Function Prototype Form:

ViStatus ats6943e_setDriverEnable (ViSession instrumentHandle, ViInt16 driverState);

Set Driver Fault State

LabVIEW Diagram:

ats6943e Set Driver Fault State.vi



Description:

This vi programs drive fault state of the sequencer.

If an output pin is enabled to also compare its state (Capture mode programmed and compare levels set), then a drive fault event will be generated if the compare level does not match the output state.

When enabled, a drive fault event will generate a driver disable pulse that disables the output drivers of the local sequencer. Multiple sequencers can be coupled by assigning "Driver Disable" to a common PXI bus signal.

If enabled a drive fault event will disable all channels of the specified sequencer and a drive fault event will be generated.

Use **Query Sequencer Event** to query the drive fault event and **Query Sequencer Drive Fault** to query which channel caused the drive fault.

DTS Operation:

Only affects sequencer programmed.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1

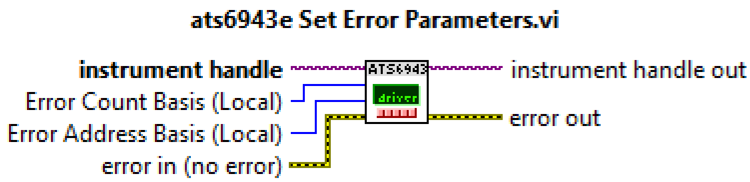
Name	Type	Description	Value
Drive Fault		This control specifies the drive fault state.	0 = Disabled 1 = Enabled

C Function Prototype Form:

ViStatus ats6943e_setDriveFaultState (ViSession instrumentHandle, ViInt16 driveFault);

Set Error Parameters

LabVIEW Diagram:



Description:

This vi programs the error parameters of the selected data sequencer.

There are two error parameters:

- Error Count Basis
- Error Address Basis

Error Count Basis:

Allows the user to select which error signal to use to determine the error count:

- Use local error.
- Use BERREN qualified local error.
- Use DTS error pulse.
- Use BERREN qualified DTS error pulse.

Error Address Basis:

Allows the user to select which error signal causes an error to be recorded in the Error Address Memory:

- Use local error.
- Use BERREN qualified local error.
- Use DTS error pulse.
- Use BERREN qualified DTS error pulse.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Error Count Basis		This control specifies the error count basis.	0 = Local Errors 1 = Qualified Local Errors 2 = DTS Errors 3 = Qualified DTS Errors
Error Address Basis		This control specifies the error address basis.	0 = Local Errors 1 = Qualified Local Errors 2 = DTS Errors 3 = Qualified DTS Errors

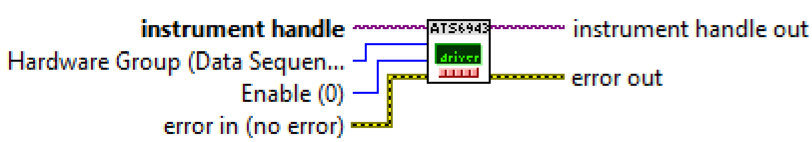
C Function Prototype Form:

ViStatus ats6943e_setErrorParameters (ViSession instrumentHandle, ViInt16 errorCountBasis, ViInt16 errorAddressBasis);

Set Event Enable

LabVIEW Diagram:

ats6943e Set Event Enable.vi



Description:

Stores the present values for all of the module data to the instrument's nonvolatile memory. This data is automatically recalled on power-up.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 232-1
Hardware Group	I16	This control specifies which hardware group setting(s) to set.	0 = Data Sequencer 1 = Driver Receiver 2 = Digital Board
Enable	I32	This control programs the enable register of the select hardware group.	Data Sequencer Enable Bit 0 = Idle sequence started 1 = Execute sequence started 2 = External halt occurred 3 = Burst error occurred

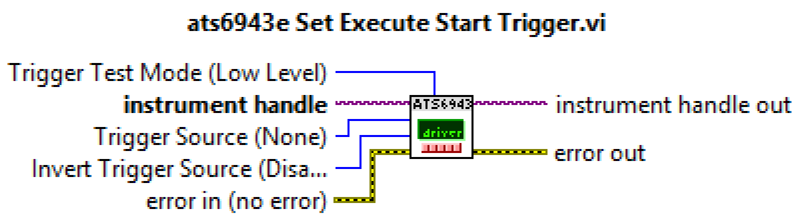
Name	Type	Description	Value
			4 = Jump occurred 5 = Over current occurred 6 = Watchdog timeout occurred 7 = Sequence timeout occurred 8 = RAM Test error occurred 9 = Inter-Module sync error occurred 10 = Phase or Window glitch occurred 11 = Capture fault occurred 12 = Pattern timeout occurred 13 = Pause occurred 14 = External stop occurred 15 = Frequency synthesizer fault occurred 16 = Subroutine fault occurred 17 = Return fault occurred 18 = In subroutine fault occurred 19 = Idle sequence complete 20 = Execute sequence complete 21 = External T0_CLK error occurred 22 = Clock generator feedback fault occurred 23 = Drive fault occurred 24 = Record address overflow occurred 25 = Record index overflow occurred Driver Receiver Enable Bit 0 = V+ Too Low < ~+9.65V 1 = V+ Too High > ~+29.00V 2 = V- Too High > ~-2.80V 3 = V- Too Low < ~-20.00V 4 = V+ to V- Delta Fault > ~34.00V 5 = GND REF Fault > ~390mV 6 = I/O chip temperature alert 7 = I2C bus error 8 = Over Voltage Digital Board Enable Bit 0 = CPU interrupt 1 = Sequencer FPGA temperature alert 2 = Power Converter alert

C Function Prototype Form:

ViStatus ats6943e_setEventEnable (ViSession instrumentHandle, ViInt16 hardwareGroup, ViInt32 enable);

Set Execute Start Trigger

LabVIEW Diagram:



Description:

This vi configures the sequence execute start trigger of the selected data sequencer.

You must call **Arm Idle Sequence** or **Arm Sequence** prior to generating the start trigger in order to prime the sequencer.

Configuring the sequence execute start trigger consists of the following:

1. Selecting the start source.
2. Program the start source inverter.
3. Select the start test mode, level (high/low) or edge (rising/falling).

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Source		This control specifies the start trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4

Name	Type	Description	Value
			21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source		This control specifies the start trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode		This control specifies the start trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

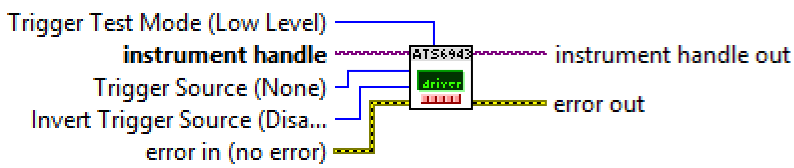
C Function Prototype Form:

ViStatus ats6943e_setExecuteStartTrigger (ViSession instrumentHandle, ViInt16 triggerSource, ViInt16 invertTriggerSource, ViInt16 triggerTestMode);

Set Execute Stop Trigger

LabVIEW Diagram:

ats6943e Set Execute Stop Trigger.vi



Description:

This vi configures the execute stop trigger of the selected data sequencer. Configuring the execute stop trigger consists of the following:

1. Selecting the stop source.
2. Program the stop source inverter.
3. Select the stop test mode, level (high/low) or edge (rising/falling).

The stop trigger causes the sequencer to stop based on the current stop mode, see **ViStatus ats6943e_setStaticTiming** (ViSession instrumentHandle, ViReal64 stimulusDelay, ViReal64 responseDelay);

Set Stop Mode vi.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Source		This control specifies the stop trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5

Name	Type	Description	Value
			6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source		This control specifies the stop trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode		This control specifies the stop trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

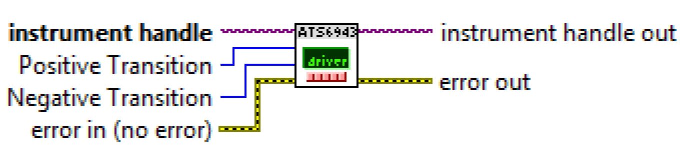
C Function Prototype Form:

ViStatus ats6943e_setExecuteStopTrigger (ViSession instrumentHandle, ViInt16 triggerSource, ViInt16 invertTriggerSource, ViInt16 triggerTestMode);

Set FE Event Transition

LabVIEW Diagram:

ats6943e Set FE Event Transition.vi



Description:

This vi sets the front-end positive and negative transition enable registers.

Bit n high of the positive transition enables the low to high transition of bit n in the condition register to set bit n in the event register high.

Bit n high of the negative transition enables the high to low transition of bit n in the condition register to set bit n in the event register high.

See **Query FE Condition** for register bit definitions.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Positive Transition	I32	This control specifies the positive transition enable register.	See description above
Negative Transition	I32	This control specifies the negative transition enable register.	See description above

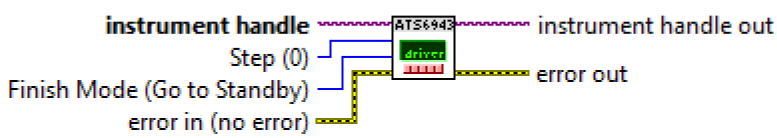
C Function Prototype Form:

ViStatus ats6943e_setFrontEndEventTransition (ViSession instrumentHandle, ViInt32 positiveTransition, ViInt32 negativeTransition);

Set Finish Sequence

LabVIEW Diagram:

ats6943e Set Finish Sequence.vi



Description:

This vi programs the finishing sequence step number and mode.

When sequence execution completes, the sequencer can enter the Standby state or the Idle state.

The Standby state will output the first pattern of the specified step and pattern memory can be reloaded by the operator.

The Idle state will output the entire pattern of the specified step and pattern memory cannot be reloaded by the operator. Idle Active signal will be true.

Key Parameters:

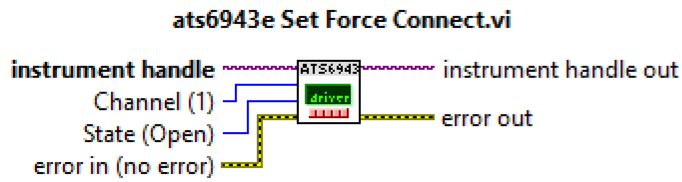
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Step	I16	This control specifies the finish sequence step number.	0 to 4095
Finish Mode	◂▸	This control specifies the mode when sequence execution finishes.	0 = Standby 1 = Idle

C Function Prototype Form:

ViStatus ats6943e_setFinishSequence (ViSession instrumentHandle, ViInt16 step, ViInt16 finishMode);

Set Force Connect

LabVIEW Diagram:



Description:

This vi selects controls the EXTFORCE connect state to the channel input path.

The EXTFORCE input is connected through a series of switches in each IO ASIC to the channel input.

If the state is set to closed, all other channels will be opened before closing the specified channel.

Key Parameters:

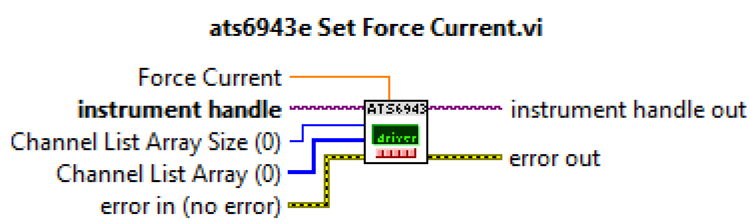
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel	I16	This control specifies which channels monitor value to set.	1 to 32
State	enum	This control specifies the connect state.	0 = Open 1 = Closed

C Function Prototype Form:

ViStatus ats6943e_setForceConnect (ViSession instrumentHandle, ViInt16 channel, ViInt16 state);

Set Force Current

LabVIEW Diagram:



Description:

This vi programs the force current level for the specified channels.

There are five current ranges:

1. IR0 5uA
2. IR1 50uA

3. IR2 500uA
4. IR3 5mA
5. IR4 50mA

Key Parameters:

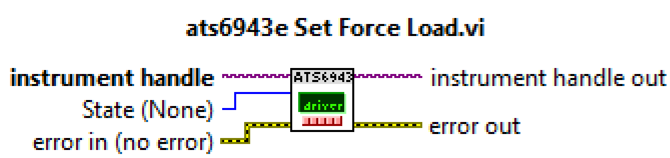
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Force Current	DBL	This control specifies the force current to set.	-2 * IR to 2 * IR IR = Current Range

C Function Prototype Form:

ViStatus ats6943e_setPmuFI (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 forceCurrent);

Set Force Load

LabVIEW Diagram:



Description:

This vi selects controls the EXTFORCE load selection.

The EXTFORCE load selections are:

- None
- 50 ohm to ground
- 10K ohm to ground

Key Parameters:

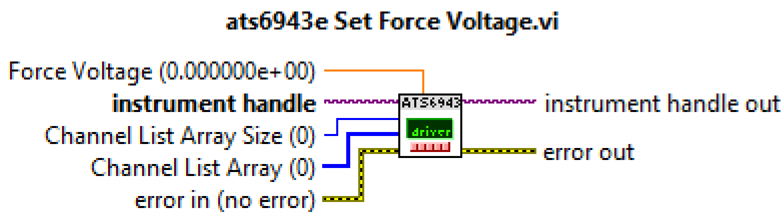
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
State		This control specifies the connect state.	0 = None 1 = 50ohm 2 = 10Kohm

C Function Prototype Form:

ViStatus ats6943e_setForceLoad (ViSession instrumentHandle, ViInt16 state);

Set Force Voltage

LabVIEW Diagram:



Description:

This vi programs the force voltage level for the specified channels.

If the channel is set to PMU FV, then the output will be programmed.

If the channel is set to PMU FI, then the voltage clamps will be set to this voltage plus and minus 100mV when the PMU is connected before setting the actual voltage clamp levels.

Key Parameters:

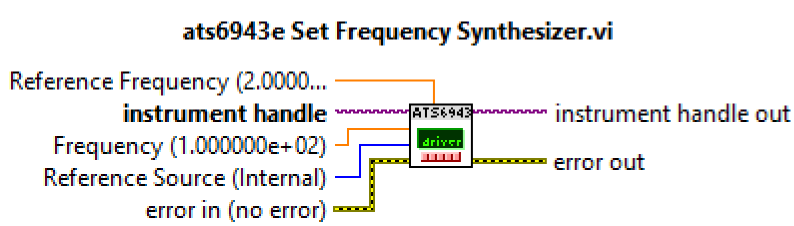
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Force Voltage	DBL	This control specifies the force current to set.	-2.0 to 7.0

C Function Prototype Form:

ViStatus ats6943e_setPmuFV (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 forceVoltage);

Set Frequency Synthesizer

LabVIEW Diagram:



Description:

This vi programs the frequency synthesizer clock.

The frequency synthesizer can be used as the "Master Clock" source, the "System Clock" source or it can simply be output through any of the front panel AUX signals.

Programming the frequency synthesizer consists of three settings:

1. Output Frequency
2. Reference Source
3. Reference Frequency

The output frequency can be from 40KHz to 500MHz, a value of zero disables the frequency synthesizer.

The reference source can be set to the 100MHz CLK100 signal, the CLK10, CLK50 (CLK100 / 2) or any of the front panel AUX signals.

The reference frequency is only required if the reference source is set to one of the AUX signal. The "Internal" reference is a 100MHz, the CLK10 is a fixed 10MHz and CLK50 is fixed at 50Mhz. This value is required in order to determine the feedback divider values for the frequency generator.

DTS Operation:

All coupled modules if FS selected as master clock by primary.

Key Parameters:

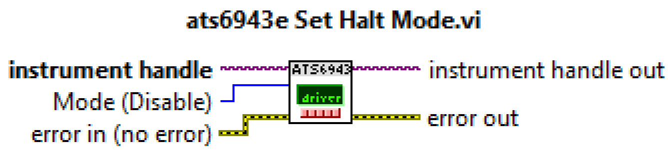
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Frequency	DBL	This control specifies the synthesizer frequency (MHz).	0 to 500 0 = Disabled
Reference Source		This control specifies the synthesizer reference clock source.	0 = Internal 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CLK10 14 = 50MHz (CLK100 / 2)
Reference Frequency	DBL	This control specifies the synthesizer reference clock frequency (MHz).	5 to 100

C Function Prototype Form:

ViStatus ats6943e_setFreqSynth (ViSession instrumentHandle, ViReal64 frequency, ViInt16 referenceSource, ViReal64 referenceFrequency);

Set Halt Mode

LabVIEW Diagram:



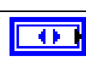
Description:

This vi programs the halt mode of the selected data sequencer.

The halt mode determines where execution will halt following either a CPU generated halt or an external trigger. 12 modes are defined:

1. The halt signal can be ignored.
2. Halt the current sequence burst at the end of the next pattern.
3. Halt the current sequence burst at the end of the next sequence step.
4. Halt the current sequence burst at the end of the sequence burst.
5. Halt the current sequence burst at the end of the next pattern where SYNC1 is set.
6. Halt the current sequence burst at the end of the next pattern where SYNC2 is set.
7. Halt the current sequence burst at the end of the next pattern if the pass/fail flag set to fail.
8. Halt the current sequence burst at the end of the next sequence step if the pass/fail flag set to fail.
9. Halt the current sequence burst at the end of the next sequence if the pass/fail flag set to fail.
10. Halt the current sequence burst at the end of the next pattern if the pass/fail flag set to pass.
11. Halt the current sequence burst at the end of the next sequence step if the pass/fail flag set to pass.
12. Halt the current sequence burst at the end of the next sequence if the pass/fail flag set to pass.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Mode		This control specifies the halt mode.	0 = Disabled 1 = Pattern 2 = Step

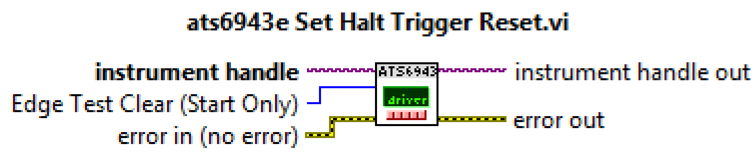
Name	Type	Description	Value
			3 = Sequence 4 = SYNC1 5 = SYNC2 6 = Pattern Fail 7 = Step Fail 8 = Sequence Fail 9 = Pattern Pass 10 = Step Pass 11 = Sequence Pass

C Function Prototype Form:

ViStatus ats6943e_setHaltMode (ViSession instrumentHandle, ViInt16 mode);

Set Halt Trigger Reset

LabVIEW Diagram:



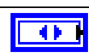
Description:

This vi programs the halt trigger edge test clear condition.

The halt trigger edge test logic consists of a pair of flip flops that monitors the rising and falling edge. The flip flops can be programmed to be cleared on the following conditions:

1. Start of Sequence (Default)
2. Start of Sequence and at the end of each sequence step.
3. Start of sequence and when the sequence is resumed.

Key Parameters:

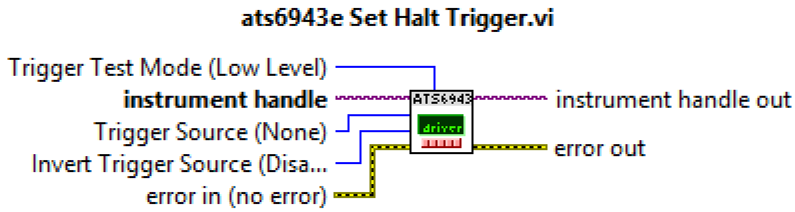
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Edge Test Clear		This control specifies the edge test clear conditions.	0 = Start Only 1 = End of Step 2 = Resume Sequence

C Function Prototype Form:

ViStatus ats6943e_setHaltTriggerReset (ViSession instrumentHandle, ViInt16 edgeTestClear);

Set Halt Trigger

LabVIEW Diagram:



Description:

This vi programs the halt trigger settings.

Programming the halt trigger consists of the following:

- Selecting the source.
- Program the source inverter.
- Select the test mode, level (high/low) or edge (rising/falling).

The halt trigger causes the sequencer to halt based on the current halt mode. Use the **Set Halt Mode** vi to program the halt mode.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Source		This control specifies the halt trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source		This control specifies the halt trigger source inverter.	0 = Disabled 1 = Enabled

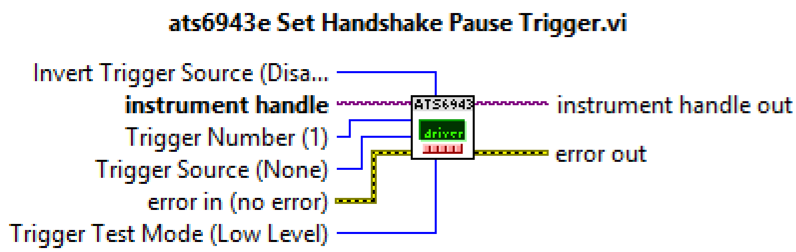
Name	Type	Description	Value
Trigger Test Mode		This control specifies the halt trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

C Function Prototype Form:

ViStatus ats6943e_setHaltTrigger (ViSession instrumentHandle, ViInt16 triggerSource, ViInt16 invertTriggerSource, ViInt16 triggerTestMode);

Set Handshake Pause Trigger

LabVIEW Diagram:



Description:

This vi configures one of the two handshake pause triggers of the selected data sequencer.

The handshake triggers are comprised of a pause signal and a resume signal. The pause signal stops the pattern timing and the resume continues the timing.

Configuring the handshake pause trigger consists of the following:

- Selecting the pause source.
- Program the pause source inverter.
- Select the pause test mode, level (high/low) or edge (rising/falling).

A pause/resume can be based on the true/false state of any of the two pause triggers. For example if handshake trigger 1 pause was set to AUX1 'Low Level' and resume was set to AUX1 'High Level', then a handshake if trigger 1 true would pause if the AUX1 is low and resume if AUX1 high.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Number	I16	This control specifies the handshake pause trigger to program.	1 to 2
Trigger Source		This control specifies the handshake pause trigger source.	0 = None 1 = AUX1

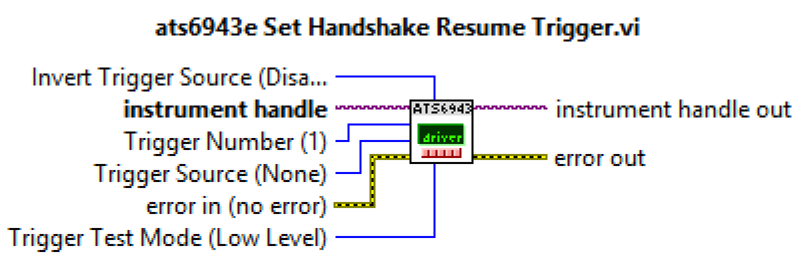
Name	Type	Description	Value
			2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source		This control specifies the handshake pause trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode		This control specifies the handshake pause trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

C Function Prototype Form:

ViStatus ats6943e_setHandshakePauseTrigger (ViSession instrumentHandle, ViInt16 triggerNumber, ViInt16 triggerSource, ViInt16 invertTriggerSource, ViInt16 triggerTestMode);

Set Handshake Resume Trigger

LabVIEW Diagram:



Description:

This vi configures one of the two handshake resume triggers of the selected data sequencer.

The handshake triggers are comprised of a pause signal and a resume signal. The pause signal stops the pattern timing and the resume continues the timing.

Configuring the handshake resume trigger consists of the following:

- Selecting the resume source.
- Program the resume source inverter.
- Select the resume test mode, level (high/low) or edge (rising/falling).

A pause/resume can be based on the true/false state of any of the two pause triggers. For example if handshake trigger 1 pause was set to AUX1 'Low Level' and resume was set to AUX1 'High Level', then a handshake if trigger 1 true would pause if the AUX1 is low and resume if AUX1 high.

Key Parameters:

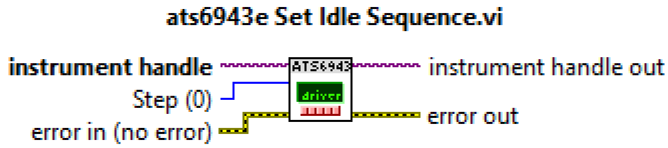
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Number	I16	This control specifies the handshake resume trigger to program.	1 to 2
Trigger Source	◂▸	This control specifies the handshake resume trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source	◂▸	This control specifies the handshake resume trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode	◂▸	This control specifies the handshake resume trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

C Function Prototype Form:

ViStatus ats6943e_setHandshakeResumeTrigger (ViSession instrumentHandle, ViInt16 triggerNumber, ViInt16 triggerSource, ViInt16 invertTriggerSource, ViInt16 triggerTestMode);

Set Idle Sequence

LabVIEW Diagram:



Description:

This vi programs the idle sequence step number.

Key Parameters:

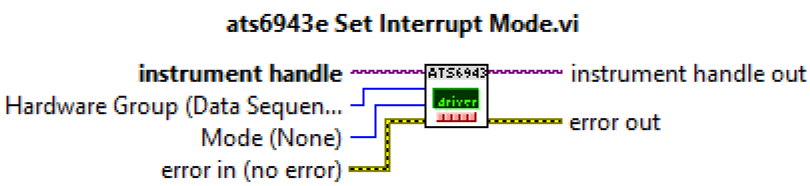
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Step		This control specifies the idle sequence step number.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_setIdleSequence (ViSession instrumentHandle, ViInt16 step);

Set Interrupt Mode

LabVIEW Diagram:



Description:

Stores the present values for all of the module data to the instrument's nonvolatile memory. This data is automatically recalled on power-up.

Key Parameters:

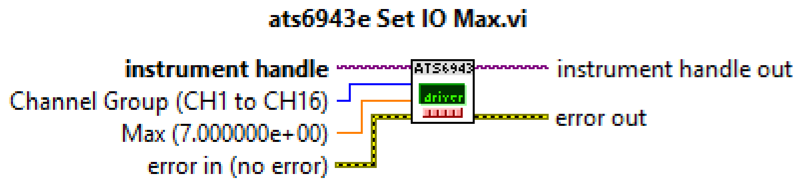
Name	Type	Description	Value
Instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_setInterruptMode (ViSession instrumentHandle, ViInt16 hardwareGroup, ViInt16 mode);

Set IO Max

LabVIEW Diagram:



Description:

This vi programs the front-end IO max level. This level is used to calculate the HV-VCC level for the DR3 front-end module. Channel settings should not exceed the max level for proper operation:

DVH: -1.5 to IOmax
DVL: -2 to IOmax - 1
VTT: -2 to IOmax
CVH,CVL: -2 to IOmax

PMU No Load
- IR0-4: -2 to IOmax

PMU Loads up to +/- Imax/2
- IR0,IR1: -2 to IOmax - 0.5
- IR2,IR3: -1.5 to IOmax - 0.5
- IR4: -1 to IOmax - 1

PMU Loads up to +/- Imax
- IR0,IR1: -1.5 to IOmax - 0.5
- IR2,IR3: -1 to IOmax - 0.5
- IR4: 0 to IOMax - 2

Active Load VCOM: 0 to IOmax - 2

Specifying a lower IO max will reduce the power and cooling required by the DR3 front-end.

There is a separate HV-VCC for channels 1-16 and 17-32.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Channel Group		This control specifies the IO max channel group.	0 = CH1 to CH16 1 = CH17 to CH32
Max	DBL	This control specifies the IO max setting for the specified group.	-1.5 to 7.0

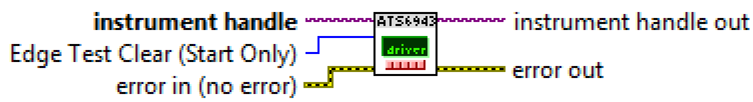
C Function Prototype Form:

ViStatus ats6943e_setIoMax (ViSession instrumentHandle, ViInt16 channelGroup, double max);

Set Jump Trigger Reset

LabVIEW Diagram:

ats6943e Set Jump Trigger Reset.vi



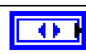
Description:

This vi programs the jump trigger edge test clear condition.

The jump trigger edge test logic consists of a pair of flip flops that monitors the rising and falling edge. The flip flops can be programmed to be cleared on the following conditions:

1. Start of Sequence (Default)
2. Start of Sequence and at the end of each sequence step.
3. Start of sequence and after a sequence jump.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Edge Test Clear		This control specifies the edge test clear conditions.	0 = Start Only 1 = End of Step 2 = Jump True

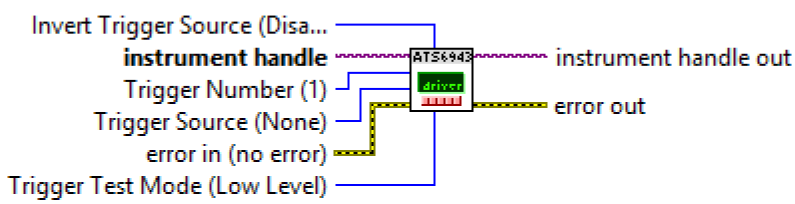
C Function Prototype Form:

ViStatus ats6943e_setJumpTriggerReset (ViSession instrumentHandle, ViInt16 edgeTestClear);

Set Jump Trigger

LabVIEW Diagram:

ats6943e Set Jump Trigger.vi



Description:

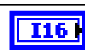
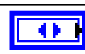
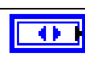
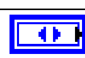
This vi programs one of the four the jump trigger settings.

Programming the jump trigger consists of the following:

- Selecting the source.
- Program the source inverter.
- Select the test mode, level (high/low) or edge (rising/falling).

The sequence jump triggers are used for conditional jumping/looping. A jump/loop can be based on the true/false state of any of the four sequence jump triggers. For example if jump trigger 1 test mode is set to 'Low Level', then a jump if trigger 1 true would occur if the selected jump trigger 1 source is low.

Key Parameters:

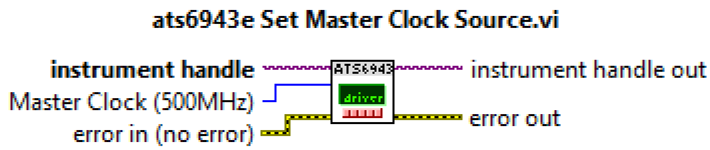
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Number		This control specifies the jump trigger to program.	1 to 4
Trigger Source		This control specifies the jump trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source		This control specifies the jump trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode		This control specifies the jump trigger test mode.	0 = Low Level 1 = High Level 2 = Rising Edge 3 = Falling Edge

C Function Prototype Form:

ViStatus ats6943e_setJumpTrigger (ViSession instrumentHandle, ViInt16 triggerNumber, ViInt16 triggerSource, ViInt16 invertTriggerSource, ViInt16 triggerTestMode);

Set Master Clock Source

LabVIEW Diagram:



Description:

This vi programs the master clock source of the sequencer.

The master clock defines the resolution for the programmable timing generator signals.

The following signals have a 1/2 master clock period resolution:

- Timing Set Phases
- Timing Set Windows
- Sequence T0CLK
- External System Clock Offset

The following signals have master clock period resolution:

- Record Offset
- Probe Offset
- Error Pulse Width

For example, if the master clock is set to internal 500Mhz, then the following resolutions exist:

- Timing Set Phases 1ns
- Timing Set Windows 1ns
- T0CLK 1ns
- System Clock Offset 1ns
- Record Offset 2ns
- Probe Offset 2ns
- Error Pulse Width 2ns

If the master clock is set to frequency synthesizer at 200MHz, then the following resolutions exist:

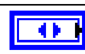
- Timing Set Phases 2.5ns
- Timing Set Windows 2.5ns
- T0CLK 2.5ns

- System Clock Offset 2.5ns
- Record Offset 5ns
- Probe Offset 5ns
- Error Pulse Width 5ns

DTS Operation:

- All coupled modules

Key Parameters:

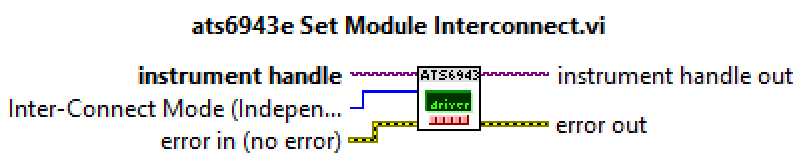
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Master Clock		This control specifies which master clock signal to select.	0 = 500MHz 15 = Frequency Synthesizer

C Function Prototype Form:

ViStatus ats6943e_setMasterClockSource (ViSession instrumentHandle, ViInt16 masterClock);

Set Module Interconnect

LabVIEW Diagram:



Description:

This vi programs the module interconnect mode.

The module interconnect mode configures the source of the sequencer timing bus signals. The source can be internal (Independent mode) or it can be external (Front Panel P1 connector).

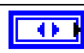
On power up the module reads the ETB code to determine the valid settings.

ETP Link	Valid Inter Module Settings
None	Independent
Primary	Primary, Independent
Secondary	Secondary, Independent
Terminator	Terminator, Independent

Independent Independent use local timing to control dynamic channels.

Primary	Provides all the timing for the DTI modules that are part of the DTS chain. The primary module must have the Primary ETB link inserted in the P1 connector.
Secondary	Secondary module(s) are located between the primary and terminating modules.
Terminating	The terminating module is used to complete the DTS timing signal chain.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Inter-Connect Mode		This control specifies the module interconnect mode setting.	0 = Independent 2 = Primary 7 = Secondary 10 = Terminating 15 = Factory Test

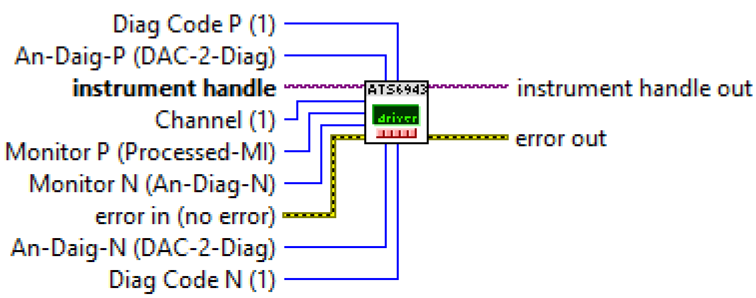
C Function Prototype Form:

ViStatus ats6943e_setModuleInterconnect (ViSession instrumentHandle, ViInt16 interConnectMode);

Set Monitor Signal

LabVIEW Diagram:

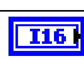
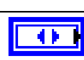
ats6943e Set Monitor Signal.vi

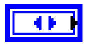
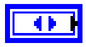
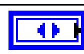


Description:

This vi selects the signal source of the Monitor signal for the specified channel.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Channel		This control specifies which channels monitor value to program.	0 to 32
Monitor P		This control specifies which signal is selected for Mon-P.	0 = An-Diag-P

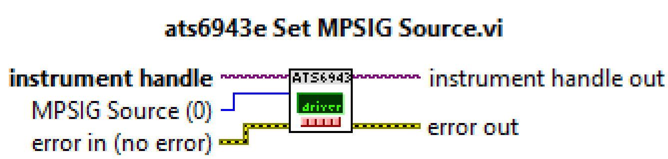
Name	Type	Description	Value
			1 = Attn-Sense 2 = Attn-MI-P 3 = Processed-MI
Monitor N		This control specifies which signal is selected for Mon-N.	0 = Buf-Ref-Gnd 1 = Attn-DG 2 = Attn-MI-N 3 = An-Diag-N
An-Diag-P		This control specifies which signal is selected for An-Diag-P and is only valid when Mon-P selection is set to An-Diag-P.	0 = Power-Diag 1 = PMU-Diag 2 = Driver-Diag 3 = DAC-2-Diag 4 = Deskew-Diag 5 = DAC-Diag 6 = Ref-Diag 7 = Central-Diag
An-Diag-N		This control specifies which signal is selected for An-Diag-N and is only valid when Mon-N selection is set to An-Diag-N.	0 = Power-Diag 1 = PMU-Diag 2 = Driver-Diag 3 = DAC-2-Diag 4 = Deskew-Diag 5 = DAC-Diag 6 = Ref-Diag 7 = Central-Diag
Diag Code P	I16	This control specifies the diagnostic code for the An-Diag-P selection.	0 to 32
Diag Code N	I16	This control specifies the diagnostic code for the An-Diag-N selection.	0 to 32

C Function Prototype Form:

ViStatus ats6943e_setMonitorSignal (ViSession instrumentHandle, ViInt16 channel, ViInt16 monitorP, ViInt16 monitorN, ViInt16 anDaigP, ViInt16 anDaigN, ViInt16 diagCodeP, ViInt16 diagCodeN);

Set MPSIG Source

LabVIEW Diagram:



Description:

This vi programs the MPSIG source.

The MPSIG can be selected as the source for the MFSIG.

Setting the bit high enables the associated signal. All the selected signals are ORed together.

Key Parameters:

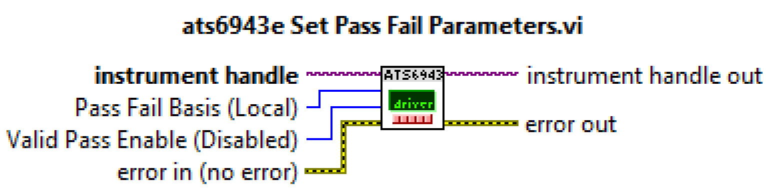
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
MPSIG Source	I16	This control programs the MPSIG source bits. Bit set high adds the signal to the OR logic of MPSIG.	Bit 0 = Sequence Active Bit 1 = Paused Bit 2 = Halted Bit 3 = Burst Error Bit 4 = NU Bit 5 = Drive Fault Bit 6 = Watchdog Timeout Bit 7 = Sequence Timeout Bit 8 = Pattern Timeout Bit 9 = Sync Error

C Function Prototype Form:

ViStatus ats6943e_setMpsigSource (ViSession instrumentHandle, ViInt16 MPSIGSource);

Set Pass Fail Parameters

LabVIEW Diagram:



Description:

This vi programs the pass/fail parameters of the sequencer.

There are two pass/fail parameters:

1. Pass/Fail Basis
2. Pass Valid Enable

Pass/Fail Basis:

Allows the user to select which error signal to use to determine the jump PASS/FAIL state.

- Use local error.
- Use CONDEN qualified error.
- Use DTS error.
- Use CONDEN qualified DTS error.

Pass Valid Enable:

Allows the user define the PASS as a VALID PASS. A VALID PASS is one where no channel errors were detected but there must be at least one valid pattern expect code.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Pass/Fail Basis	I16	This control specifies the pass/fail basis to set.	0 = Local Errors 1 = Qualified Local Errors 2 = DTS Errors 3 = Qualified DTS Errors
Valid Pass Enable	I16	This control is used to set the valid pass enable mode.	0 = Disabled 1 = Enabled

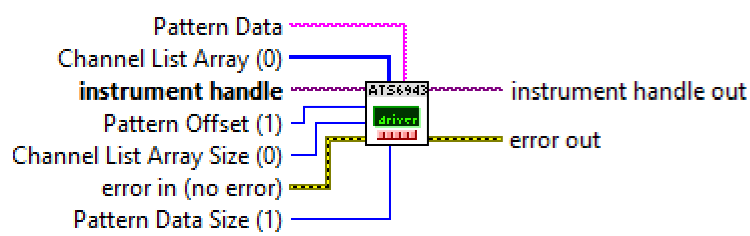
C Function Prototype Form:

ViStatus ats6943e_setPassFailParameters (ViSession instrumentHandle, ViInt16 passFailBasis, ViInt16 validPassEnable);

Set Pattern Data

LabVIEW Diagram:

ats6943e Set Pattern Data.vi



Description:

This vi program a single pattern in the pattern set of the selected sequence step. Each pattern in a pattern set contains data for 32 pins.

The pattern data is expressed as an ASCII code described below:

Pin State	Character Code	Description
Disabled	Z	Driver is either HiZ or VTT level
Collect CRC	C	Response level captured in CRC register
Drive High	1	Driver enabled and set to DVH level
Drive Low	0	Driver enable and set to DVL level
Expect Valid Low	L	Driver disabled, generate error if input > CVL
Expect Valid High	H	Driver disabled, generate error if input < CVH
Expect Valid	V	Driver disabled, generate error if input > CVL and input < DVH

Pin State	Character Code	Description
Expect Between	B	Driver disabled, generate error if input < CVL or input > DVH
Drive Low, Expect Low	l	Driver enabled and set to DVL, generator error if input > CVL
Drive High, Expect High	h	Driver enabled and set to DVH, generator error if input < CVH
Drive Low, Expect High	/	Driver enabled and set to DVL, generator error if input < CVH
Drive High, Expect Low	\	Driver enabled and set to DVH, generator error if input > CVL

An optional pin list can be specified to define which pins to program and the order with respect to the pattern data. For example, the following pin list/pattern data arrays programs pin 7 driving low, pin 1 expect low and pin 14 expect high.

Array Index	Pin List Array	Pattern Data Array
0	7	‘0’
1	1	‘L’
2	14	‘H’

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pattern Data Size	I16	This control specifies the number of elements in the pattern data array.	1 to 32
Pattern Offset	I32	This control specifies the pattern offset to program.	1 to 262144
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program	1 to 32
Pattern Data	abc	This control is used to program the pattern data.	See description above

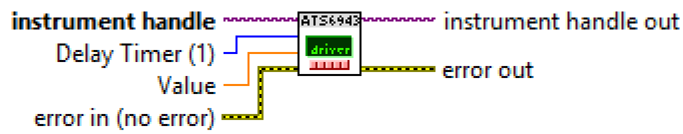
C Function Prototype Form:

ViStatus ats6943e_setPatternData (ViSession instrumentHandle, ViInt32 patternOffset, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 patternDataSize, ViChar patternData[]);

Set Pattern Delay Timer

LabVIEW Diagram:

ats6943e Set Pattern Delay Timer.vi



Description:

This vi programs the pattern delay timer value.

The timer is programmed in 10ns steps (may have an additional 10ns of error) with a range of 20ns to 42.949672970s.

The pattern delay is enabled using the "Resume Modifier" parameter in the **Set Sequence Handshake** vi.

The timer will start when the Pause starts. The Pattern Delay Timer generates a Resume when the timer times out.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Delay Timer	I16	This control specifies the pattern delay timer to program.	1 = Pattern Delay Timer 1 2 = Pattern Delay Timer 2
Value	DBL	This control specifies the pattern delay timer value.	20ns to 42.949672970s

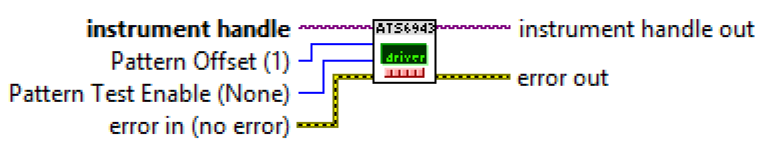
C Function Prototype Form:

ViStatus ats6943e_setPatternDelayTimer (ViSession instrumentHandle, ViInt16 delayTimer, ViReal64 value);

Set Pattern Test Enable

LabVIEW Diagram:

ats6943e Set Pattern Test Enable.vi



Description:

This function programs the pattern test enable setting at the specified offset of the selected sequence step.

The test enable setting can be set to:

None	Pin state error does not affect the conditional step error jump or burst error status.
Condition	Pin state error contributes to the pass/fail condition jump but not the burst error.
Burst Error	Pin state error contributes to the burst error but not the pass/fail condition jump.
Both	Pin state affects both the burst error and pass/fail condition.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

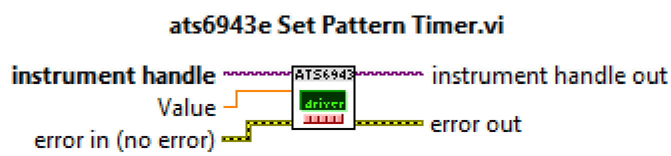
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Pattern Offset	I32	This control specifies the pattern offset to program.	1 to 262144
Pattern Test Enable		The number of elements in the "Channel List Array" parameter.	0 = None 1 = CONDEN 2 = BERREN 3 = Both

C Function Prototype Form:

ViStatus ats6943e_setPatternTestEnable (ViSession instrumentHandle, ViInt32 patternOffset, ViInt16 patternTestEnable);

Set Pattern Timer

LabVIEW Diagram:



Description:

This vi programs the pattern timeout value.

The timeout is programmed in 10ns steps (may have an additional 10ns of error) with a range of 20ns to 42.949672970s.

The pattern timeout is enabled using the "Resume Modifier" parameter in the **Set Sequence Handshake** vi.

The Pattern Timeout Timer, will generate an event when the timer times out and the occurrence of this particular event can be enabled to generate an interrupt so the S/W can query the events to see which one occurred. The pause will continue unless the termination condition is subsequently met, whereby execution will resume. If it doesn't, the user can manually resume the sequence. He may then also stop the sequence.

Key Parameters:

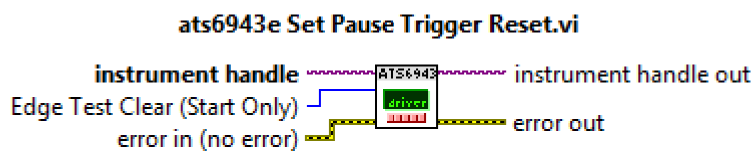
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Value	DBL	This control specifies the pattern timer value.	20ns to 42.949672970s

C Function Prototype Form:

ViStatus ats6943e_setPatternTimer (ViSession instrumentHandle, ViReal64 value);

Set Pause Trigger Reset

LabVIEW Diagram:



Description:

This vi programs the pause trigger edge test clear condition.

The pause trigger edge test logic consists of a pair of flip flops that monitors the rising and falling edge. The flip flops can be programmed to be cleared on the following conditions:

1. Start of Sequence (Default)
2. Start of Sequence and at the end of each sequence step.
3. Start of sequence and after a sequence jump.

Key Parameters:

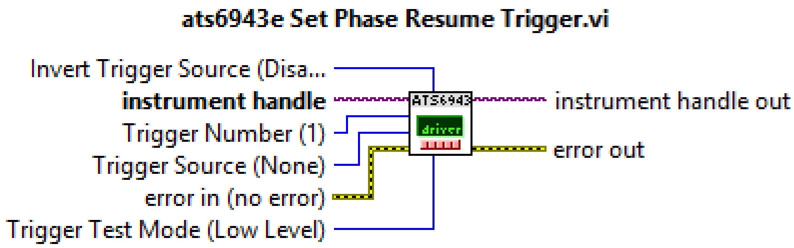
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Edge Test Clear		This control specifies the edge test clear conditions.	0 = Start Only 1 = End of Step 2 = Jump True

C Function Prototype Form:

ViStatus ats6943e_setPauseTriggerReset (ViSession instrumentHandle, ViInt16 edgeTestClear);

Set Phase Resume Trigger

LabVIEW Diagram:



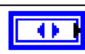
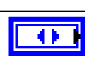
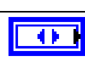
Description:

This vi configures the phase resume trigger settings.

Programming the phase resume trigger consists of the following:

- Selecting the source.
- Program the source inverter.
- Select the test mode, level (high/low) or edge (rising/falling).

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Trigger Number		This control specifies the phase resume trigger to program.	1 to 4
Trigger Source		This control specifies the phase resume trigger source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7
Invert Trigger Source		This control specifies the phase resume trigger source inverter.	0 = Disabled 1 = Enabled
Trigger Test Mode		This control specifies the phase resume trigger test mode.	0 = Low Level 1 = High Level

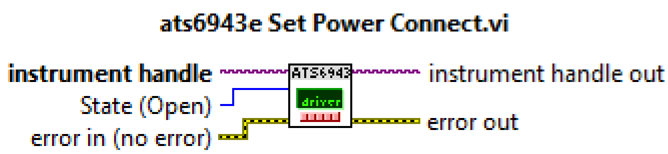
Name	Type	Description	Value
			2 = Rising Edge 3 = Falling Edge

C Function Prototype Form:

ViStatus ats6943e_setPhaseResumeTrigger (ViSession instrumentHandle, ViInt16 triggerNumber, ViInt16 triggerSource, ViInt16 invertTriggerSource, ViInt16 triggerTestMode);

Set Power Connect

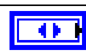
LabVIEW Diagram:



Description:

This vi programs the power connect relay state.

Key Parameters:

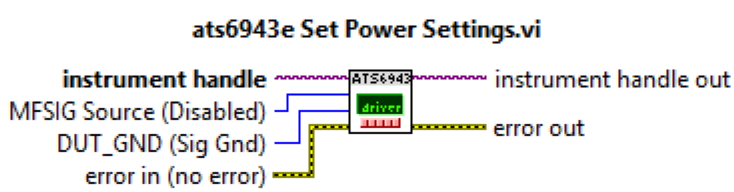
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
State		This control specifies the front-end power connect relay state.	0 = Open 1 = Close

C Function Prototype Form:

ViStatus ats6943e_setPowerConnect (ViSession instrumentHandle, ViInt16 state);

Set Power Settings

LabVIEW Diagram:



Description:

This vi programs the front panel power connector settings.

There are two power connector settings:

1. MFSIG Source: This setting programs the MFSIG source:

a. Disabled
b. MPSIG.

2. DUT_GND: This setting is used to determine the source of the DUT_GND reference signal.

    a. Signal Ground
    b. Front Panel

Key Parameters:

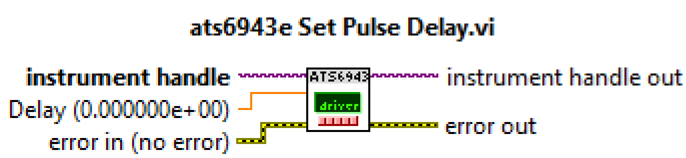
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
MFSIG Source		This control specifies the MFSIG source.	2 = Disabled 3 = MPSIG
State		This control specifies the source of the DUT_GND reference signal. When the state is set to front panel, then the DUT_GND is connected to front panel signal.	0 = Signal Ground 1 = Front Panel

C Function Prototype Form:

ViStatus ats6943e_setPowerSettings (ViSession instrumentHandle, ViInt16 MFSIGSource, ViInt16 DUT_GND);

Set Pulse Delay

LabVIEW Diagram:



Description:

This vi sets the pulse generator delay.

The period is programmed in 10ns steps with a range of 0ns to 42.949672960s.

The delay is not required for CONTINUOUS mode.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Delay	DBL	This control specifies the pulse generator delay.	0 to 42.949672960s

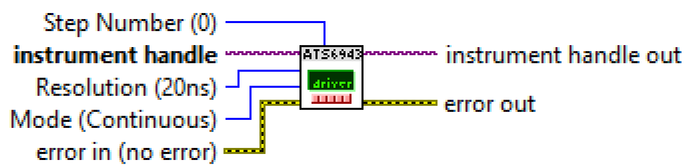
C Function Prototype Form:

ViStatus ats6943e_setPulseDelay (ViSession instrumentHandle, ViReal64 delay);

Set Pulse Parameters

LabVIEW Diagram:

ats6943e Set Pulse Parameters.vi



Description:

This vi sets the pulse generator parameters.

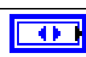
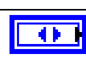
The pulse generator parameters includes the pulse resolution and the pulse mode

This module has fixed resolution of 10ns.

The pulse generator mode can be set to either CONTINUOUS, CONTINUOUS START, SINGLE or SINGLE STEP.

Continuous:	The pulse generator begins continuous output when armed.
Continuous Start:	The pulse generator begins continuous output from the start of the sequence.
Single:	The pulse generator outputs a single pulse from the start of the sequence.
Single Step:	The pulse generator outputs a single pulse from the start of the specified step.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Step Number		This control specifies the step number. This control is ignored if the mode is not set to "Single Step".	0 to 4095
Mode		This control specifies the pulse generator mode.	0 = Continuous 1 = Continuous Start 2 = Single Start 3 = Single Step
Resolution		This control is included for legacy API support and is ignored. The resolution is 10ns.	0 = 20ns (Not available) 1 = 10ns

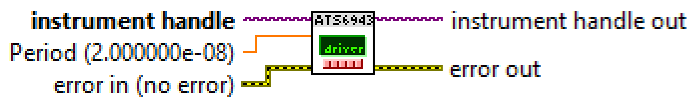
C Function Prototype Form:

ViStatus ats6943e_setPulseParameters (ViSession instrumentHandle, ViInt16 resolution, ViInt16 mode, ViInt16 stepNumber);

Set Pulse Period

LabVIEW Diagram:

ats6943e Set Pulse Period.vi



Description:

This function sets the pulse generator period.

The period is programmed in 10ns steps with a range of 20ns to 42.949672970s.

The period is not required for SINGLE START and SINGLE STEP mode.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Delay	DBL	This control specifies the pulse generator period.	0 to 42.949672970s

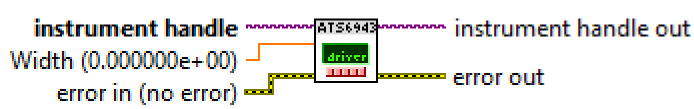
C Function Prototype Form:

ViStatus ats6943e_setPulsePeriod (ViSession instrumentHandle, ViReal64 period);

Set Pulse Width

LabVIEW Diagram:

ats6943e Set Pulse Width.vi



Description:

This vi sets the pulse generator width.

The width is programmed in 10ns steps with a range of 0 to 42.949672960s.

If the width is set to 0 then there will be no pulse.

If the width is greater than the period in CONTINUOUS and CONTINUOUS START mode, then the result will be a continuously true pulse.

If the width plus the delay is greater than the period in CONTINUOUS and

CONTINUOUS START mode, then the pulse width will be reduced proportionately.

Key Parameters:

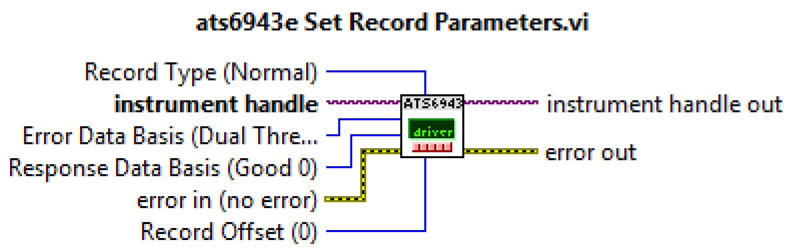
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Width	DBL	This control specifies the pulse generator width.	0 to 42.949672960s

C Function Prototype Form:

ViStatus ats6943e_setPulseWidth (ViSession instrumentHandle, ViReal64 width);

Set Record Parameters

LabVIEW Diagram:



Description:

This function programs the record parameters of the sequencer.

There are four record parameters:

- Error Data Basis
- Response Data Basis
- Record Type
- Record Offset

Error Data Basis:

Allows the user to select how the response data will be evaluated for errors when the record mode is set to "Record Errors".

1. Use both good 1 and good 0 comparator levels (Dual threshold)
2. Use only the good 1 comparator (Single threshold)

Response Data Basis:

Allows the user to select which comparator will be used to determine the data level when the record mode is set to "Record Response".

1. Use good 0 (Only available on dual threshold front-ends)
2. Use good 1

Record Type:

The record type can be normal or indexed.

Normal	Data is recorded in the record memory with the same offset as the pattern memory.
Indexed	Data is recorded sequentially starting at offset 0. The record index memory stores information that allows the recorded data to be aligned with the pattern data.

Record Offset:

The record offset allows the user to shift the record signals (pattern code expect data and window strobes) to accommodate system and UUT delay. The record offset resolution is the master clock period.

DTS Operation:

All coupled modules.

Key Parameters:

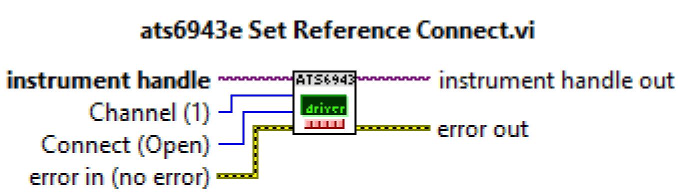
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Error Data Basis		This control specifies error data basis when the record mode is set to "Record Errors".	0 = Dual Threshold 1 = Single Threshold
Response Data Basis		This control specifies the response data basis when the record mode to set to "Record Response".	0 = Good 0 1 = Good 1
Record Type		This control is used to program the record type.	0 = Normal 1 = Indexed
Record Offset	I16	This control is included for legacy API support and is ignored. The resolution is 10ns.	0 to 63

C Function Prototype Form:

ViStatus ats6943e_setRecordParameters (ViSession instrumentHandle, ViInt16 errorDataBasis, ViInt16 responseDataBasis, ViInt16 recordType, ViInt16 recordOffset);

Set Reference Connect

LabVIEW Diagram:



Description:

This function controls the connections between E_S, E_F and the channel for calibration.

If the state is not open, all other channels E_S and E_F signals will be opened before closing the specified channel.

Key Parameters:

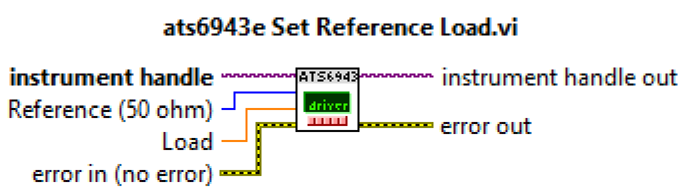
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel	I16	This control specifies which channel to program.	1 to 32
Connect		This control specifies what to connect.	0 = Disconnect Both 1 = Connect E_S to E_F 2 = Connect E_S to Channel 3 = Connect E_F to Channel 4 = Connect Both to Channel

C Function Prototype Form:

ViStatus ats6943e_setRefConnect (ViSession instrumentHandle, ViInt16 channel, ViInt16 connect);

Set Reference Load

LabVIEW Diagram:



Description:

This function specifies the actual load resistor value measured from the EXTFORCE pin to GND. This reference is used for current load Source/Sink calibration.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Reference		This control specifies the reference load to set.	0 = 50ohm 1 = 10kohm
Load	DBL	This control specifies the load resistance measured between the EXTFORCE pin and GND using an external DMM.	Should be within 10 ohms of expected reference.

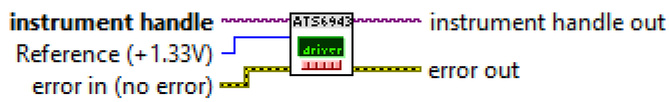
C Function Prototype Form:

ViStatus ats6943e_setRefLoad (ViSession instrumentHandle, ViInt16 load, ViReal64 load);

Set Reference Output

LabVIEW Diagram:

ats6943e Set Reference Output.vi



Description:

This function programs the EXTSOURCE mux to the specified level for ADC calibration.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Reference		This control specifies the reference voltage to output.	-1 = Off 0 = 5.0 1 = 3.66 2 = 1.33 3 = -1.0

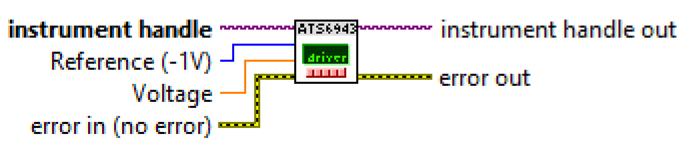
C Function Prototype Form:

ViStatus ats6943e_setRefOutput (ViSession instrumentHandle, ViInt16 reference);

Set Reference Voltage

LabVIEW Diagram:

ats6943e Set Reference Voltage.vi



Description:

This function specifies the actual reference voltage value measured at the EXTFORCE pin. These references are used for calibration.

Key Parameters:

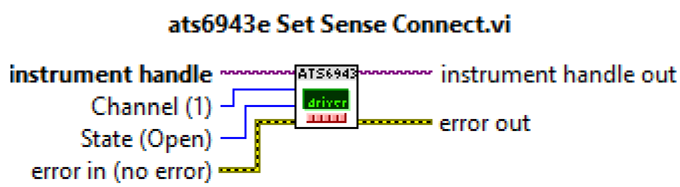
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Reference		This control specifies the reference load to set.	0 = 50ohm 1 = 10kohm
Voltage	DBL	This control specifies the reference voltage measured at the EXTFORCE pin using an external DMM.	Should be within 100mV of expected reference.

C Function Prototype Form:

ViStatus ats6943e_setRefVoltage (ViSession instrumentHandle, ViInt16 reference, ViReal64 voltage);

Set Sense Connect

LabVIEW Diagram:



Description:

This function selects controls the EXTSENSE connect state to the DIN path inside the IO ASIC.

If the state is set to closed, all other channels will be opened before closing the specified channel.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel	I16	This control specifies which channel to program.	1 to 32
Reference		This control specifies the reference voltage to output.	0 = Open 1 = Closed 2 = 1.33 3 = -1.0

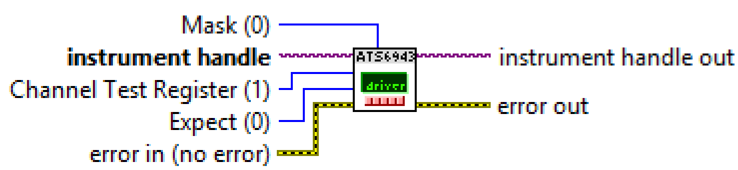
C Function Prototype Form:

ViStatus ats6943e_setSenseConnect (ViSession instrumentHandle, ViInt16 channel, ViInt16 state);

Set Sequence Channel Test

LabVIEW Diagram:

ats6943e Set Sequence Channel Test.vi



Description:

This function configures one of the four sequence channel test registers of the selected data sequencer.

Configuring the sequence channel test registers consists of the following:

1. Program the expect value.
2. Program the mask value.

The expect is compared to the response high of the input channel.

A high in the mask, disables the comparison, masked off.

The result of all four channel test registers can be routed to the TTL trigger bus. In addition channel test 1 result can also be routed to any of the sequence triggers.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel Test Register	I16	This control specifies the sequence channel test register configure.	1 to 4
Mask	I32	This control specifies the channel test mask value. A high in the mask, disables the comparison, masked off.	Bit 0 = Channel 1 mask . . Bit 31 = Channel 32 mask
Expect	I32	This control specifies the channel test expect value. The expect is compared to the response high of the input channel.	Bit 0 = Channel 1 expect . . Bit 31 = Channel 32 expect

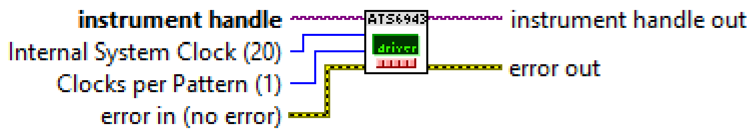
C Function Prototype Form:

ViStatus ats6943e_setSequenceChannelTest (ViSession instrumentHandle, ViInt16 channelTestRegister, ViUInt32 expect, ViUInt32 mask);

Set Sequence Clock

LabVIEW Diagram:

ats6943e Set Sequence Clock.vi



Description:

This vi sets the clock data for the selected sequence step.

The clock data consist of the following:

Clock Period — The clock period specifies the internal system clock frequency (T0CLK). The pattern period is programmed in master clock edges (rising and falling), i.e., 1/2 the master clock period.

For example if the master clock is set to 500MHz, then a setting of 20 would be;

20 * (1/2 (2ns)) = 20ns.

With a master clock of 100MHz it would be;

20 * (1/2 (10ns)) = 100ns.

Clocks per Pattern — Generates an additional pattern clock (PCLK) that can be used to reset the phase timing logic. The window timing logic is only reset with system clock.

Use **Select Sequence Step** to select the sequence step.

DTS Operation:

All coupled modules.

Key Parameters:

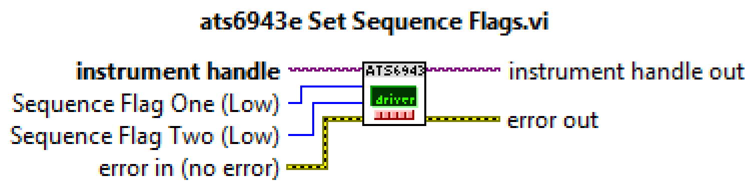
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Internal System Clock	I32	This control is used to set the internal system clock period (T0CLK) for the selected sequence step. The resolution is 1/2 the master clock period.	20 to 65550
Clocks Per Pattern	I16	This control specifies the number of system clocks per pattern clock.	1 to 256

C Function Prototype Form:

ViStatus ats6943e_setSequenceClock (ViSession instrumentHandle, ViInt32 internalSystemClock_T0CLK, ViInt16 clocksPerPattern);

Set Sequence Flags

LabVIEW Diagram:

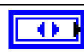
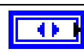


Description:

This function programs the sequence flag levels for the selected selected step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

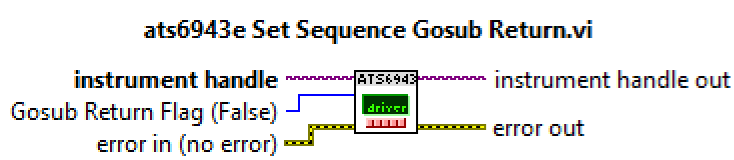
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Channel		This control specifies the level of sequence flag one when this sequence step is executed.	0 = Low 1 = High
Reference		This control specifies the level of sequence flag two when this sequence step is executed.	0 = Low 1 = High

C Function Prototype Form:

ViStatus ats6943e_setSequenceFlags (ViSession instrumentHandle, ViInt16 sequenceFlagOne, ViInt16 sequenceFlagTwo);

Set Sequence Gosub Return

LabVIEW Diagram:



Description:

This vi specifies the gosub return flag level for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

Name	Type	Description	Value
Gosub Return Flag		This control specifies the gosub return flag for this sequence step.	0 = False 1 = True

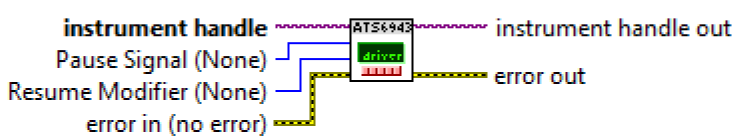
C Function Prototype Form:

ViStatus ats6943e_setSequenceGosubReturn (ViSession instrumentHandle, ViInt16 gosubReturnFlag);

Set Sequence Handshake

LabVIEW Diagram:

ats6943e Set Sequence Handshake.vi



Description:

This vi specifies the handshake signal and resume modifier for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

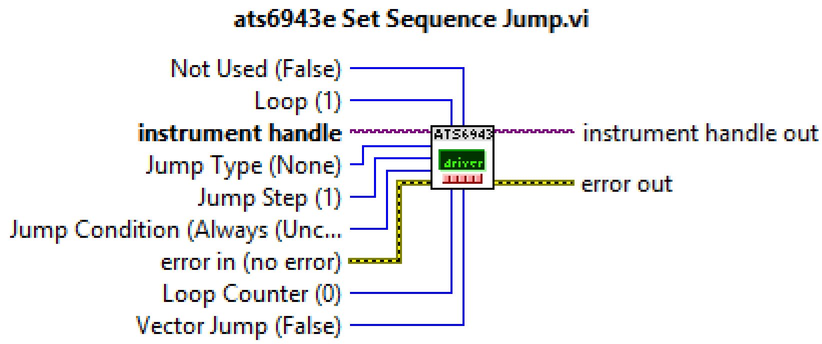
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Pause Signal		This control specifies the pause signal for the selected sequence step.	0 = None 2 = Pause Trigger 1 True 3 = Pause Trigger 1 Not True 4 = Pause Trigger 2 True 5 = Pause Trigger 2 Not True 6 = Phase 1 Assert 7 = Phase 1 Return 8 = Phase 2 Assert 9 = Phase 2 Return 10 = Phase 3 Assert 11 = Phase 3 Return 12 = Phase 4 Assert 13 = Phase 4 Return
Resume Modifier		This control specifies the resume modifier for the selected sequence step.	0 = None 1 = Pattern Delay Timer 1 2 = Pattern Delay Timer 2 3 = Pattern Timeout

C Function Prototype Form:

ViStatus ats6943e_setSequenceHandshake (ViSession instrumentHandle, ViInt16 pauseSignal, ViInt16 resumeModifier);

Set Sequence Jump

LabVIEW Diagram:



Description:

This vi specifies the jump parameters for the selected sequence step.

The jump parameters consist of the following:

Jump Type	The jump type can be set to (None, Normal or Gosub).
Jump Step	This specifies the step number to branch to.
Jump Condition	There are 15 conditions that can be selected to qualify a branch to occur:

- Always (Unconditional branch)
- Step Not PASS
- Step Not FAIL
- Step FAIL
- Step PASS
- Sequence FAIL
- Sequence PASS
- Jump Trigger 1 signal true
- Jump Trigger 1 signal not true
- Jump Trigger 2 signal true
- Jump Trigger 2 signal not true
- Jump Trigger 3 signal true
- Jump Trigger 3 signal not true
- Jump Trigger 4 signal true
- Jump Trigger 4 signal not true

Loop	The jump logic can be qualified with a loop count. The jump will only occur if the loop counter has not reached its terminal count.
Loop Counter	One of sixteen 16 bit loop counters. Each loop counter can be set to reload or disable when the terminal count is reached. (**Set Sequence Loop Mode**)
Not Used	This parameter is no longer used.

Vector Jump — The vector jump flag enables/disables the vector jump logic.

Note: The Jump Trigger level conditions are not latched and must be true during the last pattern of the pattern set for a jump to occur.

Use **Select Sequence Step** to select the sequence step.

DTS Operation:

All coupled modules.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Vector Jump		This control specifies the vector jump flag for this sequence step.	0 = False 1 = True
Loop Counter	I16	This control specifies the loop counter used if a loop count is specified.	0 to 15
Jump Type		This control specifies the jump type for the selected sequence step.	0 = None 1 = Normal 2 = Gosub
Jump Step	I16	This control specifies the sequence step number to jump to.	0 to 4095
Jump Condition		This control specifies the jump condition for the selected sequence step.	1 = Always 2 = Step Not Pass 3 = Step Not Fail 4 = Step Fail 5 = Step Pass 6 = Sequence Fail 7 = Sequence Pass 8 = Jump Trigger 1 9 = Not Jump Trigger 1 10 = Jump Trigger 2 11 = Not Jump Trigger 2 12 = Jump Trigger 3 13 = Not Jump Trigger 3 14 = Jump Trigger 4 15 = Not Jump Trigger 4
Loop Count	I32	This control specifies a loop count for the jump step.	0 to 65536 (0 disables jump)
Not Used		This control is not used and is include for legacy support.	

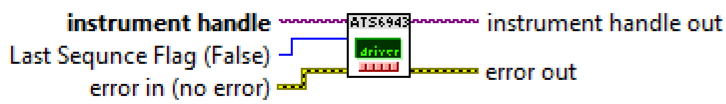
C Function Prototype Form:

ViStatus ats6943e_setSequenceJump (ViSession instrumentHandle, ViInt16 jumpType, ViInt16 jumpStep, ViInt16 jumpCondition, ViInt32 loop_0Disables, ViInt16 loopCounter, ViInt16 notUsed, ViInt16 vectorJump);

Set Sequence Last Step

LabVIEW Diagram:

ats6943e Set Sequence Last Step.vi



Description:

This vi specifies the last step flag level for the selected sequence step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Last Sequence Flag		This control specifies the last step flag for this sequence step.	0 = False 1 = True

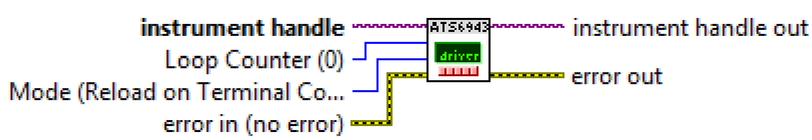
C Function Prototype Form:

ViStatus ats6943e_setSequenceLastStep (ViSession instrumentHandle, ViInt16 lastSequnceFlag);

Set Sequence Loop Mode

LabVIEW Diagram:

ats6943e Set Sequence Loop Mode.vi



Description:

This vi programs the loop counter mode.

There are sixteen 16-bit loop counters. Each of the sixteen loop counters can be programmed to either reload its count or disable when the terminal count is reached.

For example, given the following sample loop sequence:

Step 1 Output pattern set 1 jump step 1 using LC0 count 2

Step 2 Output pattern set 2 jump step 1 using LC1 count 3

If both loop counter reload on terminal count, then the step order will be:

1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 2

If counter 0 is set to disable, then the step order will be:

1, 1, 2, 1, 2, 1, 2, 1, 2

DTS Operation:

All coupled modules.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Loop Counter	I16	This control specifies the loop counter to program. 16 selects all counters.	0 to 16
Mode		This control specifies the loop counter mode.	0 = Reload on Terminal Count 1 = Disable on Terminal Count

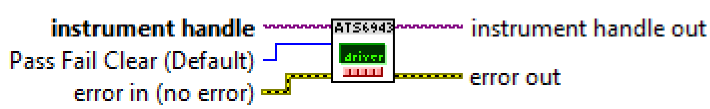
C Function Prototype Form:

ViStatus ats6943e_setSequenceLoopMode (ViSession instrumentHandle, ViInt16 loopCounter, ViInt16 mode);

Set Sequence Pass Fail Clear

LabVIEW Diagram:

ats6943e Set Sequence Pass Fail Clear.vi



Description:

This vi specifies the sequence step pass fail clear mode for the selected sequence step.

The pass fail flag is used for conditional jumping and indicates the results of a channel compare pattern code.

The pass fail flag can be set to clear at the beginning of each sequence step (default) or to hold the previous state (mask).

Use **Select Sequence Step** to select the sequence step.

DTS Operation:

Primary sequencer only.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

Name	Type	Description	Value
Pass Fail Clear		This control specifies the pass fail clear mode for the specified sequence step.	0 = Default 1 = Mask

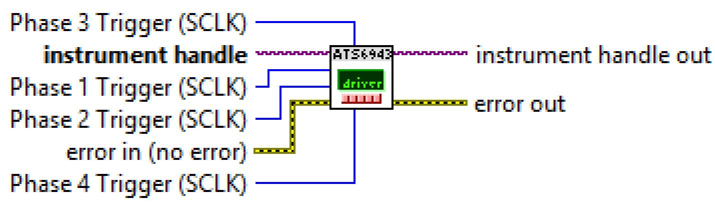
C Function Prototype Form:

ViStatus ats6943e_setSequencePassFailClear (ViSession instrumentHandle, ViInt16 passFailClear);

Set Sequence Phase Trigger

LabVIEW Diagram:

ats6943e Set Sequence Phase Trigger.vi



Description:

This vi sets the phase triggers for the selected sequence step.

The stimulus timing signal can be triggered by either the System Clock (SCLK) or the Pattern Clock (PCLK).

Note: SCLK and PCLK are identical when the clocks per pattern is set to one.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

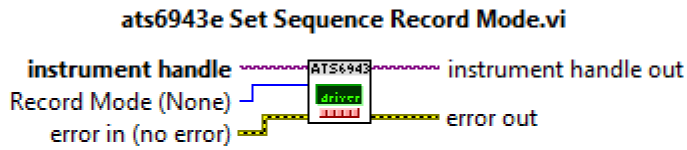
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Phase 1 Trigger		This control specifies the group 1 phase timing trigger.	0 = SCLK 1 = PCLK
Phase 2 Trigger		This control specifies the group 2 phase timing trigger.	0 = SCLK 1 = PCLK
Phase 3 Trigger		This control specifies the group 3 phase timing trigger.	0 = SCLK 1 = PCLK
Phase 4 Trigger		This control specifies the group 4 phase timing trigger.	0 = SCLK 1 = PCLK

C Function Prototype Form:

ViStatus ats6943e_setSequencePhaseTrigger (ViSession instrumentHandle, ViInt16 phase1Trigger, ViInt16 phase2Trigger, ViInt16 phase3Trigger, ViInt16 phase4Trigger);

Set Sequence Record Mode

LabVIEW Diagram:



Description:

This vi specifies the sequence step record mode of the selected sequence step.

There are three memories that stores error data from sequence burst:

1. Error Counter and Error Address Memory
2. Record Index Memory
3. Record Memory

The Error Counter indicates the number of pattern errors that occurred during the last sequence burst. The Error Count can be queried using the **Query Error Flags** vi.

The Error Address Memory stores the sequence step, address and index of each pattern that generated an error. The Error Address Memory can be queried using the **Query Error Address** vi.

The Record Index Memory contains the data required to align the record memory contents when data is stored sequentially (Record Type = Indexed).

The Record Memory contains either the error or response data for the previous sequence burst.

The sequence step record mode consist of the following four settings:

None	All three record memories are disabled (See note).
Record Count	The Error Count and Error Address Memory are enabled (See note).
Record Error	All three memories are enabled and the Record Memory is set to record error data.
Record Response	All three memories are enabled and the Record Memory is set to record response data.

Note: For the "None" and "Record Count" setting, the record memory can either be set to record all zeros (No Error) or disabled in the **Set Sequence Record Mode** vi.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

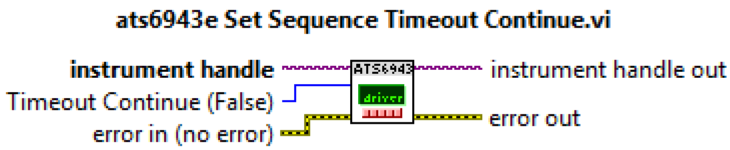
Name	Type	Description	Value
Record Mode		This control specifies the record mode for the specified sequence step.	0 = None 1 = Record Count 2 = Record Error 3 = Record Response

C Function Prototype Form:

ViStatus ats6943e_setSequenceRecordMode (ViSession instrumentHandle, ViInt16 recordMode);

Set Sequence Timeout Continue

LabVIEW Diagram:



Description:

This vi specifies the timeout continue flag level for the selected sequence step.

The sequence timeout resets at the beginning of each sequence step during a burst. By enabling this flag the sequence timeout will continue from its previous value (accumulate) during this step.

Use **Select Sequence Step** to select the sequence step.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Record Mode		This control specifies the timeout continue flag for this sequence step.	0 = False 1 = True

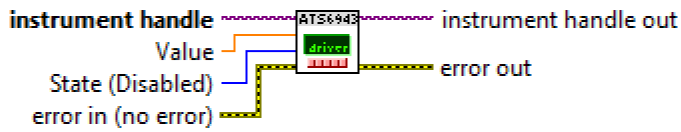
C Function Prototype Form:

ViStatus ats6943e_setSequenceTimeoutContinue (ViSession instrumentHandle, ViInt16 timeoutContinue);

Set Sequence Timer

LabVIEW Diagram:

ats6943e Set Sequence Timer.vi



Description:

This vi programs the sequence timeout value and state.

The sequence timer restarts at the beginning of every sequence step. The restart can be disabled using the STO Continue flag in the **Set Sequence Flags** vi.

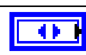
If the state is enabled, a sequence timeout will set bit 7 (STO) in the sequence event register.

The timeout is programmed in 10ns steps (may have an additional 10ns of error) with a range of 20ns to 42.949672970s.

DTS Operation:

Primary, other sequencers optional.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Value		This control specifies the sequence timeout value.	20ns to 42.949672970s
State		This control specifies the sequence timeout state.	0 = Disabled 1 = Enabled

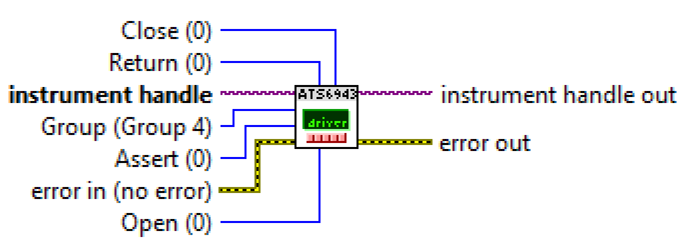
C Function Prototype Form:

ViStatus ats6943e_setSequenceTimer (ViSession instrumentHandle, ViReal64 value, ViInt16 state);

Set Sequence Timing Data

LabVIEW Diagram:

ats6943e Set Sequence Timing Data.vi



Description:

This vi programs the phase and window settings for the selected sequence step and group for non-indexed timing mode.

Each sequence step contains one or four signal groups. Each signal group contains four settings:

1. Phase Assert
2. Phase Return
3. Window Open
4. Window Close

The phase and window signals are programmed in master clock edges (rising and falling), i.e., 1/2 the master clock period. For example if the master clock is set to 500MHz, then a setting of 5 would be:

5 * (1/2 (2ns)) = 5ns.

With a master clock of 100MHz it would be:

5 * (1/2 (10ns)) = 25ns.

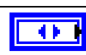
The number of groups per sequence step is determined by the timing mode setting and is programmed using the **Set Timing Mode** vi.

Use **Select Sequence Step** to select the sequence step to program.

DTS Operation:

Primary only.

Key Parameters:

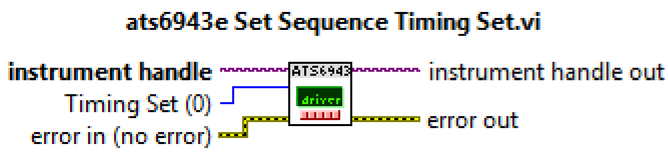
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Group		This control specifies which timing group to set.	0 = Group 1 1 = Group 2 2 = Group 3 3 = Group 4
Assert	I32	This control is used to set the phase assert time.	0 to 65535
Return	I32	This control is used to set the phase return time.	0 to 65535
Open	I32	This control is used to set the window open time.	0 to 65535
Close	I32	This control is used to set the window close time.	0 to 65535

C Function Prototype Form:

ViStatus ats6943e_setSequenceTimingData (ViSession instrumentHandle, ViInt16 group, ViInt32 assert, ViInt32 return, ViInt32 open, ViInt32 close);

Set Sequence Timing Set

LabVIEW Diagram:



Description:

This vi programs the timing set number for the selected sequence step.

In the indexed timing mode, each sequence step contains a pointer to one of 256 timing sets. Each timing set contains four groups of the following signals:

1. Phase Assert
2. Phase Return
3. Window Open
4. Window Close

This function will return "ATS6943E_ERROR_INVTM" if the timing mode is not set to indexed.

Use **Select Sequence Step** to select the sequence step to program.

Use **Set Timing Mode** to select the indexed timing mode.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Timing Set	I16	This control specifies the timing set number.	0 to 255

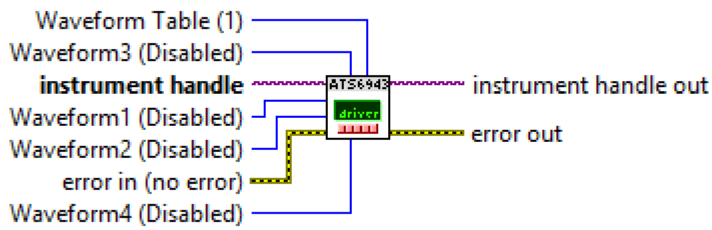
C Function Prototype Form:

ViStatus ats6943e_setSequenceTimingSet (ViSession instrumentHandle, ViInt16 timingSet);

Set Sequence Waveform

LabVIEW Diagram:

ats6943e Set Sequence Waveform.vi



Description:

This vi sets the waveform enables as well as the waveform table for the selected sequence step:

- Waveform1 replaces Phase 4.
- Waveform2 replaces Window 4.
- Waveform3 replaces Phase 3.
- Waveform4 replaces Window 3.

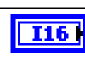
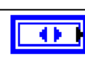
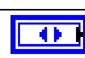
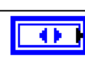
Waveforms are segmented and assigned in multiple tables. The valid segmenting is listed below:

- 16 Tables by 1024 bits per table.
- 8 Tables by 2048 bits per table.
- 4 Tables by 4096 bits per table.
- 2 Tables by 8192 bits per table.
- 1 Table by 16384 bits.

Set Waveform Table Size sets the number of waveform table segments.

Use **Select Sequence Step** to select the sequence step to program.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Waveform Table		This control specifies which timing group to set.	1 to 16
Waveform 1		This control specifies the enable state of Waveform1.	0 = Disabled 1 = Enabled
Waveform 2		This control specifies the enable state of Waveform2.	0 = Disabled 1 = Enabled
Waveform 3		This control specifies the enable state of Waveform3.	0 = Disabled 1 = Enabled

Name	Type	Description	Value
Waveform 4		This control specifies the enable state of Waveform4.	0 = Disabled 1 = Enabled

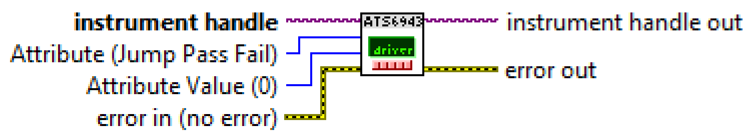
C Function Prototype Form:

ViStatus ats6943e_setSequenceWaveform (ViSession instrumentHandle, ViInt16 waveform1, ViInt16 waveform2, ViInt16 waveform3, ViInt16 waveform4, ViInt16 waveformTable);

Set Sequencer Attribute

LabVIEW Diagram:

ats6943e Set Sequencer Attrubute.vi



Description:

This vi programs the sequencer attribute values.

The following sequencer attributes can be set.

Attribute	Value
Jump Pass Fail	0 = Normal 1 = Legacy
Phase 3 Mode	0 = Normal 1 = Jump Trigger 1
Window 3 Mode	0 = Normal 1 = Jump Triger 2
Window 3 Delay	0 to 15
CRC Preload	0 to hex FFFFFFFF
CRC Feedback	0 to hex FFFFFFFF

Key Parameters:

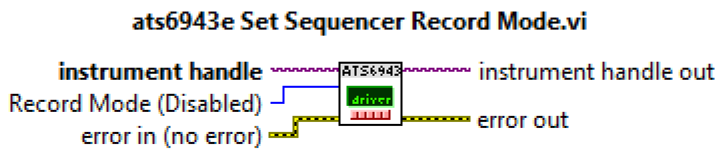
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Attribute		This control specifies the sequencer attribute to program.	0 = Jump Pass Fail 1 = Phase 3 Mode 2 = Window 3 Mode 3 = Window 3 Delay 4 = CRC Pre Load 5 = CRC Feedback 6 = Capture Mask
Attribute Value	U32	This control programs the specified attribute value.	See description above

C Function Prototype Form:

ViStatus ats6943e_setSequencerAttribute (ViSession instrumentHandle, ViInt16 attribute, ViUInt32 attributeValue);

Set Sequencer Record Mode

LabVIEW Diagram:



Description:

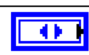
This vi programs the sequencer record mode settings.

The sequencer record modes are described below:

Record Disabled	When the sequence step record mode is set to None or Count, then the record memory will be not be written to.
Record Non-Error	When the sequence step record mode is set to None or Count, then the record memory will be set to zero.

The sequence step record mode is set using the **Set Sequence Record Mode** vi.

Key Parameters:

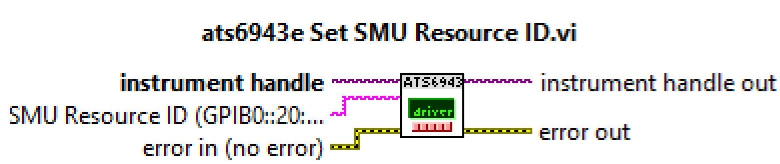
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Record Mode		This control specifies the sequencer record mode setting.	0 = Record Disabled 1 = Recorded Non-Error

C Function Prototype Form:

ViStatus ats6943e_setSequencerRecordMode (ViSession instrumentHandle, ViInt16 recordMode);

Set SMU Resource ID

LabVIEW Diagram:



Description:

This function specifies the SMU resource name used for calibrating the voltage reference, PMU and Active Load.

The SMU can be any of the Keithley Model 24nn series.

Key Parameters:

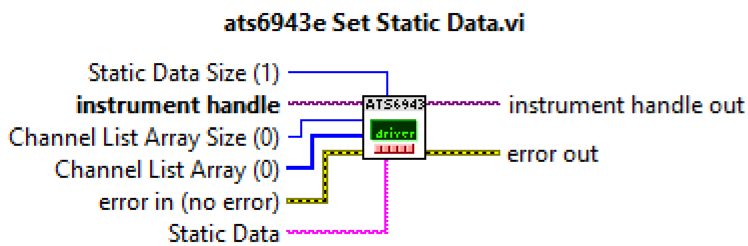
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session	0 to 2^{32}-1
SMU Resource	abc	This control specifies the SMU instrument resource name.	

C Function Prototype Form:

ViStatus ats6943e_setSmuResourceId (ViSession instrumentHandle, ViString SMUResourceID);

Set Static Data

LabVIEW Diagram:



Description:

This vi programs the static pattern.

Static data is expressed as an ASCII code described below:

Pin Action	Static Data Code
Disable channel	'Z'
Drive Low	'0'
Drive High	'1'

The channel list specifies which pins to program and the order with respect to the static data.

There is a one to one correspondence between the data in channel list array and the static data array. The static code in index n of the static data array contains the static output for the channel specified in index n of the channel list array, i.e., for every channel in the channel list array n = 0 to (channel list size - 1)

Static Output for Channel @ ChannelArray[n] = Static Code @ StaticData[n]

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Static Data Size	I16	This control specifies the number of pin codes in the static data array.	1 to 32 Should be >= Channel List Array Size.
Static Data	abc	This control contains the static data to program.	See description above

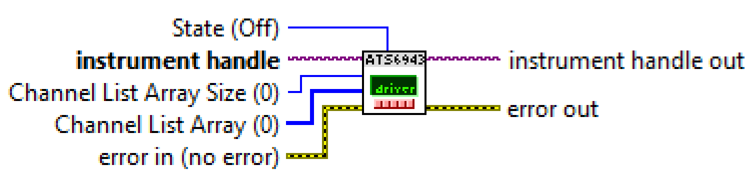
C Function Prototype Form:

ViStatus ats6943e_setStaticData (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 staticDataSize, ViChar staticData[]);

Set Static Mode

LabVIEW Diagram:

ats6943e Set Static Mode.vi



Description:

This vi programs the static mode enable for the specified channels.

When the Static Mode Enable is set on, the designated channel is put into the Static Mode and whatever is currently in the Static Broadside Stimulus Register will be applied to the output. Channels not in Static Mode will operate in the normal dynamic mode. When the channel is returned from Static to Dynamic Mode, dynamic operation will resume as though it had never been put into the Static Mode.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32

Name	Type	Description	Value
State		This control specifies the static mode state.	0 = Off 1 = On

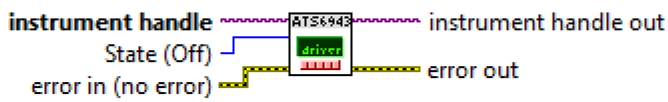
C Function Prototype Form:

ViStatus ats6943e_setStaticMode (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViInt16 state);

Set Static State

LabVIEW Diagram:

ats6943e Set Static State.vi



Description:

This vi programs static state of the data sequencer.

The static state is used to enable or disable static operation of all channels whose function is set to “Dynamic HiZ” or “Dynamic VTT”.

When turned off, all channels are set to dynamic operation.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
State		This control is used to set the static state of the sequencer.	0 = Off 1 = On

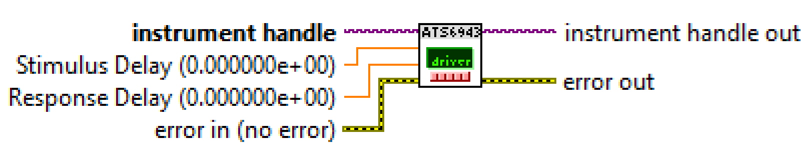
C Function Prototype Form:

ViStatus ats6943e_setStaticState (ViSession instrumentHandle, ViInt16 state);

Set Static Timing

LabVIEW Diagram:

ats6943e Set Static Timing.vi



Description:

This vi programs the static timing for the specified sequencer.

The stimulus delay is included for legacy support.

The response delay can be set from 0 (disabled) to 6.5535ms with 100ns resolution.

DTS Operation:

Primary Only

Key Parameters:

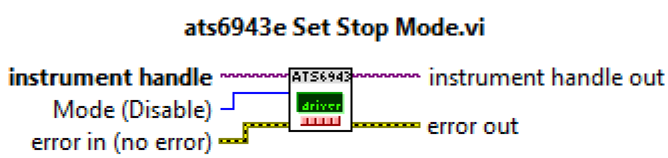
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Stimulus Delay	DBL	This control is included for legacy support.	
Response Delay	DBL	This control specifies the static response delay.	0 to 6.5535ms

C Function Prototype Form:

ViStatus ats6943e_setStaticTiming (ViSession instrumentHandle, ViReal64 stimulusDelay, ViReal64 responseDelay);

Set Stop Mode

LabVIEW Diagram:



Description:

This vi programs the stop mode of the data sequencer.

The stop mode determines what action a CPU generated stop or an external trigger will perform. Four modes are defined:

1. The stop signal can be ignored.
2. The stop signal can cause the current sequence burst to terminate at the end of the next pattern.
3. The stop signal can cause the next jump to be ignored. Sequence execution will continue at the step following the jump step.
4. The stop signal can cause the current sequence burst to terminate at the end of the sequence of a continuous or looped burst.

DTS Operation:

All coupled modules.

Key Parameters:

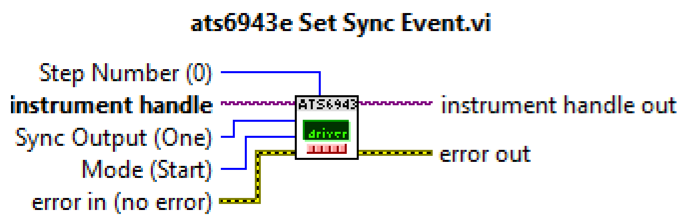
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Mode		This control specifies the stop mode to set.	0 = Disabled 1 = Stop after next pattern 2 = Stop looping and continue 3 = Stop at end of sequence

C Function Prototype Form:

ViStatus ats6943e_setStopMode (ViSession instrumentHandle, ViInt16 mode);

Set Sync Event

LabVIEW Diagram:



Description:

This vi sets the sync output event.

There are two sync outputs that can be routed to any of the AUX or TTLTRG outputs.

The sync pulse mode can be set to the following:

Start	The sync pulse begins from the start of the sequence.
Single Step	A sync pulse will be generated the first time the specified sequence step is executed.
Continuous Step	A sync pulse will be generated every time the specified sequence step is executed.

Key Parameters:

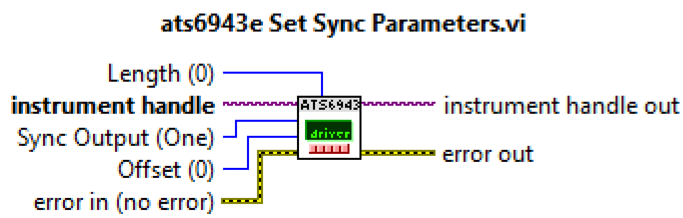
Name	Type	Description	Value
instrument handle	I/0	Identifier to a device I/O session.	0 to 2^{32}-1
Sync Output		This control specifies which sync output to set.	0 = Sync1 1 = Sync2
Mode		This control specifies the sync output mode.	0 = Start 1 = Single Step 2 = Continuous Step.
Step Number	I16	This control specifies the sync step number. This control is ignored if the mode is set to "Start".	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_setSyncEvent (ViSession instrumentHandle, ViInt16 syncOutput, ViInt16 mode, ViInt16 stepNumber);

Set Sync Parameters

LabVIEW Diagram:



Description:

This vi sets the sync output parameters.

There are two sync outputs that can be routed to any of the AUX or TTLTRG outputs.

The sync pulse parameters consist of an offset and a length. Once the programmed sync event occurs, the sync pulse will begin after the "offset" and last for "length". Both "offset" and "length" are specified in pattern clocks.

The sync pulse will not extend past the end of the sequence. In the "Step" event, the sync pulse will not extend beyond the specified step.

Key Parameters:

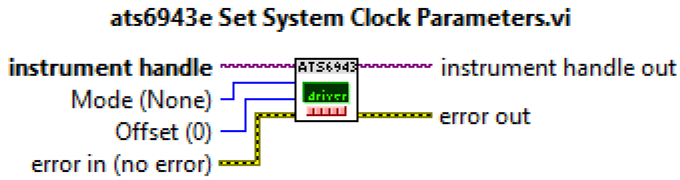
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Sync Output		This control specifies which sync output to set.	0 = Sync1 1 = Sync2
Offset	I32	This control specifies the sync offset in patterns.	0 to 1048575
Length	I16	This control specifies the sync length in patterns.	0 to 4095

C Function Prototype Form:

ViStatus ats6943e_setSyncParameters (ViSession instrumentHandle, ViInt16 syncOutput, ViInt32 offset, ViInt16 length);

Set System Clock Parameters

LabVIEW Diagram:



Description:

This vi programs the external system clock parameters of the sequencer.

There are two system clock parameters:

1. Mode
2. Offset

Mode:

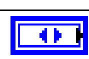
The mode allows the user to select the active external system clock edge:

1. Rising Edge.
2. Falling Edge
3. Both Edges
4. Divide by 2 Rising Edge
5. Divide by 2 Falling Edge

Offset:

The offset allows the user to shift the system clock in order to align the clock/data relationship. The resolution is 1/2 the master clock period.

Key Parameters:

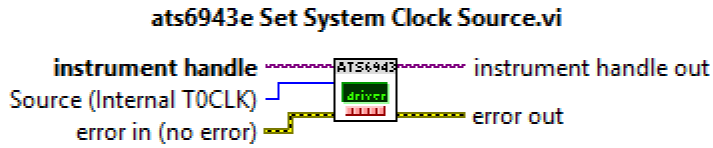
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Mode		This control is used to program the system clock mode.	0 = None 1 = Rising Edge 2 = Falling Edge 3 = Both Edges 4 = Divide by 2 Rising Edge 5 = Divide by 2 Falling Edge
Offset		This control programs the system clock offset. Resolution is 1/2 the master clock period.	0 to 65534

C Function Prototype Form:

ViStatus ats6943e_setSystemClockParameters (ViSession instrumentHandle, ViInt16 mode, ViUInt16 offset);

Set System Clock Source

LabVIEW Diagram:



Description:

This vi programs the system clock source of the sequencer.

The system clock is used to trigger the timing set phase and window logic.

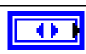
The system clock can be set to the following sources;

Internal T0CLK	The period is defined in the sequence step memory using the **Set Sequence Clock** vi.
Pulse Generator	The period is defined by the pulse generator using the **Set Pulse Parameters**, **Set Pulse Period**, **Set Pulse Delay** and **Set Pulse Width** vi.
Frequency Synthesizer	The period is defined by the frequency synthesizer using the **Set Frequency Synthesizer** vi.
AUX1 - AUX12	The period is defined by the AUX input signal using the **Set System Clock Parameters** vi.

A pattern clock is also generated from this clock, every 'CPP' system clock periods. CPP (Clocks per Pattern) is defined in the **Set Sequence Clock** vi. Note, when CPP = 1, then pattern clock is equal to system clock. When CPP = 3, then pattern clock period is three times the system clock period.

Timing set phases can trigger on either system clock or pattern clock. Windows only trigger on pattern clock.

Key Parameters:

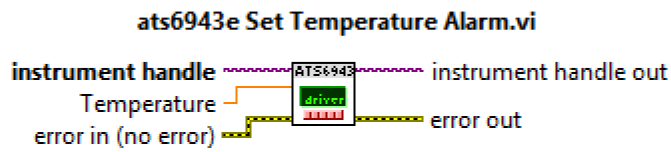
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Source		This control specifies which signal to select as the system clock source.	0 = Internal T0CLK 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 14 = Pulse Generator 15 = Frequency Synthesizer

C Function Prototype Form:

ViStatus ats6943e_setSystemClockSource (ViSession instrumentHandle, ViInt16 source);

Set Temperature Alarm

LabVIEW Diagram:



Description:

This vi sets the temperature alarm trip point for the front-end board from 70 to 130 degrees Celsius with a resolution of 10.

The power relay will open if the temperature exceeds the trip value.

Key Parameters:

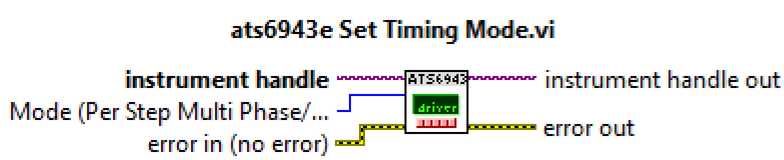
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Temperature	DBL	This control specifies the temperature at which the alarm will trip.	70 to 130

C Function Prototype Form:

ViStatus ats6943e_setTemperatureAlarm (ViSession instrumentHandle, ViReal64 temperature_C);

Set Timing Mode

LabVIEW Diagram:



Description:

This vi programs the timing mode of the sequencer.

There are three timing modes available:

Per Step Multi	1024 sequence steps with one timing set per step. Four phase/window signals per timing set.

Per Step Single	4096 sequence steps with one timing set per step. One phase/window signals per timing set.
Indexed	4096 sequence steps with 256 timing sets indexed. Four phase/window signals per timing set.

Key Parameters:

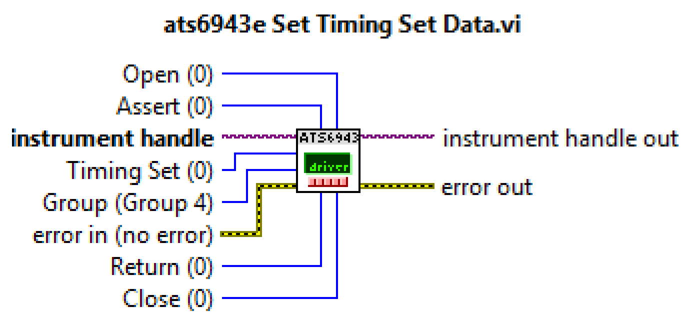
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Source		This control specifies which timing mode to set.	0 = Per Step Multi 1 = Per Step Single 2 = Indexed

C Function Prototype Form:

ViStatus ats6943e_setTimingMode (ViSession instrumentHandle, ViInt16 mode);

Set Timing Set Data

LabVIEW Diagram:



Description:

This vi programs the stimulus and capture settings for the specified timing set and group when the timing mode is set to "Indexed".

Each indexed timing set contains four signal groups. Each signal group contains four settings:

1. Stimulus assert
2. Stimulus return
3. Capture open
4. Capture close

The timing mode is programmed using the **Set Timing Mode** vi. If the timing mode is set to "Per Step Multi" or "Per Step Single", then use the **Set Sequence Timing Data** vi.

Key Parameters:

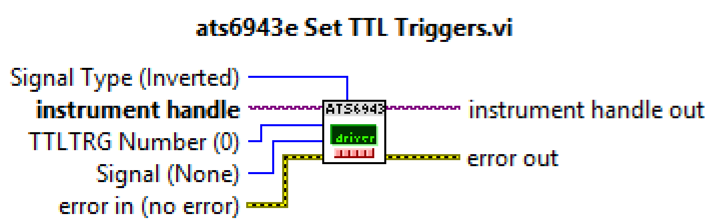
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Timing Set	U16	This control specifies the timing set number to program.	0 to 255
Group	◄►	This control specifies which timing group to set.	0 = Group 1 1 = Group 2 2 = Group 3 3 = Group 4
Assert	I32	This control is used to set the phase assert time.	0 to 65535
Return	I32	This control is used to set the phase return time.	0 to 65535
Open	I32	This control is used to set the window open time.	0 to 65535
Close	I32	This control is used to set the window close time.	0 to 65535

C Function Prototype Form:

ViStatus ats6943e_setTimingSetData (ViSession instrumentHandle, ViInt16 timingSet, ViInt16 group, ViInt32 assert, ViInt32 return, ViInt32 open, ViInt32 close);

Set TTL Triggers

LabVIEW Diagram:

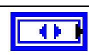


Description:

This vi sets the TTLTRG signal source and routing.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Trigger	I16	This control specifies which TTL trigger to program.	0 to 7
Signal	◄►	This control specifies the selected TTLTRG signal source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4

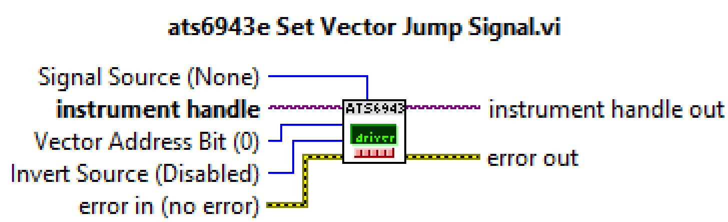
Name	Type	Description	Value
			5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = Halted 14 = Static Pulse 15 = Pulse Generator 16 = Sequence Flag 1 17 = Sequence Flag 2 18 = Sync 1 19 = Sync 2 20 = CHT1 21 = CHT2 22 = CHT3 23 = CHT4 24 = Idle Active 25 = Sequence Active 26 = Error Pulse 27 = Pass Valid 28 = Sequence Reset 29 = DTS Sync 30 = Driver Disable 31 = Master Reset
Signal Type		This control is used specify the signal type.	0 = Normal 1 = Inverted

C Function Prototype Form:

ViStatus ats6943e_setTtlTriggers (ViSession instrumentHandle, ViInt16 TTLTRGNumber, ViInt16 signal, ViInt16 signalType);

Set Vector Jump Signal

LabVIEW Diagram:



Description:

This vi configures one of the four vector jump signals of the selected data sequencer.

The four vector signals comprise an index in to a vector jump table that specifies the jump address as well as the timing set (indexed timing mode only). The vector table/signals are only used if the vector jump bit is set during a sequence jump step.

Configuring the vector signal consists of the following:

- Selecting the signal source.
- Program the source inverter.

Key Parameters:

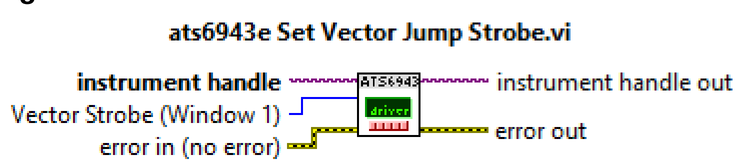
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Vector Address Bit	I16	This control specifies the vector address bit to program.	0 to 3
Invert Source		This control is used to enable/disable the source inverter.	0 = Disabled 1 = Enabled
Signal Source		This control specifies the vector signal source.	0 = None 1 = AUX1 2 = AUX2 3 = AUX3 4 = AUX4 5 = AUX5 6 = AUX6 7 = AUX7 8 = AUX8 9 = AUX9 10 = AUX10 11 = AUX11 12 = AUX12 13 = CHT1 16 = TTLTRG0 17 = TTLTRG1 18 = TTLTRG2 19 = TTLTRG3 20 = TTLTRG4 21 = TTLTRG5 22 = TTLTRG6 23 = TTLTRG7

C Function Prototype Form:

ViStatus ats6943e_setVectorJumpSignal (ViSession instrumentHandle, ViInt16 vectorAddressBit_VAn, ViInt16 invertSource, ViInt16 signalSource);

Set Vector Jump Strobe

LabVIEW Diagram:

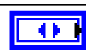


Description:

This vi selects one of the four timing set windows as the vector strobe.

The closing edge of the selected window will sample the four vector signals VA0 (LSB) to VA3 (MSB). The vector signals are only used if the vector jump bit is set during a sequence jump step. The vector signals form an address in to the vector table to determine the jump step and timing set (if timing mode set to indexed).

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Vector Strobe		This control is used to select the vector strobe signal.	0 = Window 1 1 = Window 2 2 = Window 3 3 = Window 4

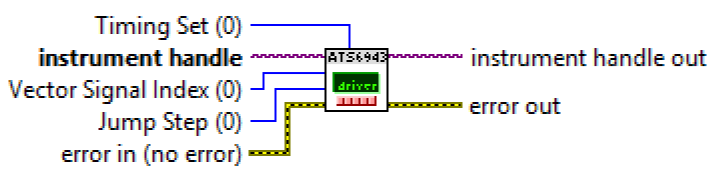
C Function Prototype Form:

ViStatus ats6943e_setVectorJumpStrobe (ViSession instrumentHandle, ViInt16 vectorStrobe);

Set Vector Jump Table

LabVIEW Diagram:

ats6943e Set Vector Jump Table.vi



Description:

This vi configures one of the sixteen vector jump table entries for the selected data sequencer.

The vector table is indexed by the four vector signals VA0 (LSB) to VA3 (MSB). Each vector table entry supplies the jump address as well as the timing set (indexed timing mode only). The vector table/signals are only used if the vector jump bit is set during a sequence jump step.

Configuring the vector table signal consists of the following:

- Selecting the jump step.
- Program the timing set. (only used in the indexed timing mode)

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Vector Signal Index Bit	I16	This control specifies the vector signal index to program.	0 to 15
Jump Step	I16	This control is used to program the vector table jump step.	0 to 4095
Timing Set	I16	This control specifies the timing set for the jump step if the timing mode is set to "Indexed".	0 to 255

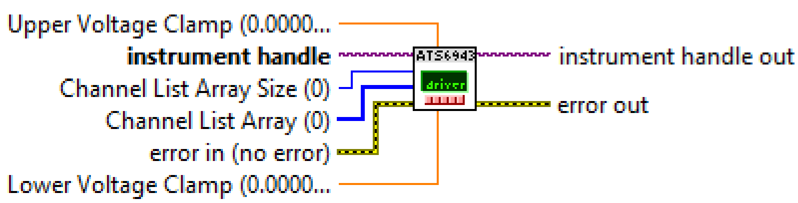
C Function Prototype Form:

ViStatus ats6943e_setVectorJumpTable (ViSession instrumentHandle, ViInt16 vectorSignalIndex, ViInt16 jumpStep, ViInt16 timingSet);

Set Voltage Clamps

LabVIEW Diagram:

ats6943e Set Voltage Clamps.vi



Description:

This vi programs the upper and lower voltage clamps for the specified channels.

The channel function must be set force current (PMUFI)

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Channel List Array Size	I32	The number of elements in the "Channel List Array" parameter.	-1 to 32 -1 and 0 specifies all channels.
Channel List Array	[I32]	Array contains the channel numbers to program.	1 to 32
Upper Voltage Clamp	DBL	This control specifies the upper voltage clamp value (V).	-2.5 to +7.5
Lower Voltage Clamp	DBL	This control specifies the lower voltage clamp value (V).	-2.5 to +7.5

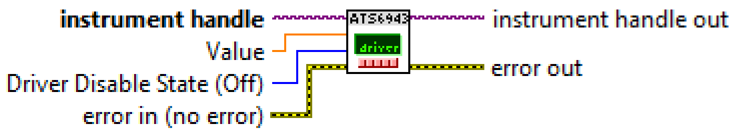
C Function Prototype Form:

ViStatus ats6943e_setPmuVClamps (ViSession instrumentHandle, ViInt32 channelListArraySize, ViInt32 channelListArray[], ViReal64 upperVoltageClamp, ViReal64 lowerVoltageClamp);

Set Watchdog Timer

LabVIEW Diagram:

ats6943e Set Watchdog Timer.vi



Description:

This vi programs the watchdog timeout value and driver disable state.

The watchdog timer will set bit 6 (WDTO) in the sequence event register if the sequence active time exceeds the specified value.

A WDTO event can optionally generate a driver disable pulse that disables the output drivers of the local sequencer. Multiple sequencers can be coupled by assigning "Driver Disable" to a common PXI bus signal.

The timeout is programmed with a range of 40ns to 4000s.

The resolution is set based on the timeout:

- Timeout between 40ns and 10ms, resolution = 20ns
- Timeout between 10ms and 10s, resolution = 100ns
- Timeout between 1s and 4000s, resolution = 1us

Regardless of the resolution, the watchdog timer has an accuracy of 30ns with a range of 40ns to 4000s.

The Watchdog Timeout Timer starts when SEQACT begins. If the driver disable is turned on, it disables all 32 drivers (any active load or resistive loading remains).

DTS Operation:

Primary, other sequencers optional.

Key Parameters:

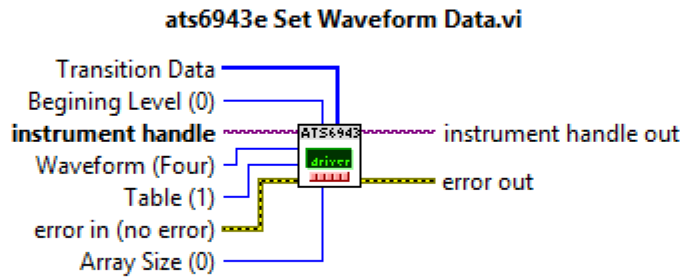
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Value	DBL	This control specifies the watchdog timeout value.	40ns to 4000s
Driver Disable State		This control specifies the sequence timeout state.	0 = Off 1 = On

C Function Prototype Form:

ViStatus ats6943e_setWatchdogTimer (ViSession instrumentHandle, ViReal64 value, ViInt16 driverDisableState);

Set Waveform Data

LabVIEW Diagram:



Description:

This vi programs the specified waveform table data.

The waveform is programmed by specifying the beginning level and the bit number of subsequent transitions.

Example 1:

Beginning Level = 0;

Array Size = 3;

Transition Array = {5, 10, 15};

Would generate the following waveform;

"00000111110000011111111..."

Bits 1-5 low

Bits 6-10 high

Bits 11-15 low

Bits 16 through the size of the table high.

Example 2:

Beginning Level = 1;

Array Size = 0;

Transition Array = {Empty};

Would generate the following waveform;

"111..."

Bits 1 through the size of the table high.

Key Parameters:

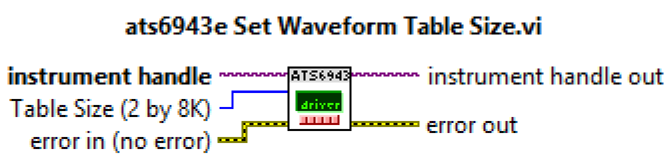
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Waveform		This control specifies which waveform to program.	0 = Waveform 1 1 = Waveform 2 2 = Waveform 3 3 = Waveform 4
Table	I16	This control specifies the table number to program.	1 to 16
Array Size	I32	This control is used to indicate the number of elements in the transition data array.	0 to 16384
Beginning Level	I16	This control specifies the beginning level of the waveform data.	0 to 1
Transition Data	[I32]	This control specifies the transition bits of the specified waveform table.	See description above.

C Function Prototype Form:

ViStatus ats6943e_setWaveformData (ViSession instrumentHandle, ViInt16 waveform, ViInt16 table, ViInt16 beginingLevel, ViInt32 arraySize, ViInt32 transitionData[]);

Set Waveform Table Size

LabVIEW Diagram:



Description:

This function programs the waveform table size.

Waveforms can be segmented and assigned in multiple tables. The valid segmenting is listed below:

- 16 Tables by 1024 bits per table.
- 8 Tables by 2048 bits per table.
- 4 Tables by 4096 bits per table.
- 2 Tables by 8192 bits per table.
- 1 Table by 16384 bits.

Key Parameters:

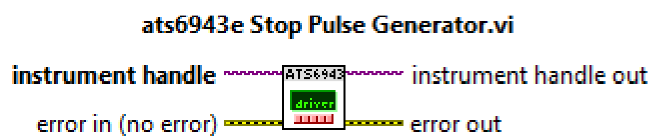
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
Table Size		This control specifies the waveform table size.	0 = 16 by 1K 1 = 8 by 2K 2 = 4 by 4K 3 = 2 by 8K 4 = 1 by 16K

C Function Prototype Form:

ViStatus ats6943e_setWaveformTableSize (ViSession instrumentHandle, ViInt16 tableSize);

Stop Pulse Generator

LabVIEW Diagram:



Description:

This vi stops the pulse generator.

Key Parameters:

Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_stopPulseGenerator (ViSession instrumentHandle);

Stop Sequence

LabVIEW Diagram:

Description:

This function generates a stop command to the sequencer.

The stop mode must be set prior to calling this function using **Set Stop Mode**.

DTS Operation:

Primary only.

Key Parameters:

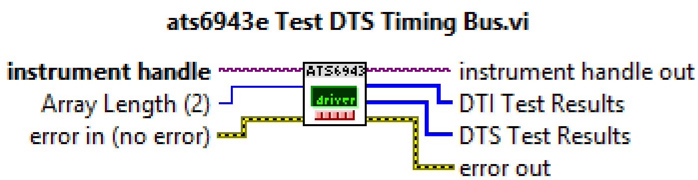
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_stopSequence (ViSession instrumentHandle);

Test DTS Timing Bus

LabVIEW Diagram:



Description:

This function performs a DTI and DTS timing bus test to verify the external ETB timing bus and internal connections used for DTS timing.

There must be at least as many elements in the DTI and DTS arrays as is specified in the "Array Length" control.

The DTI test is performed using a special module inter connect mode. After the DTI test is run, the current module inter connect setting is restored.

The DTS test is performed using the current module inter connect settings for each DTI.

The test results are returned in two arrays. Element n of each array corresponds to instrument n in the handle array. A bit set high indicates a signal is not active.

Bit Number	Signal
0	Phase 1
1	Phase 2
2	Phase 3
3	Phase 4
4	Window 1
5	Window 2
6	Window 3
7	Window 4

Bit Number	Signal
8	SEQ_CLK
9	SEQ_CLK_D
10	T0_CLK
11	Jump

Key Parameters:

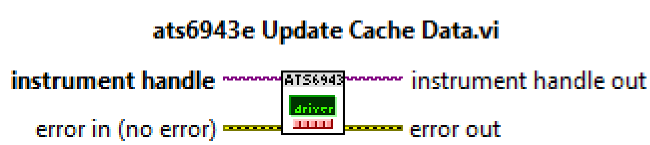
Name	Type	Description	Value
Array Length	I16	Size of the array.	2 to 13
DTI Test Results	[U32]	Array of instrument handles.	See description above.
DTS Test Results	[U32]	Array of instrument handles.	See description above.

C Function Prototype Form:

ViStatus ats6943e_testDrsTimingBus (ViSession *DTSArray, ViInt16 arrayLength, ViInt32 DTITestResults[], ViInt32 DTSTestResults[]);

Update Cache Data

LabVIEW Diagram:



Description:

This vi writes the current cache data to the hardware.

Key Parameters:

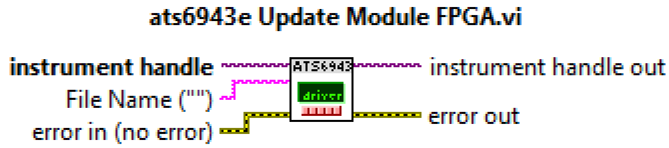
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1

C Function Prototype Form:

ViStatus ats6943e_updateCacheData (ViSession instrumentHandle);

Update Module FPGA

LabVIEW Diagram:



Description:

This vi will update the FPGA with the file specified.

The FPGA updated is based on the file name:

if the file name contains "801045-001", then the PXIe Bridge FPGA eprom will be programed. Chassis power must be cycled for the new firmware to load.

if the file name contains "801058_001", then the sequence logic FPGA eprom will be programed. After the eprom is programmed the FPGA will be re-loaded.

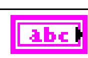
if the file name contains "801057_001", then the inter module control primary FPGA eprom will be programed. After the eprom is programmed the FPGA will be re-loaded if the DTI is configured as a primary from the front panel ETB connectors.

if the file name contains "801057_002", then the inter module control secondary FPGA eprom will be programed. After the eprom is programmed the FPGA will be re-loaded if the DTI is configured as a secondary from the front panel ETB connectors.

if the file name contains "801057_003", then the inter module control terminator FPGA eprom will be programed. After the eprom is programmed the FPGA will be re-loaded if the DTI is configured as a terminator from the front panel ETB connectors.

if the file name contains "801029-001", then the digital board bridge FPGA eprom will be programed. After the eprom is programmed the FPGA will be re-loaded.

Key Parameters:

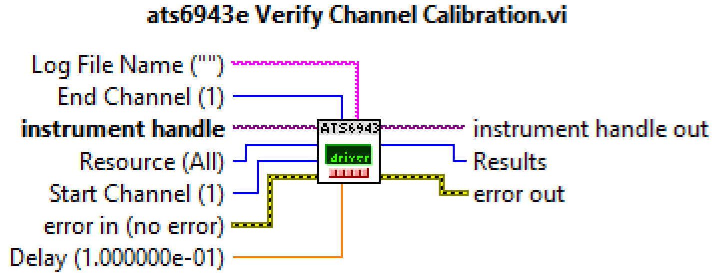
Name	Type	Description	Value
instrument handle		Identifier to a device I/O session.	0 to 2^{32}-1
File Name		This control specifies the file to load.	

C Function Prototype Form:

ViStatus ats6943e_updateModuleFpga (ViSession instrumentHandle, ViChar fileName[]);

Verify Channel Calibration

LabVIEW Diagram:

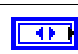


Description:

This function performs the calibration validation on the programmable front-end modules.

Prior to calling this function, the user should allow the front-end to warm up. A warmup period is recommended so that validation is performed at operating temperature.

Key Parameters:

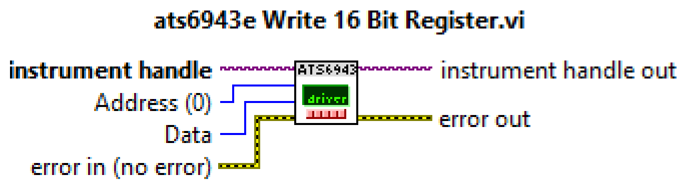
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Resource		Resource to verify.	0 – ALL 2 – Measure Voltage 6 – Driver Levels 7 – Comparator Levels
Start Channel	I16	First Channel to calibrate.	1 to 32
End Channel	I16	Last Channel to calibrate.	Start Channel to 32
Delay	DBL	Delay applied before measurement for settling.	0.005 to 1s
File Name	abc	This control specifies the log file path and name that the validation results will be written to.	

C Function Prototype Form:

ViStatus ats6943e_verifyChannelCalibration (ViSession instrumentHandle, ViInt16 resource, ViInt16 startChannel, ViInt16 endChannel, ViReal64 delay_s, ViString logFileName, ViInt32 *results);

Write 16 Bit Register

LabVIEW Diagram:



Description:

This function programs the contents of a single 16 bit BAR0 register address.

Key Parameters:

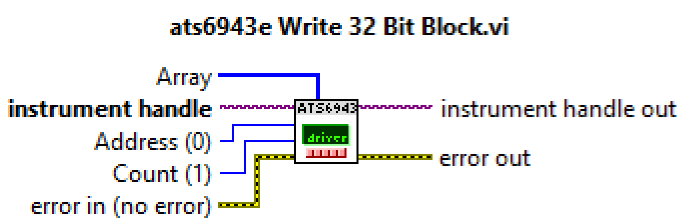
Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Address	I32	This control specifies the address to write.	0 to hex 1001000
Data	I16	This control contains the data to write at the address.	0 to 65535

C Function Prototype Form:

ViStatus ats6943e_write16_data (ViSession instrumentHandle, ViInt32 address, ViInt16 data);

Write 32 Bit Block

LabVIEW Diagram:



Description:

This function returns the contents of a block of 32 bit BAR0 register address.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Address	I32	This control specifies the address to write.	0 to hex 1001000
Count	I32	This control specifies the number of locations to write.	1 to (Size – Start Address) See description above

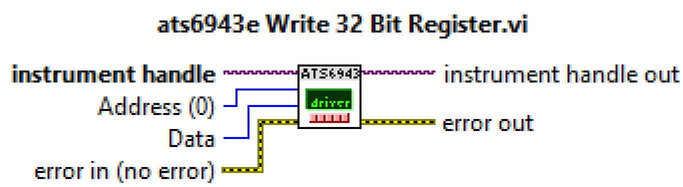
Name	Type	Description	Value
Array	[I32]	This control contains the data to write at the address.	0 to 4294967295

C Function Prototype Form:

ViStatus ats6943e_write_block32_data (ViSession instrumentHandle, ViInt32 address, ViInt32 count, ViInt32 array[]);

Write 32 Bit Register

LabVIEW Diagram:



Description:

This function programs the contents of a single 32 bit BAR0 register address.

Key Parameters:

Name	Type	Description	Value
instrument handle	I/O	Identifier to a device I/O session.	0 to 2^{32}-1
Address	I32	This control specifies the address to write.	0 to hex 1001000
Data	I32	This control contains the data to write at the address.	0 to 4294967295

C Function Prototype Form:

ViStatus ats6943e_write32_data (ViSession instrumentHandle, ViInt32 address, ViInt32 data);

Chapter 5
Channel Calibration

This chapter provides calibration and verification information for the programmable driver/receiver characteristics. The following table lists the calibration functions:

Calibration Function
ADC Reference
Load Reference
ADC Gain
Load PGIA
Measure Voltage
DAC Overlap
PMU
Drive Levels
Compare Levels
Active Load
Delete

Table 5-1 Calibration Functions

CAUTION

ALWAYS PERFORM DISASSEMBLY, REPAIR AND CLEANING AT A STATIC SAFE WORKSTATION.

Performance Verification

Do not attempt to calibrate the instrument before verifying first that the instrument is in working order. A complete set of specifications is listed in Appendix A. If the instrument fails to perform within the specified limits, the instrument must be tested to find the source of the problem.

If there is a reasonable suspicion that an electrical problem exists within the DTI, perform a complete self-test on the instrument prior to running a verification or calibration procedure.

Environmental Conditions

The DTI can operate over an ambient temperature range of 0°C to 45°C.

Adjustments should be performed under laboratory conditions having an ambient temperature of 25°C, ±5°C and at relative humidity of less than 80%. Turn on the power to the DTI and allow it to warm up to the desired operating temperature before beginning the adjustment procedure. If the instrument has been subjected to conditions outside these ranges, allow additional time for the instrument to stabilize before beginning the calibration procedure.

Warm-up Period

Most equipment is subject to a small amount of drift when it is first turned on. To ensure accuracy, turn on the power to the T940 module and allow it to warm-up to the desired operating temperature before beginning the calibration procedure.

Required Test Equipment

The required equipment for calibration is listed in Table 5-2. Test instruments other than those listed may be used only if their specifications equal or exceed the required characteristics. Also listed below are accessories required for calibration.

Equipment	Model No.	Manufacturer
SourceMeter	2420-C or equivalent	Keithley
50 ohm MCX to Male BNC calibration cable 1M or less.	NA	Multiple
BNC Female to Dual Banana Plug Adapter.	NA	Multiple
50 ohm MCX breakout adapter	405654	Astronics Test Systems
50 ohm coax cable	SEAC-020-06-6.0-TU-TU	Samtec

Table 5-2 Required Calibration Equipment

Calibration Interval

The DTI should be calibrated at a regular time interval determined by the accuracy requirements of your application. A one-year interval is adequate for most applications. Accuracy specifications are valid only when calibration is performed at regular time intervals. Accuracy specifications presented herein are not valid beyond the one-year calibration interval. Astronics Test Systems does not recommend extending calibration intervals beyond three years.

Calibration Temperature

The DTI should be calibrated at the nominal temperature of your application. Application temperature can PXIe chassis characteristics, as well as the exact usage of the features of the module. Using more channels simultaneously at higher selected slew rates, for example, can create a higher operating temperature. For best accuracy, run the Soft Front Panel during a typical test execution and monitor the programmable channel temperatures. The module should settle on a temperature if the test is long enough to establish equilibrium. The highest of these temperatures should be used as the calibration temperature for best accuracy in similar applications.

Basic Setup

The DTI should be installed in a High Power PXIe chassis. Connect the breakout board to the DTI using the 50 ohm coax cable. Connect the BNC end of the coax calibration cable to the source meter using the BNC female to dual banana plug adapter. Connect the MCX end of the calibration cable to the EXTFORCE connector on the breakout adapter. An example configuration is shown Figure 5-1.

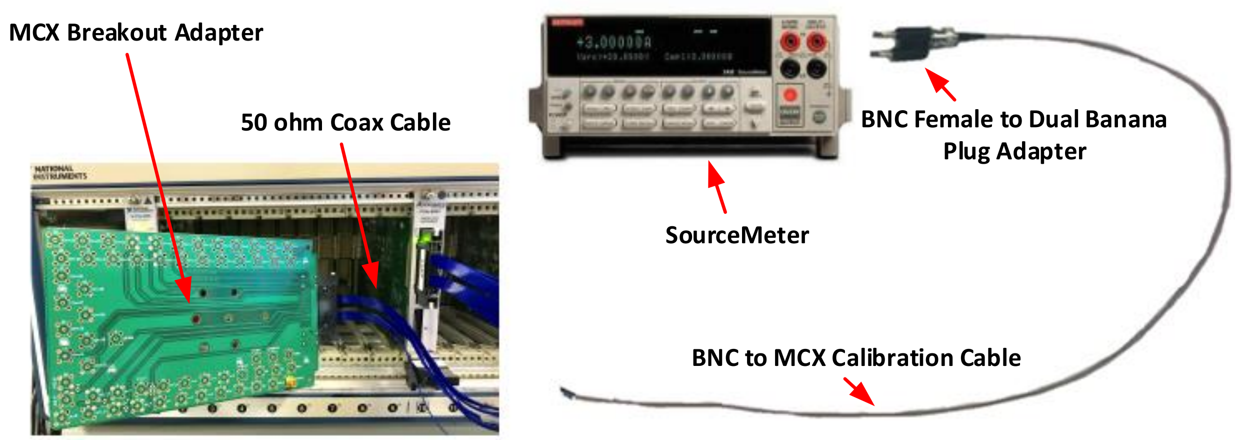


Figure 5-1 Setup

Calibration Procedures

Use the following procedures to calibrate the DTI module. Calibration is done with the DTI module installed in a PXIe chassis. The calibration procedure requires that the Soft Front Panel utility program be installed and interfaced to the instrument. The VISA library is required.

Calibration is performed from the Calibration Panel in the Soft Front Panel. To invoke this panel, access the Calibrate menu item from the Instrument menu.

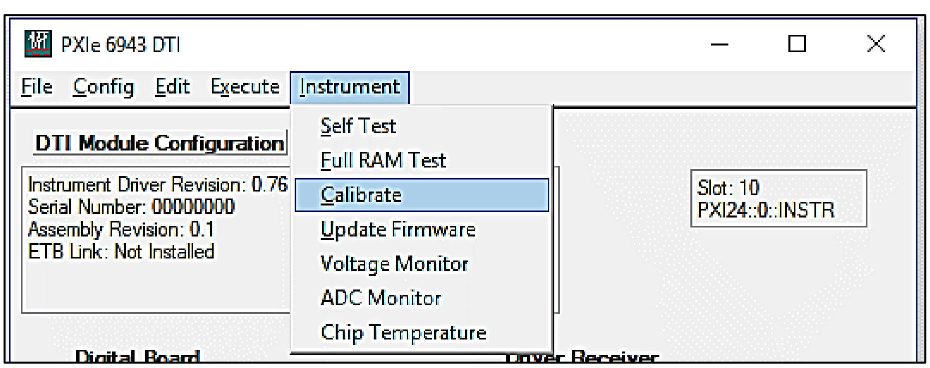


Figure 5-2: Invoke the Calibrate DTI Panel from the SFP

The Calibrate panel, before opening, will inform the user that calibration mode requires the instrument to be automatically reset to its power-on defaults. If the instrument settings need to be saved prior to calibration, or if the instrument is running a critical test, now is the time to exit. Select “Yes” if it is OK to continue, or “No” if calibration mode should be exited.

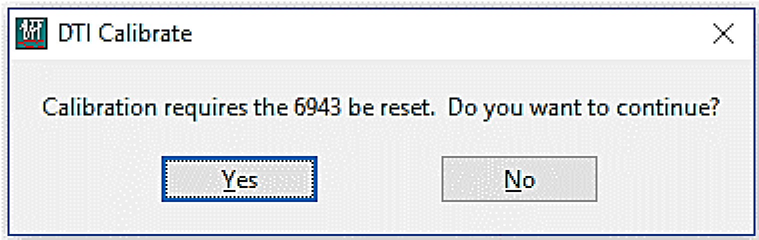


Figure 5-3 Calibration Prompt

Access the Calibrate Function to be performed using the Calibrate Function drop-down list. From this list, select the function to be calibrated.

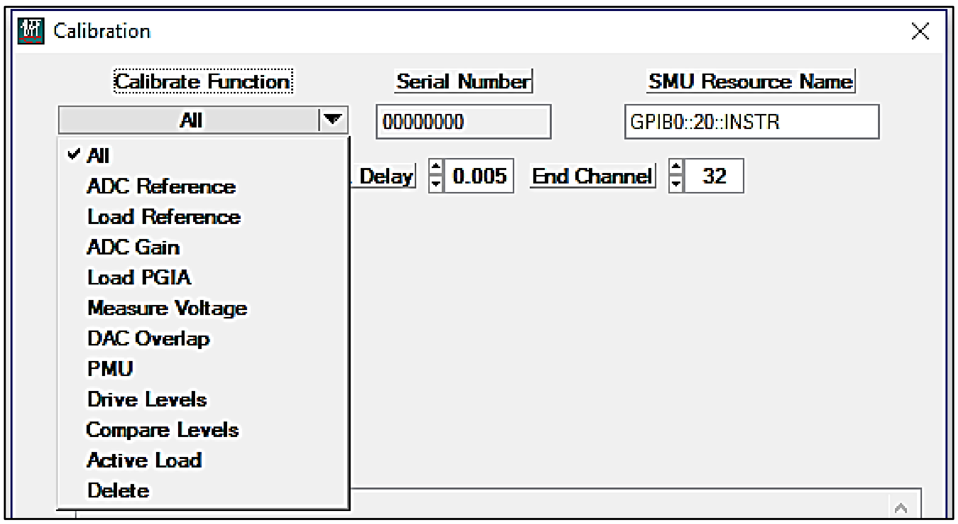


Figure 5-4 Calibration Function List

NOTE

Calibration procedures must be performed in the order shown in Figure 5-4. Changing the order of calibration from that which is shown can invalidate the results. Selecting “All” as the calibration function executes all the functions in the proper order.

After selecting the function, the start and end channel can be selected. The Delay control sets the settling time between test points.

Run Calibration Prompts

After depressing the **Run** command button. A confirmation prompt will ask if you want to run the selected calibration procedure. Select **Yes** to run and **No** to exit.

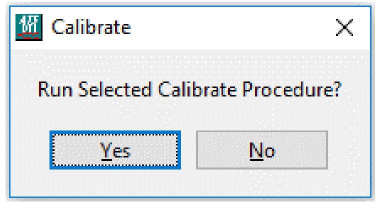


Figure 5-5 Calibration Run Confirmation

After confirming yes to run the calibration a calibration warmup prompt will display.

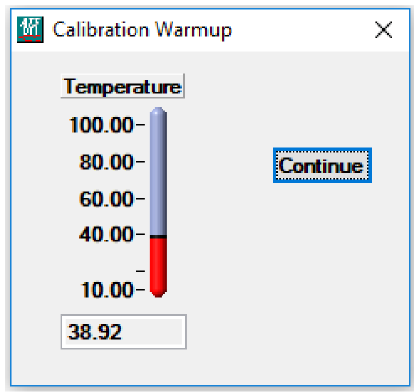


Figure 5-6 Calibration Warmup Prompt

This prompt will display until the temperature is > 80 or the Continue command button is pressed.

After warmup, an SMU connect prompt will display. Select **Yes** to run and **No** to exit.

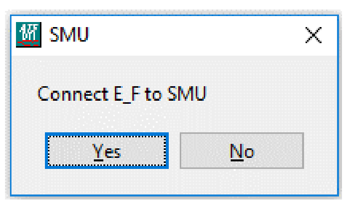


Figure 5-7 SMU Connect Prompt

NOTE

Not all calibration functions require the SMU but the SMU connect prompt will still display. The functions that do not require the SMU are ADC Gain, DAC Overlap, Drive Levels and Compare Levels.

Once a calibration is run, the values are stored in volatile memory. Depressing the **Update** command button will write the calibration values to non-volatile memory.

The sections to follow describe the individual procedures in detail.

ADC Reference

The ADC reference function characterizes the reference voltages that are used to calibrate and verify hardware.

There are four voltage references:

- +5.0V
- +3.33V
- +1.66V
- -1V

Each voltage reference is routed to the EXTFORCE pin and the Source Meter is programmed to take a voltage measurement.

1. Select ADC Reference from the **Calibrate Function** pull-down control. The current calibration values will be displayed.

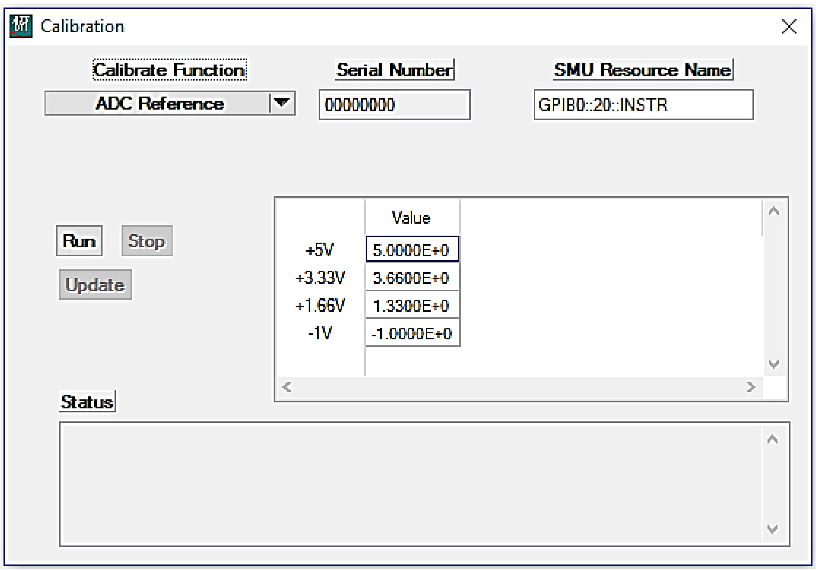


Figure 5-8 ADC Reference Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

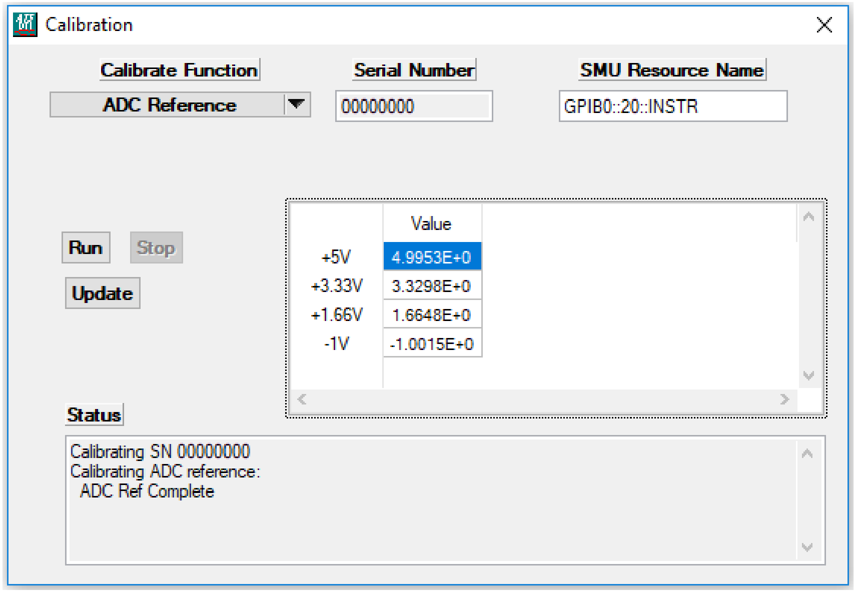


Figure 5-9 ADC Reference Complete

Load Reference

The load reference function characterizes the reference loads that are used to verify hardware.

There are two load references:

- 50Ω
- 10050Ω

Each load reference is connected to the EXTFORCE pin and the Source Meter is programmed to take a resistance measurement.

1. Select Load Reference from the **Calibrate Function** pull-down control. The current calibration values will be displayed.

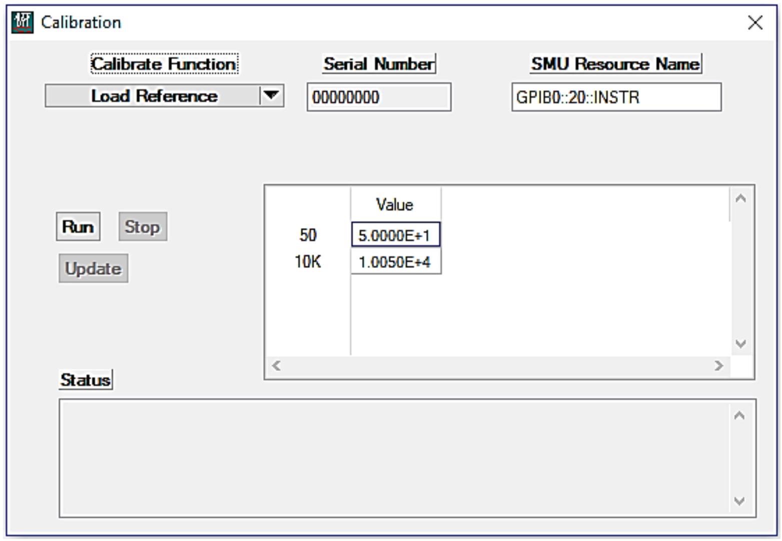


Figure 5-10 Load Reference Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

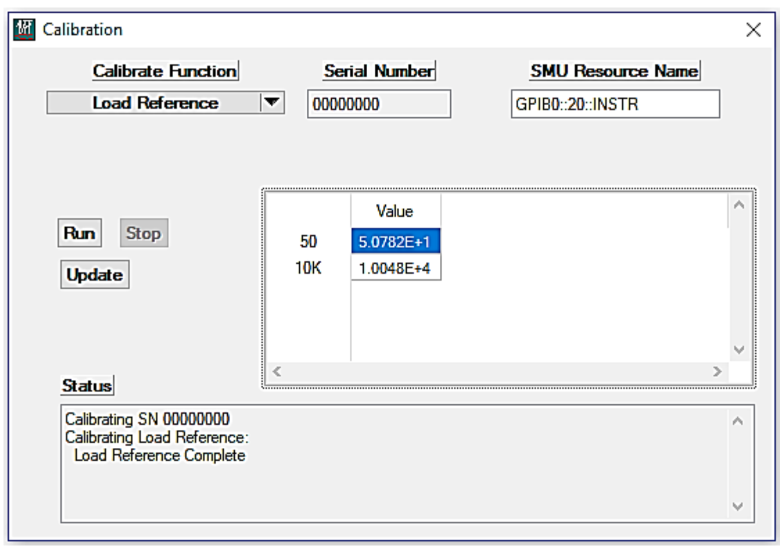


Figure 5-11 Load Reference Complete

ADC Gain

The ADC gain function calibrates the three ADC PGIA settings used by the DTI.

Two voltage references are routed to the ADC and measured. A two point algorithm is used to calculate an offset and gain for each ADC PGIA setting.

1. Select ADC Gain from the **Calibrate Function** pull-down control. The current calibration values will be displayed.

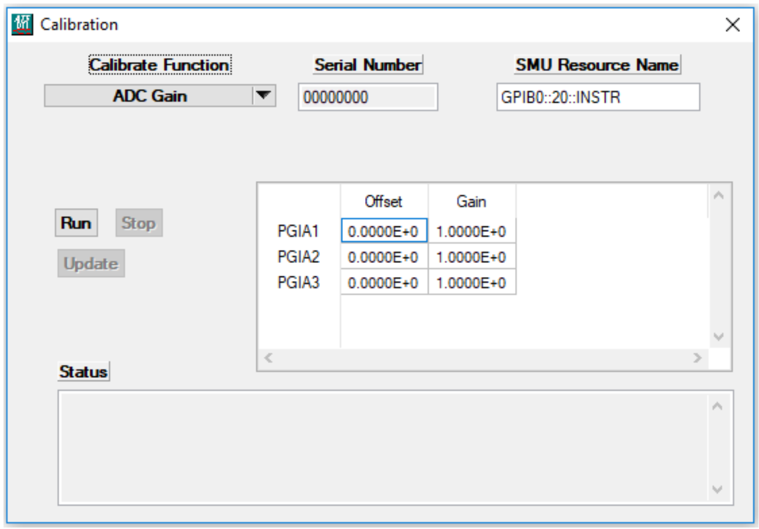


Figure 5-12 ADC Gain Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

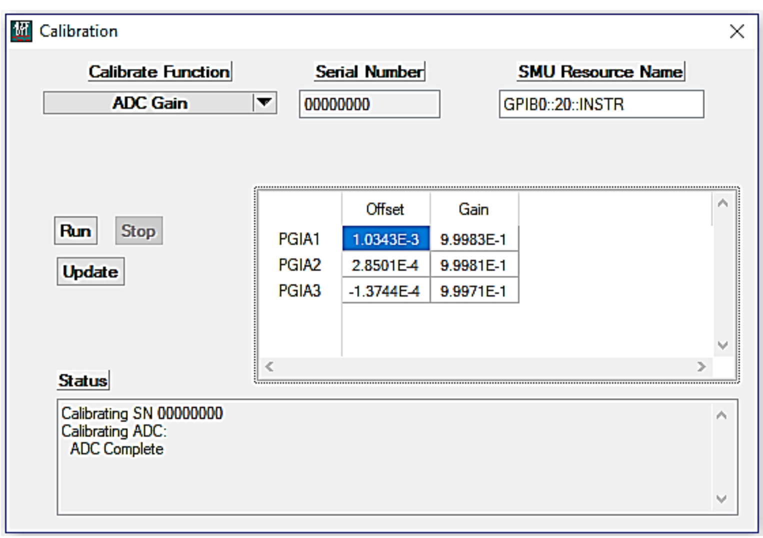


Figure 5-13 ADC Gain Complete

Load PGIA

The Load PGIA function calibrates the three load PGIA settings used by the DTI.

Two voltage references are routed to the EXTFORCE and the EXTFORCE load is enable. The ADC variable load input is measured. A two point algorithm is used to calculate an offset and gain for each load PGIA setting.

1. Select Load PGIA from the **Calibrate Function** pull-down control. The current calibration values will be displayed.

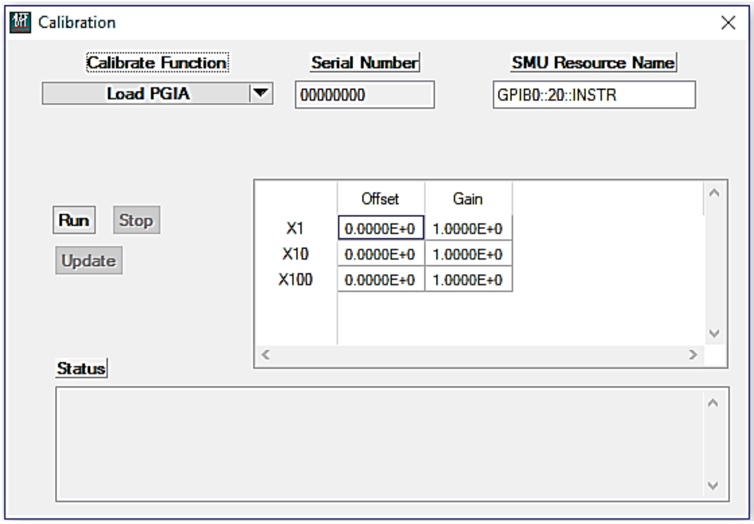


Figure 5-14 Load PGIA Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

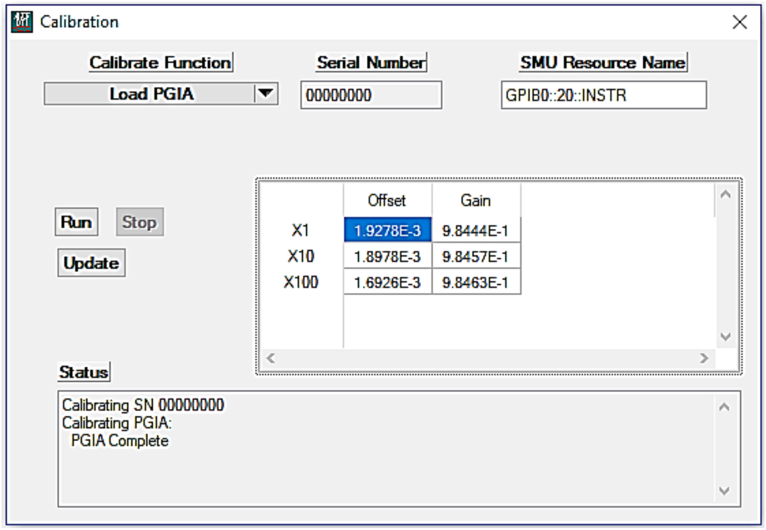


Figure 5-15 Load PGIA Complete

Measure Voltage

The Measure Voltage function calibrates the circuitry path used to measure a voltage for each channel.

For each channel the EXTFORCE is connected to the channel. The SMU is set to output four test points and the voltage is measured by the DTI. Four polynomial coefficients are calculated from the measurement results.

1. Select Measure Voltage from the **Calibrate Function** pull-down control. The current calibration values will be displayed for the selected channel number.

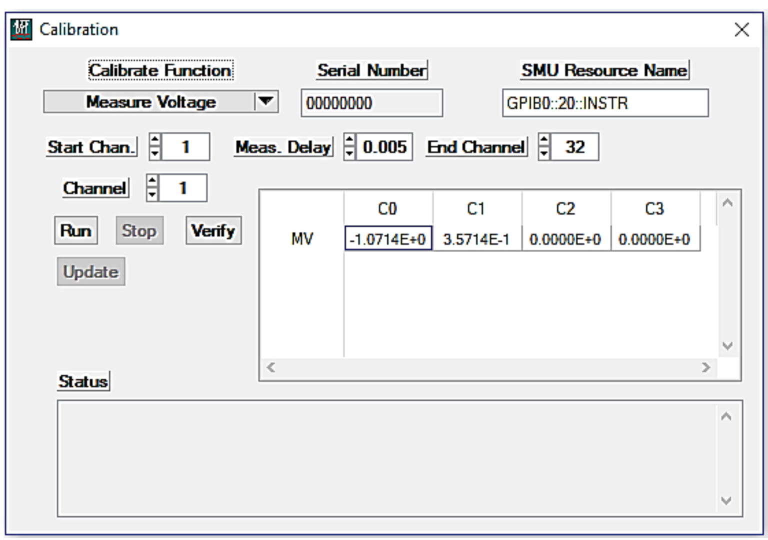


Figure 5-16 Measure Voltage Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

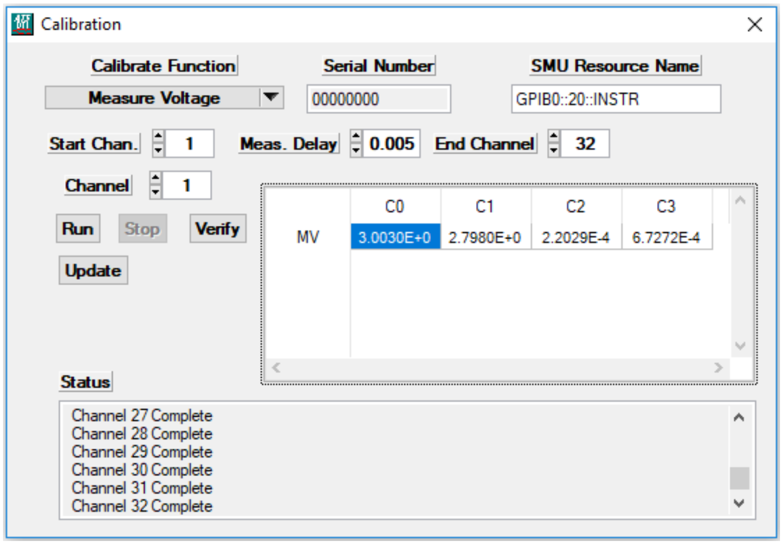


Figure 5-17 Measure Voltage Complete

DAC Overlap

The DACs that generate the driver, comparator and PMU levels are made up of two 8-Bit DACs: Coarse DAC and Fine DAC. The DAC calibration will correct for any step error when transitioning from one coarse segment to the next coarse segment.

There are six DACs per channel:

- F0 – DVL level
- F1 – DVH level
- F2 – VTT/VCOM Source level
- F3 – CVH level
- F4 – CVL level
- F5 – VCOM Sink/FI/FV level

Each DAC has a hardware register that contains the DAC overlap correction number. This calibration calculates the best overlap correction.

1. Select DAC Overlap from the **Calibrate Function** pull-down control. The current calibration values will be displayed for the selected channel number.

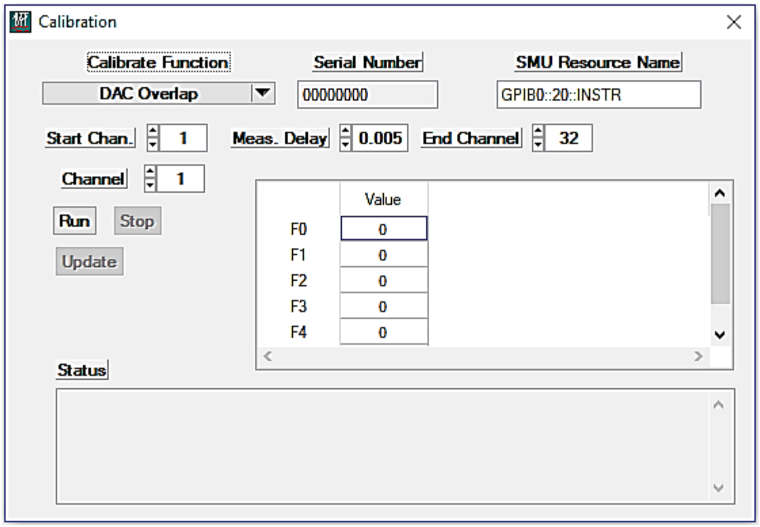


Figure 5-18 DAC Overlap Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

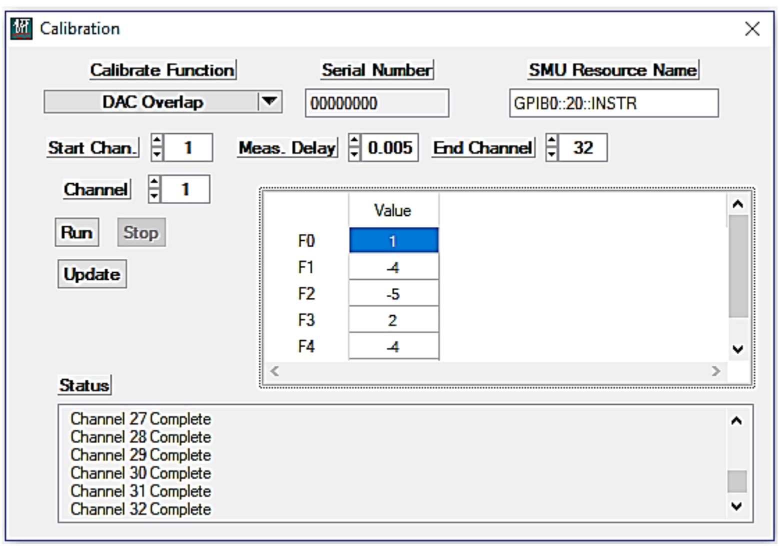


Figure 5-19 DAC Overlap Complete

PMU

Each channel contains a PMU that can be used to perform the following:

- Force Voltage
- Force Current
- Measure Voltage
- Measure Current

When forcing voltage, current clamps are used to prevent over current damage.

When forcing current, voltage clamps are used to prevent over voltage.

The PMU has five current ranges:

- 5µA
- 50µA
- 500µA
- 5mA
- 50mA

Each Channel and Current Range (IR#) needs a unique set of calibration factors:

- FV – Force Voltage level polynomial
- MI-OS – Measure current offset
- CM-AdjP – Common mode measure current positive error adjustment.
- CM-AdjN – Common mode measure current negative error adjustment.
- MI-SRC – Measure current source offset and gain
- MI-SNK – Measure current sink offset and gain
- ICH – Current clamp high level offset and gain
- ICL – Current clamp low level offset and gain
- FI – Force current level polynomial
- VCH – Voltage clamp high level offset and gain
- VCL – Voltage clamp low level offset and gain

All the factors listed above are calculated during calibration.

1. Select PMU from the **Calibrate Function** pull-down control. The current calibration values will be displayed for the selected channel number and current range.

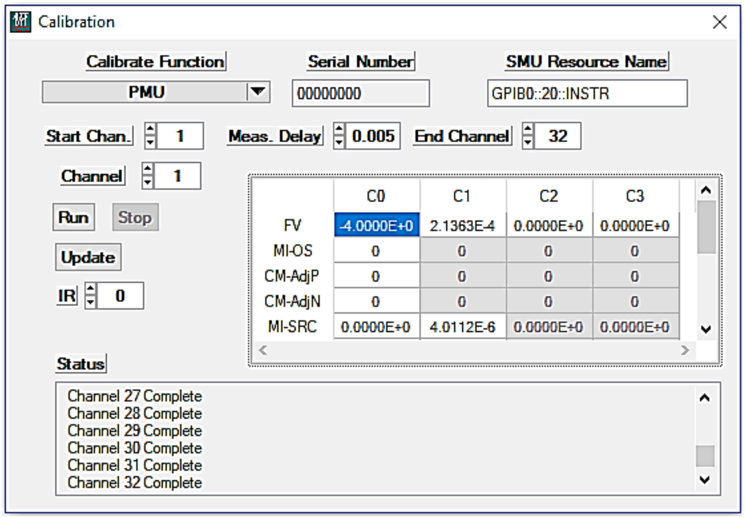


Figure 5-20 PMU Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

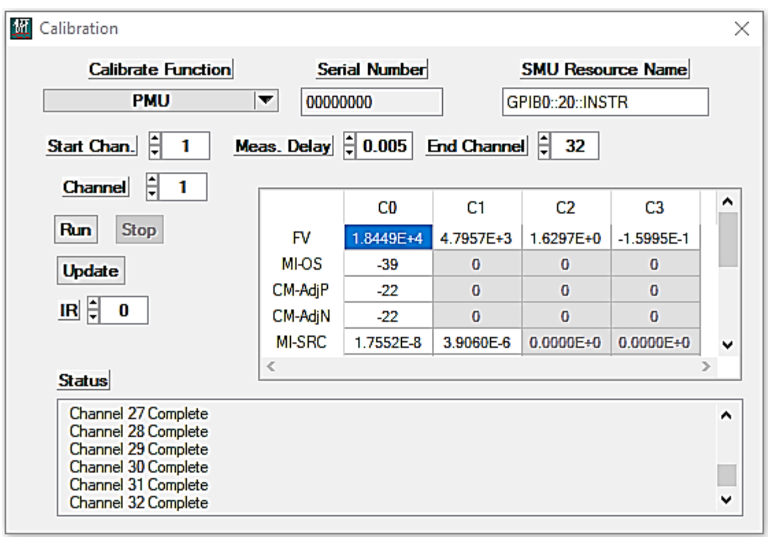


Figure 5-21 PMU Complete

Drive Levels

Each channel is capable of driving three output drive levels when run in dynamic mode:

- DVH – level when channel programmed to output a high.
- DVL – level when channel programmed to output a low.
- VTT – level when channel programmed to go HiZ in Dynamic VTT mode.

The polynomial factors for each level are calculated during calibration.

1. Select Drive Levels from the **Calibrate Function** pull-down control. The current calibration values will be displayed for the selected channel number and current range.

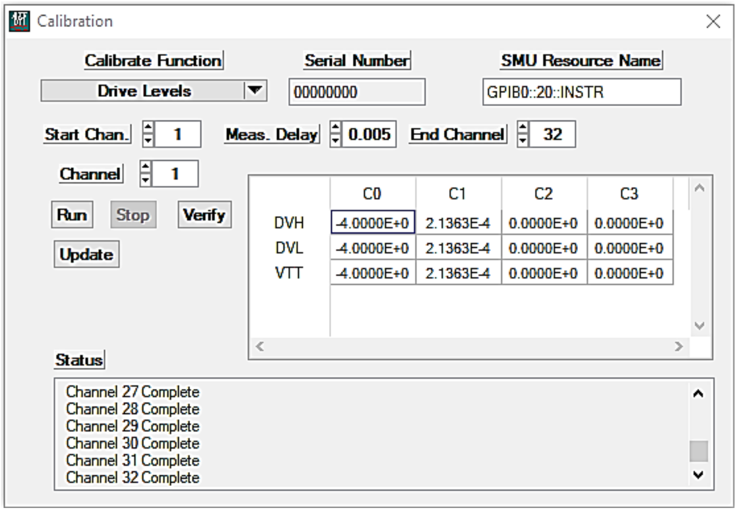


Figure 5-22 Drive Level Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

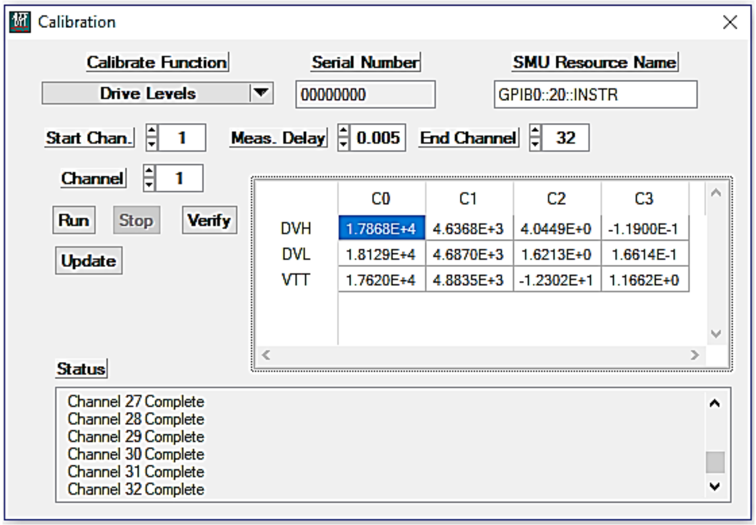


Figure 5-23 Drive Levels Complete

Compare Levels

Each channel has a dual threshold comparator when run in dynamic mode:

- CVH – Comparator level for the good one signal.
- DVL – Comparator level for the good low signal.

The offset and gain for each level is calculated during calibration.

1. Select Compare Levels from the **Calibrate Function** pull-down control. The current calibration values will be displayed for the selected channel number and current range.

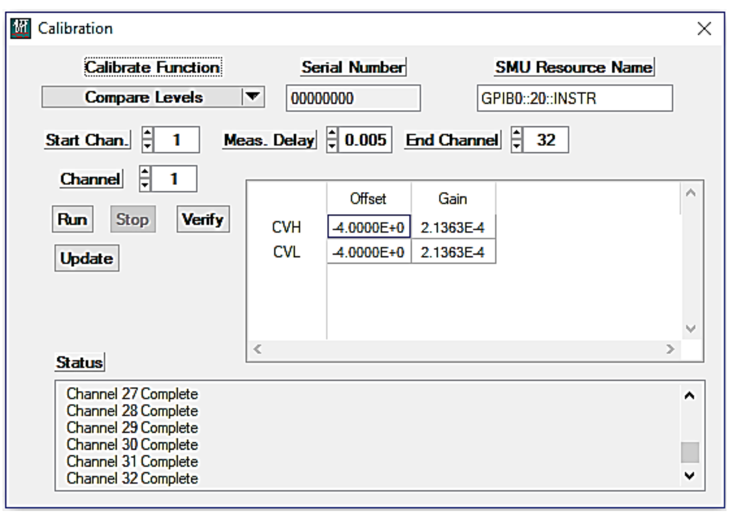


Figure 5-24 Compare Level Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

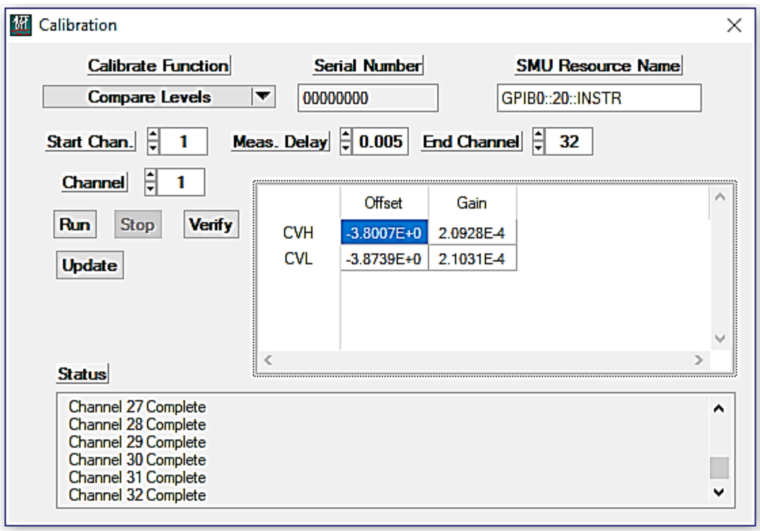


Figure 5-25 Compare Levels Complete

Active Load

The active load calibration includes the following levels:

- VCOMSRC – Source current commutating voltage.
- VCOMSNK – Sink current commutating voltage.
- SRC – Source current.
- SNK – Sink current.

The offset and gain for each level is calculated during calibration.

1. Select Active Load from the **Calibrate Function** pull-down control. The current calibration values will be displayed for the selected channel number and current range.

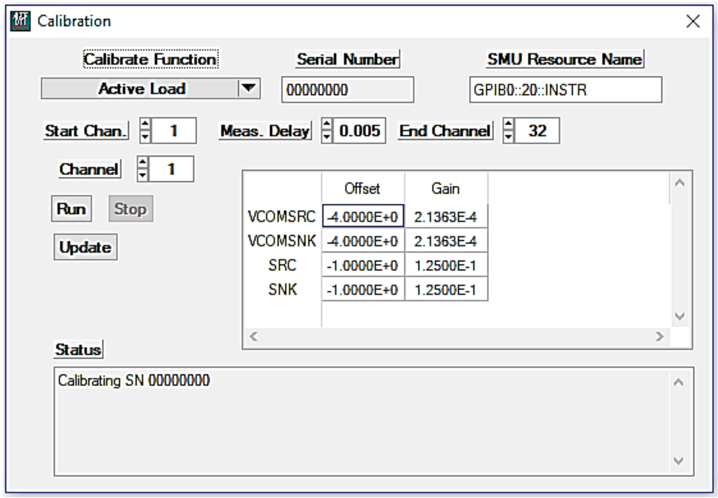


Figure 5-26 Active Load Calibration

2. Enter the resource descriptor of the SMU in the **SMU Resource Name** control.
3. Depress the **Run** command button and respond to the prompts described in Run Calibration Prompts.
4. After responding to the run prompts, the calibration procedure will start. Information messages will be displayed in the Status box. The calibration can be halted by depressing the **Stop** command button. After calibration is completed, the new values will be displayed.

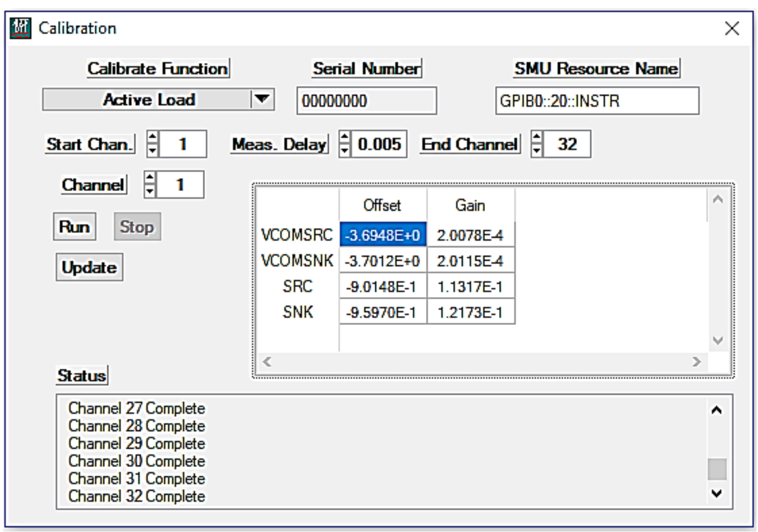


Figure 5-27 Active Load Complete

Delete

Allows the user to delete calibration data stored in non-volatile memory and sets all calibration factors to default.

1. Select Delete from the **Calibrate Function** pull-down control.
2. Depress the **Run** command button. A confirmation prompt will display, select **Yes** to delete the calibration data and **No** to exit.

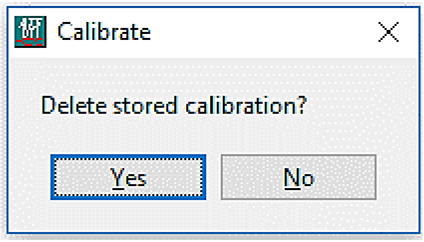


Figure 5-28 Delete Calibration Prompt

Chapter 6
Advanced Topics

This section describes advanced topics of the DTI, giving more details than what were provided in previous chapters. Because references are made to DTS configurations, relevant API calls are also both provided.

The topics covered include:

- DTS Signaling
- Recording Sequence Results
- Pass/Fail Flag Operation
- Jumping
- Pause and Halt
- Sequencer Operation
- PXI Backplane Trigger Bus

DTS Signaling

A Digital Test Suite (DTS) is comprised of two to thirteen DTI modules connected together by an external timing bus (ETB). The DTI modules are configured into one of four types:

- Primary
- Secondary
- Terminating
- Independent

The ETB connects the DTI modules together and routes the signals required to couple the DTI modules into a larger DTS. Table 6-2 lists the ETB signals.

Signal	Connection	Description
Phase 1 - 4	Bussed	The output timing phase signals sourced by the primary DTI to the secondary and terminating DTIs.
Window 1 - 4	Bussed	The input timing window signals sourced by the primary DTI to the secondary and terminating DTIs.
SEQ_CLK	Bussed	Sequencer timing signal signals sourced by the primary DTI to the secondary and terminating DTIs.
SEQ_CLK+D	Bussed	Sequencer timing signal signals sourced by the primary DTI to the secondary and terminating DTIs.
T0-CLK	Bussed	Sequencer timing signal signals sourced by the primary DTI to the secondary and terminating DTIs.
Jump	Bussed	Sequencer timing signal signals sourced by the primary DTI to the secondary and terminating DTIs.
Error 1 – 12	Point to Point	Error signal sourced by the secondary and terminating DTIs to the primary DTI.

Signal	Connection	Description
PV 1 - 12	Point to Point	Pass valid signal sourced by the secondary and terminating DTIs to the primary DTI.

Table 6-1 ETB Signal Description

The PXI backplane trigger bus is used to provide additional DTS signaling between the DTI modules. Table 6-2 summarizes the applicable signals and their usage.

Signal	Description
Halted	Allows coupled Sequencers to have their Pattern Data and Record memories accessible when halted.
DTS Sync	Allows one to detect and create an event that says that that a connected/coupled Sequencer is out of sync with the primary Sequencer
Sequence Reset	Allows a Sequence Reset performed on any coupled Sequencer to reset all of the Sequencers coupled together in a DTS.
Master Reset	Allows a Master Reset performed on any coupled Sequencer to reset all of the Sequencers coupled together in a DTS.
Driver Disable	If programmed to do so on each Sequencer, a channel fault which occurs on the Master or any connected/coupled Sequencer will disable all of the channel drivers.
Static Pulse	Couples the Static Stimulus/Response Pulse from the Master to all connected/coupled sequencers. It is only needed if Static Mode is being used.

Table 6-2 Backplane DTS Signals

From the Configure Module panel, select PXI Triggers for each coupled Sequencer.

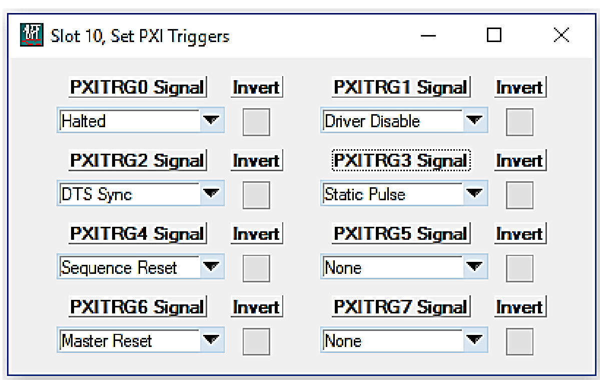


Figure 6-1: Configure PXI Triggers

Relevant API function(s):

- ats6943e_setTtlTriggers

NOTE

Do not program the same trigger signal on sequencers which are not part of the DTS.

Recording Sequence Results

Results from a sequence run captured in three possible places:

1. Record Data
2. Error Counter
3. CRC Memory

This following sections discusses each space and user settings that can be used to control them.

Record Data

The record data consists of the following:

- Record Memory
- Record Index Memory
- Error Address Memory

Record Memory

The record memory is 256K x 32 bit static memory. The contents of the memory can be set per step to indicate the channel level or the channel error where bit 0 is mapped to CH0 and bit 31 is mapped to CH32.

When set to capture channel errors, a one indicates an error and a zero indicates a non-error.

When set to capture channel levels, a one indicates that the channel is higher than the raw record basis and a zero indicates that the channel is lower than the raw record basis.

The record memory can be filled synchronous with the pattern memory (normal) or indexed starting at zero. Indexed addressing is useful when looping or bursting.

See **Step Record Mode** and **Record Type** sections in this chapter for record memory options.

Record Index Memory

When the record type is set to indexed, this memory contains the step number and record memory index for every step and is used re-align the recorded data with the pattern data. The first 1024 step indexes will be captured.

Error Address Memory

If the record memory is set to capture channel errors and a pattern contains one or more channel errors, the step number and pattern address will be written to the error address memory. The first 1024 error address will be captured.

Error Counter

The error counter contains the number of pattern errors in the previous sequence execution.

A pattern error is accumulation of all the channel errors tested at the end of each pattern period

Channel errors are set true at the beginning of each pattern period for every channel that has an expect code programmed. At the terminating capture event, Window Open if Capture Mode set to Open Edge or Window Close is Capture Mode set to Close Edge or Window, the error signal is reset if the channel input matches the expected code.

Errors can be counted for Independent Sequencers or multiple Sequencers which are part of a DTS. Similarly, Errors can be logged into the error address memory for Independent Sequencers or multiple Sequencers which are part of a DTS.

For both of these circumstances, the Errors can be non-qualified Errors or Qualified Errors.

When Non-qualified Errors are chosen all of the Pattern Errors in a Sequence Step are counted if the Step Record mode calls for Errors to be counted.

When Qualified Errors are chosen, only those patterns enabled by BERREN (Burst Error Enable) are counted if the Step Record mode calls for Errors to be counted.

CRC Memory

The CRC memory contains the CRC results for all 32 channels. The CRC memory is initialized at the beginning of a sequence execution.

User Settings

The following sections describes the user settings associated with recording sequence results.

Step Record Mode

Step Record Mode is programmed per sequencer step. On the SFP, it is set on the **Edit>Data Sequencer>Sequence Steps** panel.

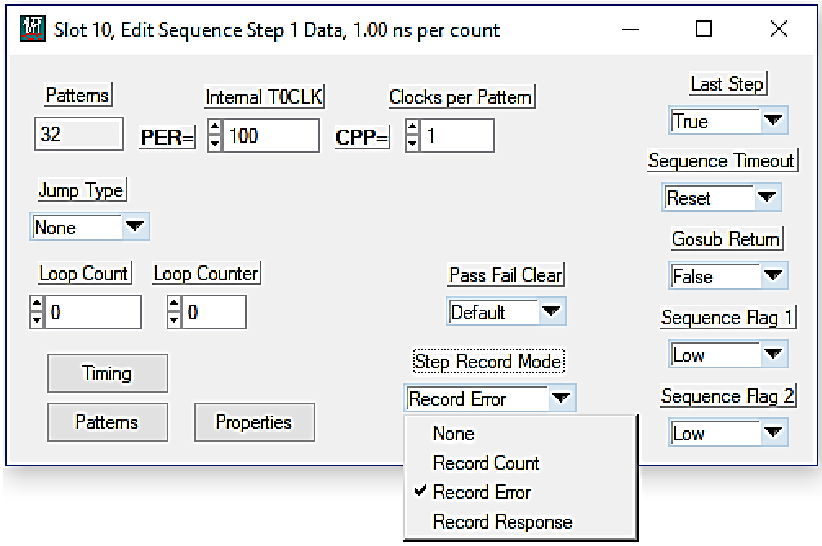


Figure 6-2: Step Record Mode Control

This setting determines what is written in the record data. This selection is made per sequence step. Table 6-3 describes the record memory contents for each selection.

Step Record Mode	Record Memory Action
None	Causes the record, error address and record index memory to be disabled and nothing will be recorded for any pattern in this step.
Record Count	The record, error address and error index memory will be set by the Sequencer Record Mode setting.
Record Errors	Write error results in the record and error index memory. If record type is set to indexed, then update the record index memory.
Record Response	Write response results in the record memory. If record type is set to indexed, then update the record index memory.

Table 6-3 Step Record Mode Effect on Record Memory

Relevant API function(s):

- ats6943e_setSequenceRecordMode

Sequencer Record Mode

The Sequencer Record Mode is set per module. On the SFP, it is set on the **Config>Module** panel.

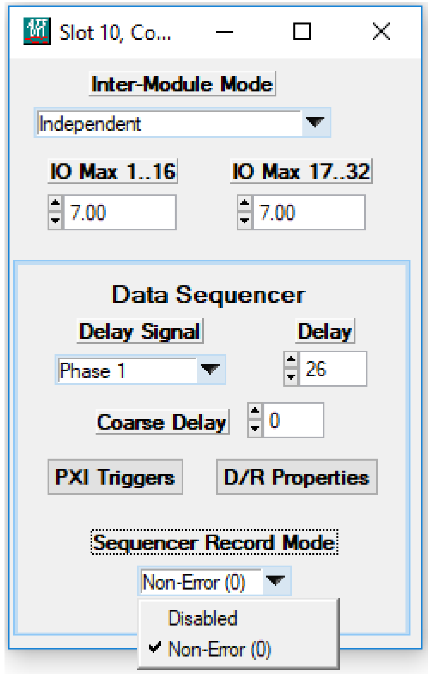


Figure 6-3 Sequencer Record Mode Control

Setting the sequencer record mode to **Disabled** has the same effect as setting the Step Record Mode to None as shown above.

Setting the sequencer record mode to **Non-Error** means that "zeros" will be written into the Record Memory for the Patterns on that Sequence Step, effectively clearing the memory.

Relevant API function(s):

- ats6943e_setSequencerRecordMode

NOTE

If using Indexed Recording, "Disabled" is the better choice to avoid filling up the Record Memory unnecessarily with "zeros."

Record Type

The Record Type is programmed on the **Config>Data Sequencer>Settings** panel.

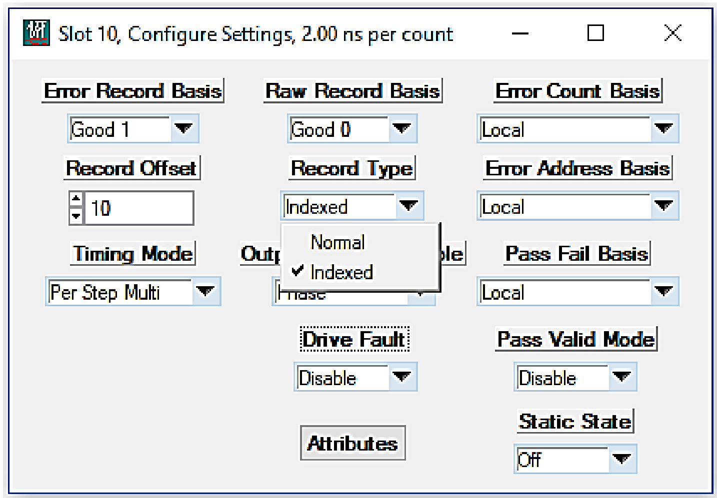


Figure 6-4 Record Type Settings

Setting the record type to **Normal** records into the Record Memory at the same address which corresponds to the pattern data. Thus, when looping a Step or repeating a Step at some later point during the primary sequence execution, the data in the record memory will be over-written.

Setting the record type to **Indexed** recording means that data will be written into the record memory consecutively. The record index memory keeps track of how the data is written into the memory so it can be reconstructed, i.e., which data belongs to each step and/or loop.

Relevant API function(s):

- ats6943e_setRecordParameter

Error Record Basis

The Error Record Basis is set from the **Config>Data Sequencer>Settings** panel.

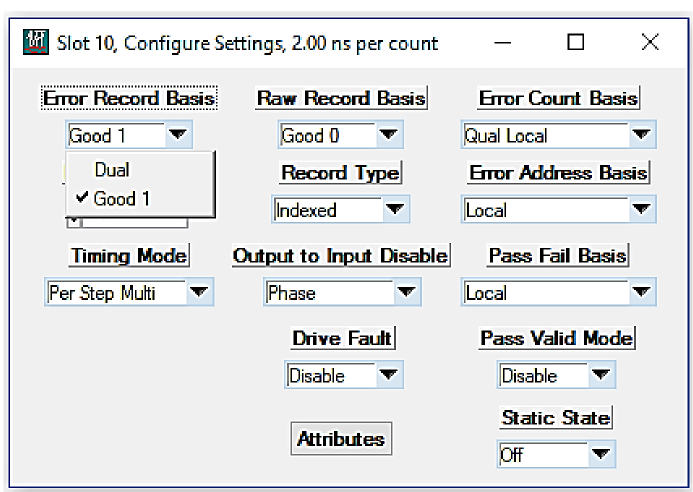


Figure 6-5 Error Record Basis Settings

Setting the error record basis to **Good 1** causes the error signal to be determined by the Good 1 signal only.

Setting the error record basis to **Dual** causes the error signal to be determined by the Good 1 and Good 0 signal.

Expect Code	Comparator Signals		Error Record Basis Results	
	Good 1	Good 0	Good 1	Dual
Valid Low (L)	0	0	Non Error	Error
	0	1	Non Error	Non Error
	1	0	Error	Error
	1	1	Error	Error
Valid Low (H)	0	0	Error	Error
	0	1	Error	Error
	1	0	Non Error	Non Error
	1	1	Non Error	Error
Valid (V)	0	0	Non Error	Error
	0	1		Non Error
	1	0		Non Error
	1	1		Non Error
Between (B)	0	0		Non Error
	0	1		Error
	1	0		Error
	1	1		Error

Table 6-4 Error Record Basis Results

NOTE

Expect valid (V) and expect between (B) codes are only applicable with the Dual error record basis and should not be used with the Good 1 basis.

Relevant API function(s):

- ats6943e_setRecordParameter

Raw Record Basis

The Raw Record Basis is set from the **Config>Data Sequencer>Settings** panel.

This setting is only applicable when the Step Record Mode is set to Record Response.

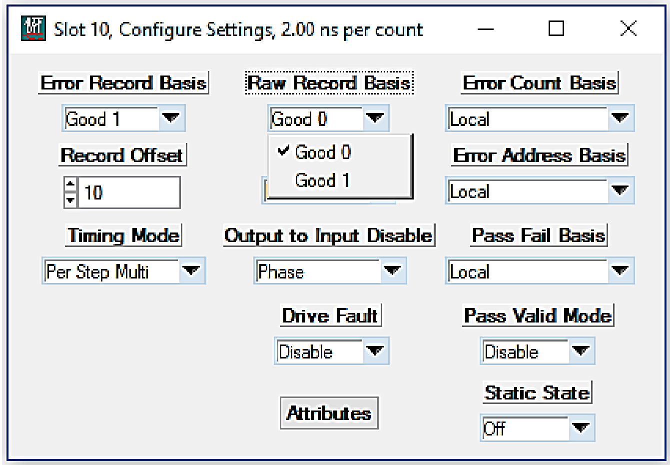


Figure 6-6 Raw Record Basis Settings

Setting the raw record basis to **Good 0** causes the response signal to be determined by the Good 0 signal.

Setting the raw record basis to **Good 1** causes the response signal to be determined by the Good 1 signal.

Relevant API function(s):

- ats6943e_setRecordParameter

Error Count Basis

The Error Count Basis is set from the **Config>Data Sequencer>Settings** panel.

This setting is only applicable when the Step Record Mode is set to Record Errors and sets the source of the error counter.

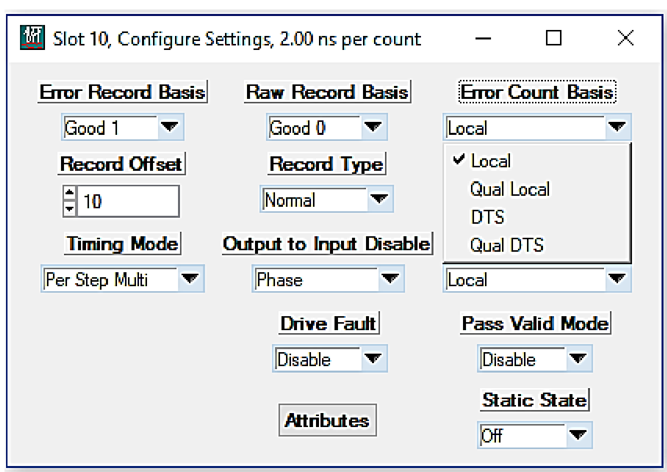


Figure 6-7 Error Count Basis Settings

Error Count Basis	Error Counter Action
Local	The error counter will be incremented by one if the module error signal set true.
Qual Local	The error counter will be incremented by one if the module error signal and the BERREN flag are set true.
DTS	The error counter will be incremented by one if the DTS error signal set true.
Qual DTS	The error counter will be incremented by one if the DTS error signal and the BERREN flag are set true.

Table 6-5 Error Counter Operation with Error Count Basis Settings

Relevant API function(s):

- ats6943e_setErrorParameter

Error Address Basis

The Error Address Basis is set from the **Config>Data Sequencer>Settings** panel.

This setting is only applicable when the Step Record Mode is set to Record Errors and sets the source of the error address memory.

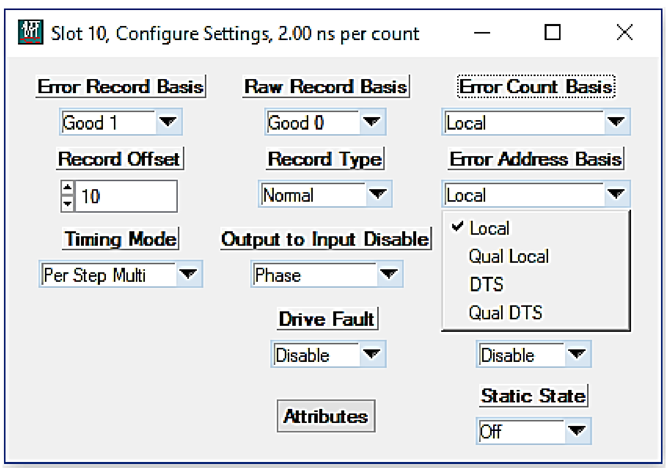


Figure 6-8 Error Address Basis Settings

Error Address Basis	Error Address Memory Action
Local	The error address memory will be updated if the module error signal set true.
Qual Local	The error address memory will be updated if the module error signal and the BERREN are set true.
DTS	The error address memory will be updated if the DTS error signal set true.
Qual DTS	The error address memory will be updated if the DTS error signal and the BERREN are set true.

Table 6-6 Error Address Memory Operation with Error Address Basis Settings

Relevant API function(s):

- ats6943e_setErrorParameter

CRC Type

The CRC Type is set from the Exec>Data Sequencer panel.

This setting sets the CRC algorithm type.

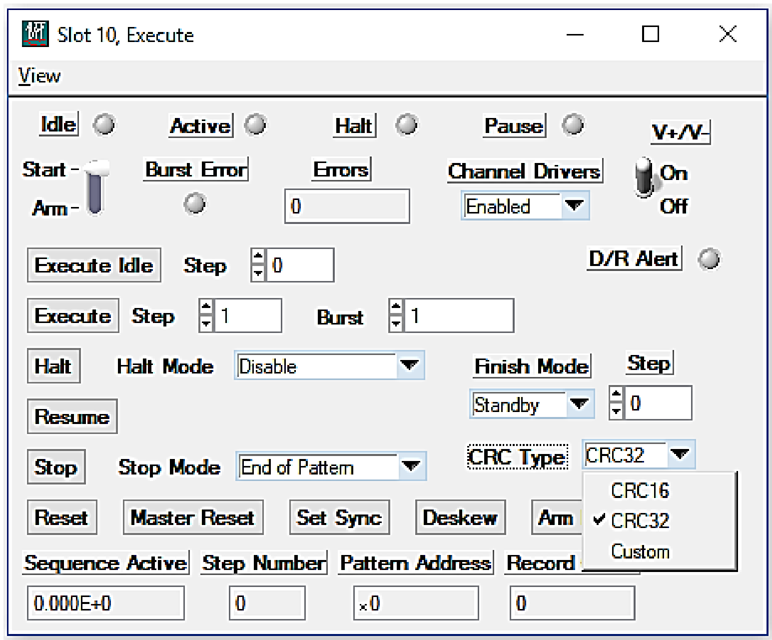


Figure 6-9 CRC Type Settings

CRC Type	CRC Polynomial Feedback
CRC16	This sets the polynomial algorithm to hex 8940, Bit 16, Bit 12, Bit 9, Bit 7 and Bit 0.
CRC32	This sets the polynomial algorithm to hex 82608EDB, Bit 32, Bit 26, Bit 23, Bit 22, Bit 16, Bit 12, Bit 11, Bit 10, Bit 8, Bit 7, Bit 5, Bit 4, Bit 2, Bit 1 and Bit 0.
Custom	The polynomial algorithm is user specified from the **Config>Data Sequencer** panel by depressing the **Attributes** command button.

Table 6-7 CRC Type Polynomial Feedback

Relevant API function(s):

- ats6943e_setSequencerAttribute
- ats6943e_setCrcType

CRC Preload

The CRC Preload is set from the **Config>Data Sequencer>Settings** panel and depressing the Attributes command button.

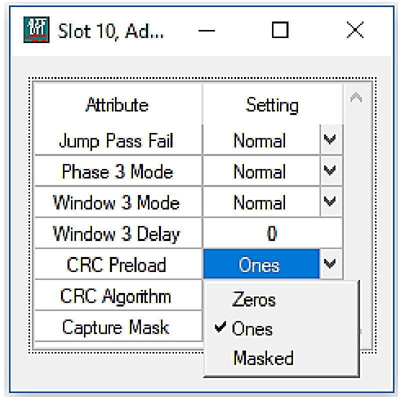


Figure 6-10 CRC Preload Settings

CRC Preload	CRC Register Action
Ones	This loads the channel CRC registers with ones prior to sequence execution.
Zeros	This loads the channel CRC registers with zeros prior to sequence execution.
Masked	This disables pre-loading the channel CRC registers. This is used to accumulate CRCs from multiple sequence executions.

Table 6-8 CRC Preload Settings

Relevant API function(s):

- ats6943e_setSequencerAttribute

CRC Algorithm

The CRC Algorithm is set from the **Config>Data Sequencer>Settings** panel and depressing the **Attributes** command button.

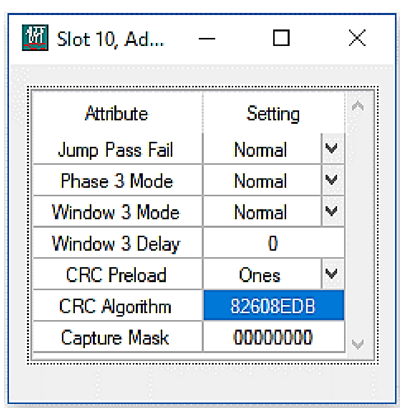


Figure 6-11 CRC Algorithm Settings

The CRC algorithm is specified by entering a polynomial representation. The polynomial representation defines the feedback bits used by the CRC generator.

For example the hex 82608EDB represents the following polynomial equation:

1. Convert the value to binary.
   1000 0010 0110 0000 1000 1110 1101 1011
2. Shift the value left by 1 and add 1
   1-0000 0100 1100 0001 0001 1101 1011 0111 (0x04C11DB7)
3. For every 1 add $x^{bit\ position}$ into the equation. Note $x^1=x;\ x^0=1$
   $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$

By reversing the steps above, we can determine the CRC algorithm value from the equation.

Convert the CRC16-CCITT equation into the DTI polynomial representation:

1. CRC16-CCITT equation.
   $X^{16}+x^{12}+x^{5}+1$
2. Convert to binary number.
   1-0001 0000 0010 0001 (0x1021)
3. Shift value right by 1.
   1000 1000 0001 0000 (0x8810)

Relevant API function(s):

- ats6943e_setSequencerAttribute

BERREN Bit

The BERREN (Burst Error Enable) bit is set in the Pattern Memory. The Pattern Data can be accessed on either the **Edit>Data Sequencer>Patterns** panel or **Edit>Data Sequencer>Sequencer Steps** panel.

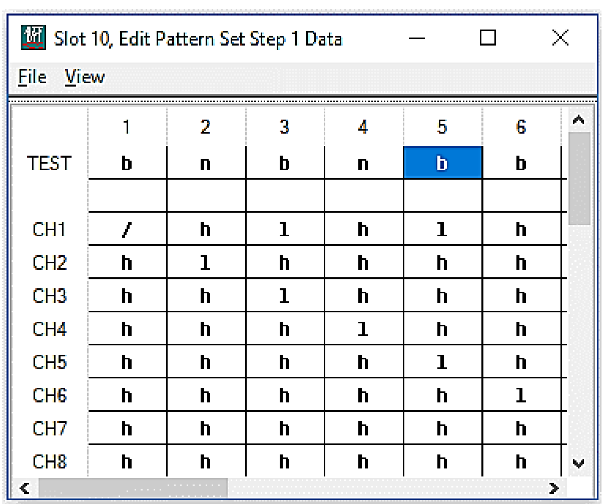


Figure 6-12 Setting the BERREN Bit in the Edit Pattern Set Step Panel

In the TEST row for each pattern (column), a “b” sets BERREN true whereas an “n” sets it false.

Thus, in the example, above, patterns 1, 3, 5 & 6 have BERREN set whereas patterns 2 & 4 do not have BERREN set.

Relevant API function(s):

- ats6943e_setPatternTestEnable

Pass/Fail Flag Operation

The DTI sequencer has extensive capability when it comes to Jumping or Halting on various Pass/Fail conditions. This section describes the pass/fail operation and settings.

The Soft Front Panel (SFP) is used to program the pass/fail settings but API references are provided.

Pass/Fail Basics

The pass/fail flags are set during sequence execution from the channel error signals and are used for jumping and halting.

Pass Fail Basis

Similar to the Error Counter, there is a Basis for the Pass/Fail signals. Pass/Fail signals can be derived from the local sequencer errors or errors from multiple sequencers which are part of a DTS including the local. For both of these settings, the pass/fail signals can be non-qualified or qualified. Setting the pass/fail basis is performed from the **Config>Data Sequencer>Settings** panel.

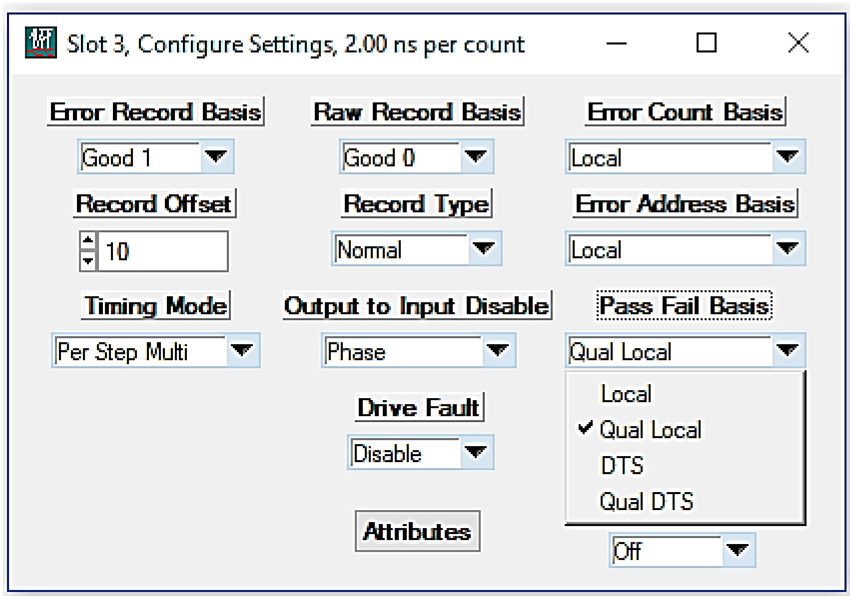


Figure 6-13 Setting the Pass/Fail Basis

Relevant API function(s):

- ats6943e_setPassFailParameter

A qualified Pass Fail Basis, in this case, is based on CONDEN flag (Condition Enable). The CONDEN flag is programmed in the **Edit>Data Sequencer>Pattern Data** panel.

Relevant API function(s):

- ats6943e_setPatternTestEnable

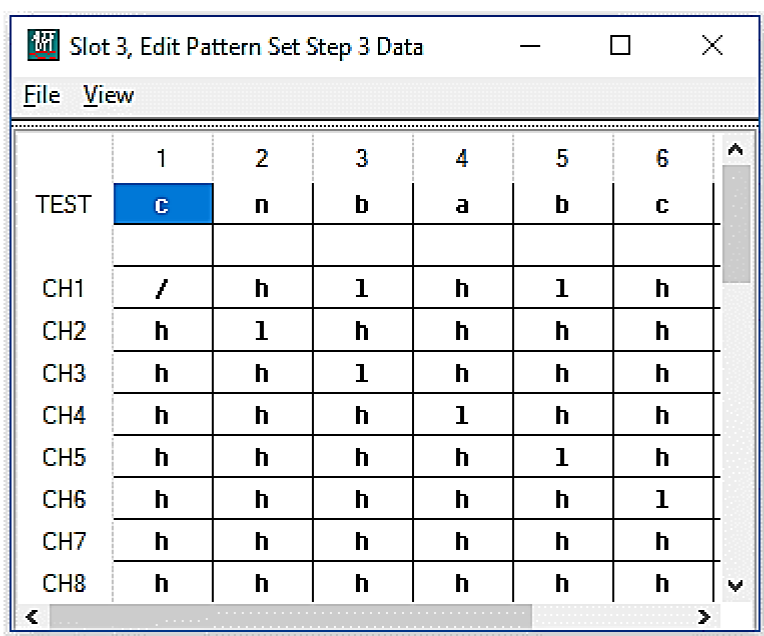


Figure 6-14 Setting the CONDEN Flag

The CONDEN flag is programmed on the TEST row for each pattern (column). “c” indicates that just CONDEN is set, patterns 1 & 6. “b” indicates that just BERREN is set, pattern 3 and 5. “a” indicates that both CONDEN and BERREN are set, pattern 4. “n” indicates that neither CONDEN nor BERREN are set, pattern 2.

There are three sets of pass/fail flags, Pattern, Step and Sequence. Pattern Fail indicates a single pattern did not match the expected value and Pattern Pass is the compliment of Pattern Fail. Step Fail indicates that one or more patterns in the step did not match the expected value and Step Pass is the compliment of Step Fail. Sequence Fail indicates that one or more patterns in any of the sequence steps did not match the expected value and sequence pass is the compliment of Sequence Fail.

Pass Fail Clear

The Step Fail signal is cleared at the end of each sequence step but can be accumulated across consecutive sequence steps. This setting is programmed on the **Edit>Data Sequencer>Sequence Steps** panel.

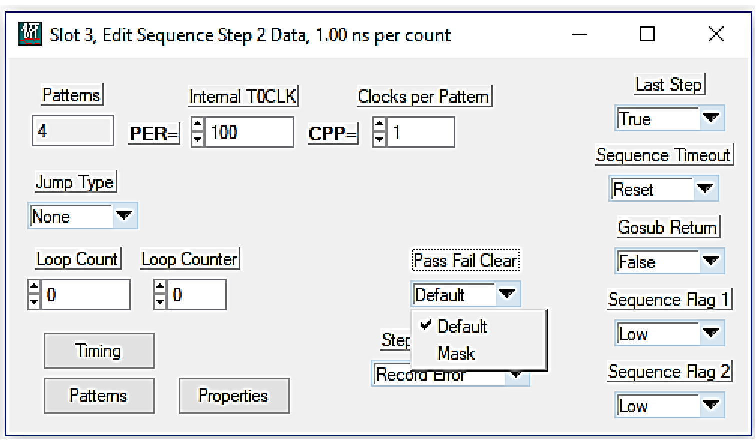


Figure 6-15 Setting Pass Fail Clear

The **Default** setting causes the Step Fail signal to be cleared at the end of each sequence step while the **Mask** setting accumulates the signal to the next sequence step.

Relevant API function(s):

- ats6943e_setSequencePassFailClear

Pass Valid Mode

While the Fail signal indicates that an expect pattern code caused an error, the Pass signal is only the compliment of Fail and indicates that there were no errors or no expect codes. A “Pass Valid Mode” can be enabled that modifies the Pass flag such that there was at least one pattern with an expect condition programmed. If this mode is enabled and there is neither a Pass nor a Fail, it is called “Indeterminate”.

A Valid Pass for a given pattern occurs if there is at least one channel with an

Expect and a Window Capture Mode (an Open Edge, Close Edge or Window) but it does not verify that there is an appropriate window programmed to occur during the period.

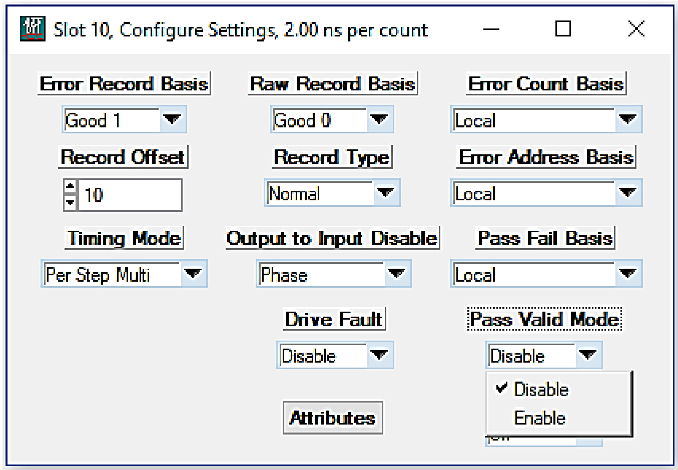


Figure 6-16 Setting Pass Valid Mode

Relevant API function(s):

- ats6943e_setPassFailParameters

Pipelining

The pipeline is FIFO memory used to process the error and pass valid signals to compensate for the comparator delay as well as the UUT latency.

The pipeline may be from 0-16 Patterns deep. A pipeline depth of "0" is called "zero pipeline depth" or "non-pipelined". A pipeline depth of "1-16" is called "non-zero pipeline depth" or "pipelined".

The pipeline depth needs to be set in the same in the Primary and all coupled sequencers and is set on the **Edit>Data Sequencer>Sequence Parameters** panel.

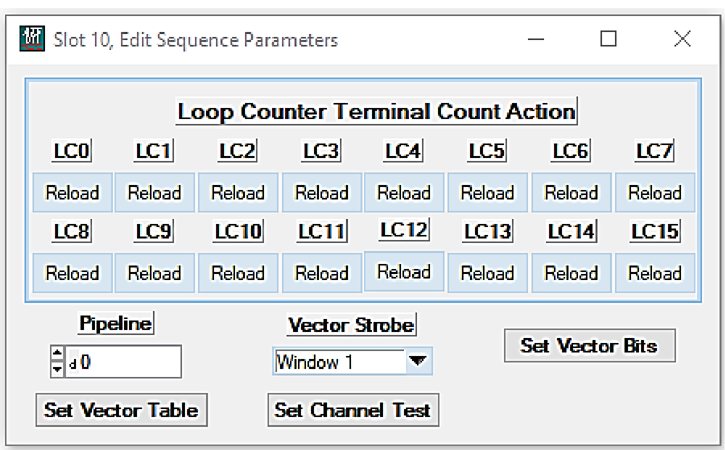


Figure 6-17 Setting the Pipeline Depth

Relevant API function(s):

- ats6943e_setConditionPipeline

Zero Pipeline Operation

In this mode the Error signal is sourced by Raw Error which comes directly from the channel-in logic. An Error is generated at the beginning of each pattern until the capture event (Window Open or Close) at which time it will be the actual Error/Non-Error status. A capture pulse (also called jump strobe) samples the Error signal to generate the pass/fail flags. If the capture event occurs after the capture pulse then the result will always be a fail. The following equations can be used to calculate the capture event maximum value and capture pulse location during the pattern period:

CEmax = Period(ns) – RecordOffset(ns) – 52ns

CP = Period(ns) – (MCLKperiod(ns)) * 13 + 17ns

Example 1 MCLK = 500MHz, Record Offset = 10, Period = 100 T0CLK:

MCLKperiod(ns) = 2

Period(ns) = 100

RecordOffset(ns) = 10 * 2 (20)

CEmax = 100 – 20 – 52 (28ns)

CP = 100 – 2 * 13 + 17 (57ns)

Example 2 MCLK = 250MHz, Record Offset = 5, Period = 75 T0CLK:

MCLKperiod(ns) = 4

Period(ns) = 150

RecordOffset(ns) = 5 * 4 (20)

CEmax = 150 – 20 – 52 (78ns)

CP = 150 – 4 * 13 + 17 (81ns)

Table 6-18 illustrates the zero pipeline timing of the pass/fail flags with respect to the capture event, error signal and capture pulse. This is a two-step sequence with 4 patterns in step 1 and 2 patterns in step 2. An Error is generated in step 1, patterns 1 and 3.

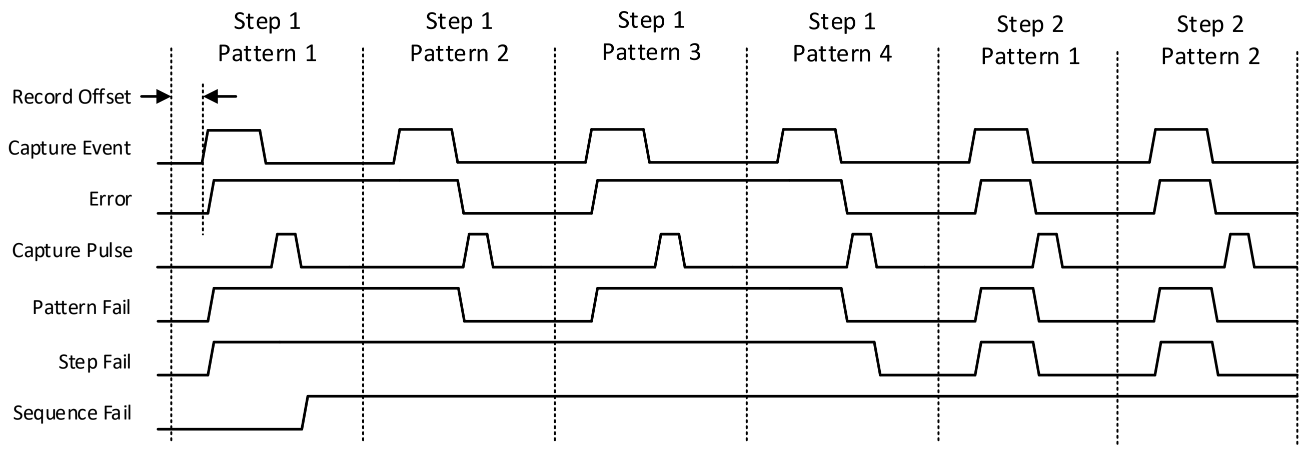


Figure 6-18 Zero Pipeline Timing Diagram

The Capture Event and Error signals are delayed by the record offset. The Pattern, Step and Sequence Pass signals are compliments of the associated Fail signals. The Step Fail accumulates until after the capture pulse of the last pattern in the step. The Sequence Fail accumulates until the next sequence run or burst loop.

Non Zero Pipeline Operation

In this mode the Error is captured at the end of the Pattern period and then propagated as a pulse and output at the beginning of the next pattern. This pulse is inserted, along with the pass valid flag, into the pipeline. The output of the pipeline is the Fail signal and is offset by the depth of the pipeline, i.e., an error in pattern 1 with a pipeline of four will be valid during pattern 5.

There is a minimum pipeline depth required depending on programmed settings. The following equations are used to calculate the minimum depth.

CD = IMdelay + RecordOffset + RespDelay + Period + 11MCLK + 21

Minimum Depth = CD / Period rounded up

Where:

CD — Capture Delay Is the total time from a beginning of the first pattern to when the data can be captured for Jumping or Halting on Pass/Fail.

IMdelay — System delay based on the inter-module mode.

Mode	IMdelay
Independent	14ns
DTS	1ns/DTI + 21ns

RecordOffset — Record Offset setting times the master clock period (ns)

RespDelay — Delay for the Error signal based on inter-module mode.

Mode	RespDelay
Independent	28ns
DTS	2-6 DRMs = 28ns 8 DRMs = 30ns 10 DRMs =34ns 12 DRMs = 36ns

Period	Pattern period in nano seconds.
11MCLK	11 master clock periods(ns)

Example 1 MCLK = 500MHz, Record Offset = 10, Period = 100ns, Independent:

IMdelay = 14

RecordOffset(ns) = 10 * 2 (20)

Period = 100

RespDelay = 28

11MCLK = 22

CD = 14 + 20 + 28 + 100 + 22 + 21 (205)

Minimum Depth = 205 / 100 rounded up (3)

Example 2 MCLK = 500MHz, Record Offset = 16, Period = 40ns, DTS 6 DTI:

IMdelay = 21 + 6 (27)

RecordOffset(ns) = 16 * 2 (32)

Period = 40

RespDelay = 28

11MCLK = 22

CD = 27 + 32 + 28 + 40 + 22 + 21 (177)

Minimum Depth = 170 / 40 rounded up (5)

Table 6-18 illustrates the non-zero pipeline timing of the pass/fail flags with respect to the capture event, error signal and capture pulse. This is a two-step sequence with 4 patterns in step 1 and 2 patterns in step 2. An Error is generated in step 1, patterns 1 and 3. The pipeline depth is set to 2.

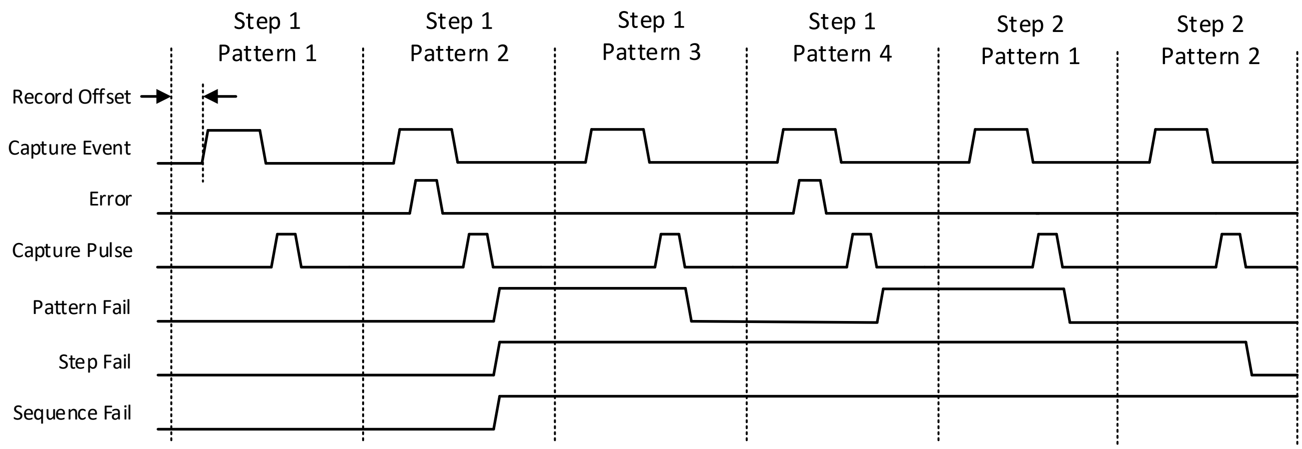


Figure 6-19 Non Zero Pipeline Timing Diagram

The Capture Event is delayed by the record offset. The Error signal is delayed by one pattern and converted to a pulse. The Pattern, Step and Sequence Pass signals are compliments of the associated Fail signals. The Step Fail accumulates until after the capture pulse of the last pattern in the step but is also delayed by the depth of the pipeline. The Sequence Fail accumulates until the next sequence run or burst loop.

The Step Pass/Fail accumulator can be disabled causing the Step Fail signal to operate similar to the Pattern Fail signal. This is programmed on the **Config>Data Sequencer>Setting** panel by clicking **Attributes**.

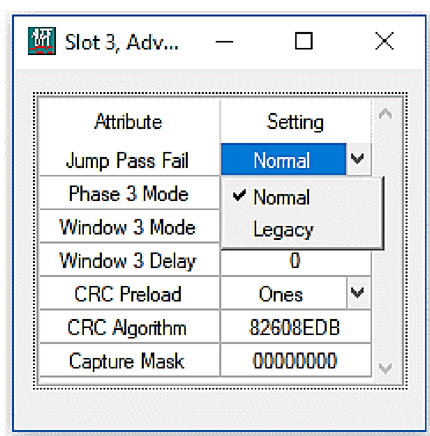


Figure 6-20 Setting the Jump Pass Fail Setting

In **Normal** mode, the Error signal (and Pass Valid, if used) are delayed by the pipeline, these signals will not be aligned with the Jump Test made at the end of an individual Sequence Step (or at the end of a Primary Sequence). The last N patterns before the end of the Sequence Step (or primary sequence) will not be included in the accumulated Pass/Fail decision. The last N patterns of the previous Sequence Step will be included. If these are not to be included in the accumulated Pass/Fail Jumping decision, then they need to be preconditioned. The easiest way to not produce any Errors (or Indeterminates) is to employ a Qualified Pass/Fail basis and disable CONDEN for these N Patterns. Using this method, Errors can still be: Recorded, Counted or Logged into the EAM if desired.

The **Legacy** mode is typically used when one is looping a single Pattern, looking for a Pass or Fail. Jump on NOT Pass will fall through on a Pass and Jump on NOT Fail will fall through on a Fail. In some applications, this may be known as PATC WAIT. This mode requires that the Pipeline be "preconditioned."

Relevant API function(s):

- ats6943e_setSequencerAttribute

Pipeline Preconditioning

This section discusses the methods for preconditions the pipeline.

Jump on NOT Fail (PASS)

We want to "precondition" the pipeline with NOT Fail (PASS).

There are two options for clearing a pipeline of depth "N" to NOT Fail (not generate an Error):

1. If the Jump Basis is <u>not</u> qualified, use a sequence step with a jump to self for a count of "N". For the one Pattern in this step, have an expect condition which is known to NOT Fail (not generate an Error).
2. If the Jump Basis <u>is</u> qualified, use a sequence step with a jump to self for a count of "N" and set CONDEN <u>low</u> (e.g., "b" or "n") for the one pattern in this step (this will fill the pipe with NOT Fail).

Note: this case does not require the Pass Valid Mode to be used. But it may be used and will have no effect.

Jump on NOT Pass (FAIL)

We want to "precondition" the pipeline with NOT Pass (FAIL).

There are two options for clearing a pipeline of depth "N" to NOT Pass (generate an error):

1. If the Jump Basis is <u>not</u> qualified, use a sequence step with a jump to self for a count of "N". For the one Pattern in this step, have an expect condition which is known to Fail (generate an Error).
2. If the Jump Basis <u>is</u> qualified, use a sequence step with a jump to self for a count of "N" and set CONDEN high (e.g., "c" or "a") for the one pattern in this step. For the one Pattern in this step, have an expect condition which is known to Fail (generate an Error).

To generate an Error, there must be at least one channel which has an "expect" which is the complement of the level that is driving that channel. One can, of course, drive one channel and expect the complement if that won't adversely affect the UUT (e.g. it's an unused channel).

Since we're falling through on a Pass, do we want it to be a Valid Pass? As described above, a "Valid Pass" is one where there were no channel Errors but it also says that there was at least one channel with an expect condition. If this additional "qualification" of a Pass is important, then the Pass Valid Mode also needs to be enabled.

Jumping

Each sequence step specifies the following jump settings:

- Jump Type
- Jump Condition
- Jump Sequence Address (JSA)
- Vector Jump Flag

The **Jump Type** is programmed from the **Edit>Data Sequencer>Sequence Steps** panel.

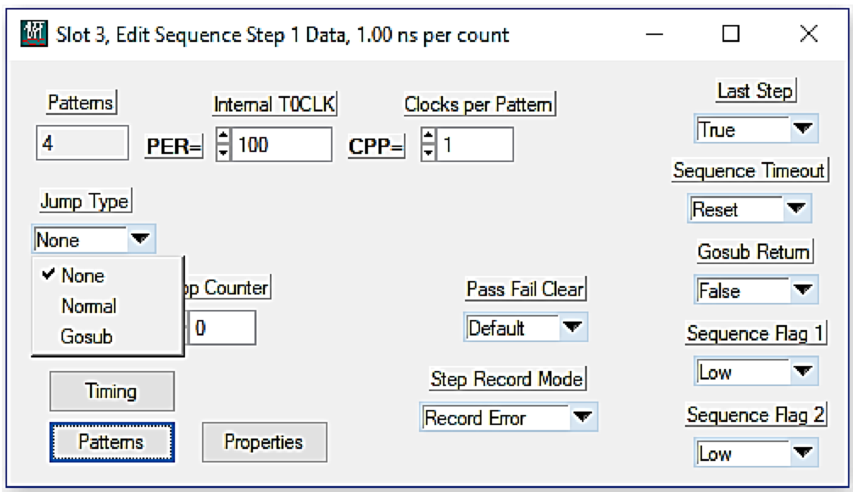


Figure 6-21 Setting the Sequence Step Jump Type

The **None** setting disables jumps and sequence flow will continue with the next sequential step number if **Last Step** is false.

The **Normal** setting tests the jump condition and if true will set the next step to either the specified **Jump Step** or the vector table step if **Vector Jump** is set true. If the jump condition is false then step execution will continue from the next sequential step number if **Last Step** is false.

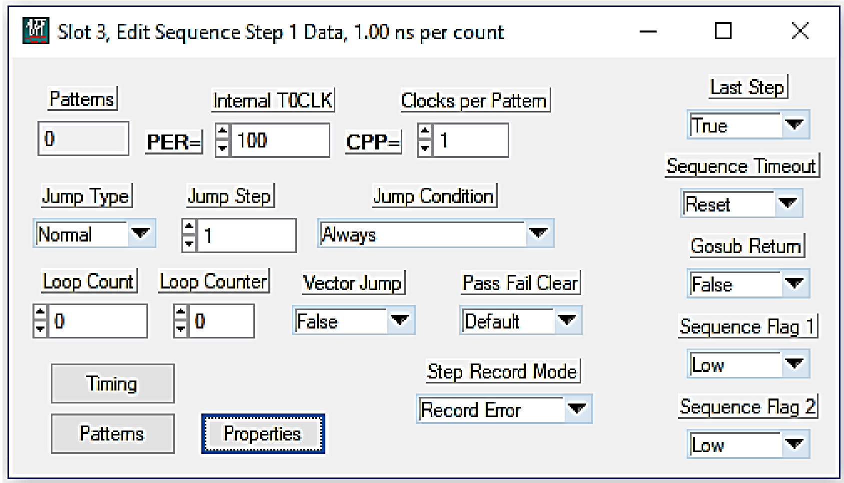


Figure 6-22 Sequence Step Normal and Jump Settings

The **Gosub** setting tests the jump condition and if true will set the next step as the Return Step Address (RSA) and then branch to either the specified **Jump Step** or the vector table step if **Vector Jump** is set true. When the step with the "Gosub Return" flag set has completed, the next step executed will be the RSA. If the jump condition is false then step execution will continue from the next sequential step number if **Last Step** is false. Subroutines cannot be nested. Subroutines may consist of multiple Sequence Steps which contain Loops and/or Jumps.

Relevant API function(s):

- ats6943e_setSequenceJump

The **Jump Condition** specifies a qualifier that must be true for the jump to execute and is set from the **Edit>Data Sequencer>Sequence Steps** panel.

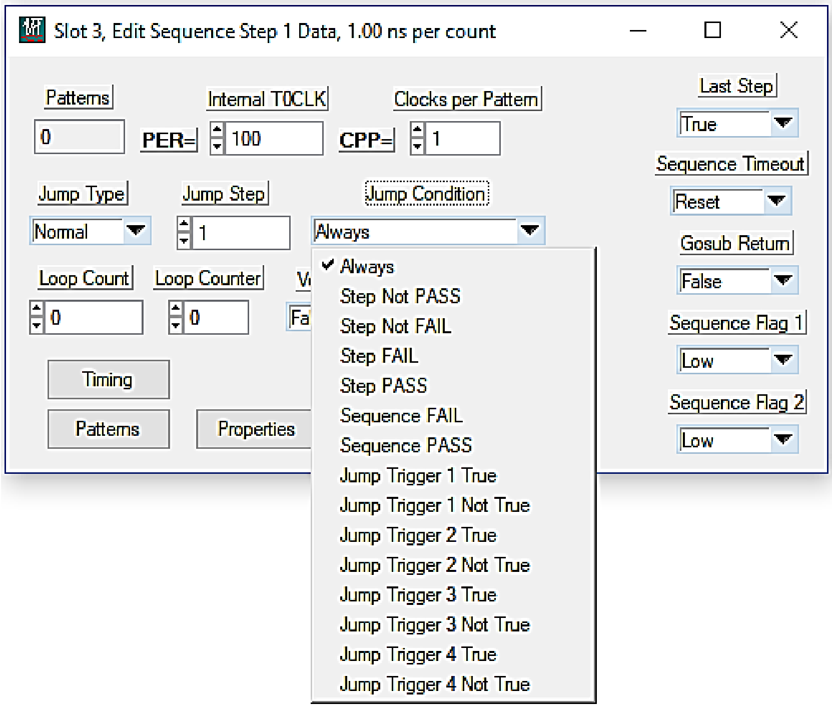


Figure 6-23 Setting the Jump Condition

The **Always** setting forces an unconditional jump at the end of the sequence step.

The **Step NOT PASS** setting enables a jump if the Step Pass flag is false.

The **Step NOT FAIL** setting enables a jump if the Step Fail flag is false.

The **Step FAIL** setting enables a jump if the Step Fail flag is true.

The **Step PASS** setting enables a jump if the Step Pass flag is true.

The **Sequence FAIL** setting enables a jump if the Sequence Fail flag is true.

The **Sequence PASS** setting enables a jump if the Sequence Pass flag is true.

The **Jump Trigger n True** setting enables a jump if the specified jump trigger signal is true.

The **Jump Trigger n False** setting enables a jump if the specified jump trigger signal is false.

Relevant API function(s):

- ats6943e_setSequenceJump

If the jump condition is set to “Always” it is called an unconditional jump otherwise it is called a conditional jump.

All Jumps are to the Jump Step unless the Vector Jump Flag is true in which case the Jump will be to the step address programmed in the Vector Jump Table and indexed by the Vector Bits.

All jumps are executed at the end of a sequence step only if the jump condition is true and the counted loop is not zero if active.

If the sequence step is designated as the last step and has a jump to a subroutine, then upon returning from the subroutine, execution will proceed to the Finishing Sequence.

Jump has priority over gosub return.

Gosub return has priority over last step.

Unlike for the Counting and Logging of Errors, Jumping is not based on the Step Record Mode. The same action is taken for all Step Record Modes.

Pause and Halt

Pause and Halt are used to suspend sequence execution.

A “Halt” disables the System and Pattern Clocks at the end of the Pattern cycle after all Phases and Windows complete their action also known as system clutch in legacy systems.

A “Pause” disables the System and Pattern Clocks and freezes the Phases and Windows also known as pattern clutch in legacy systems.

A “Resume” de-asserts a Pause or Halt and allows the normal operation to continue.

Pause Operation

A Pause can be used to:

- Pause the data output when doing a handshake.
- Pause on a pattern at a Phase edge or with an external signal.
- Insert a fixed wait time.
- An external Resume can be used as a handshake resume.

A Pause operation is defined within a Sequence Step and can thus be programmed to occur only at particular times during the Sequence.

The **Pause Signal** is programmed in the **Edit>Data Sequencer>Sequence Steps Properties** panel.

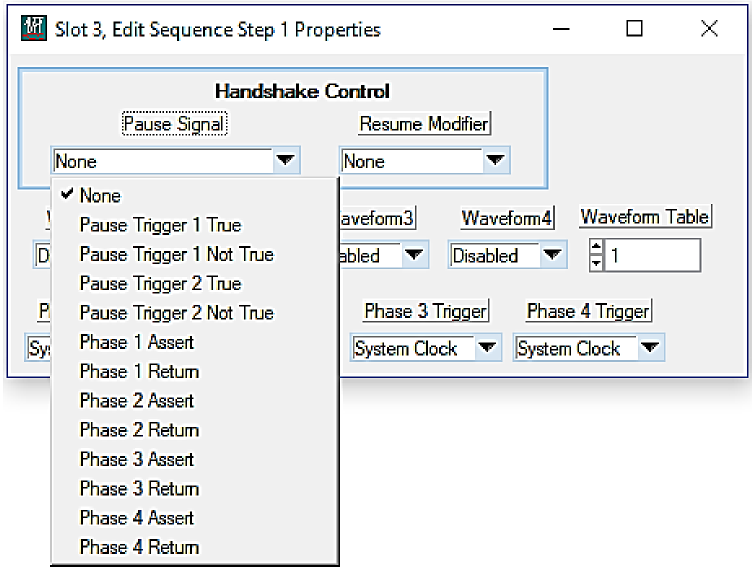


Figure 6-24 Selecting the Pause Signal

The **None** selection disables Pauses for this sequence step.

Pause Trigger n selects the signal assigned to the specified pause trigger to initiate the pause and the signal assigned to the **Pause Trigger n Resume** to resume the pause. The **True** and **Not True** level of the trigger can be selected. For example if the pause trigger was set to AUX1 Low Level, then the True selection would be AUX1 Low and the Not True selection would be AUX1 High. Refer to the **Configure Triggers** section in chapter 4 for Pause/Resume Trigger settings and options.

Phase n Assert/Return selects the specified phase signal to initiate the pause and the signal assigned to the **Phase n Resume** to resume the pause. Refer to the Configure Triggers section for Phase Resume Trigger settings and options.

The **Resume Modifier** is programmed in the **Edit>Data Sequencer>Sequence Steps Properties** panel.

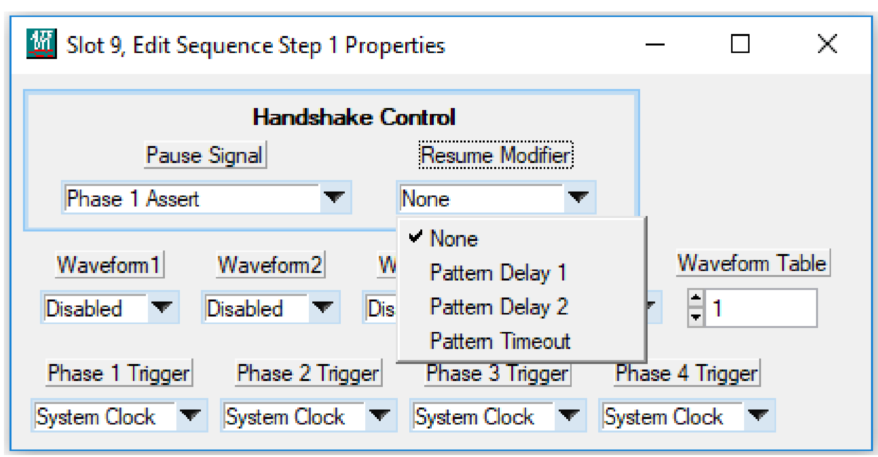


Figure 6-25 Selecting the Resume Modifier

The **None** selection disables Resume Modifier for this sequence step.

Pattern Delay n forces a resume after the specified timer has exhausted if the selected resume signal has not occurred. Refer to the **Pattern Delay 1-2** section in chapter 4 for timer settings and options.

Pattern Timeout forces a resume after the pattern timeout timer has exhausted if the selected resume signal has not occurred. Refer to the **Set Sequence Timer**

Pattern Timeout section in chapter 4 for timer settings and options.

Pause Examples

The following lists common examples using Pauses.

Handshake — Pause on Phase 4 falling edge (right after the output data is formatted and before the input data is to be captured) of Sequence Step 3 (a one pattern Sequence Step) and Resume on the Rising Edge of Aux. 5.

- In Sequence Step 3, set the Pause Signal: Phase 4 Return
- Set Phase 4 Resume Trigger Source: AUX5
- Set Phase 4 Resume Trigger Test Condition: Rising Edge

Pattern Clutch — Simulate a pattern clutch function using the Aux. 6 input (an active high clutch).

- For all Sequence Steps within the Sequence, set the Pause Signal: Pause Trigger 1 True
- Set Pause Trigger 1 Source: AUX6
- Set Pause Trigger 1 Test Condition: High Level

Note: a “high” on Aux. 6 pauses and a “low” resumes (a Resume need not be programmed in this case.)

Timed Delay — Add a delay to a pattern

- Isolate the Pattern in one Sequence Step.
- In this Sequence Step, set the Pause Signal: Phase 4 Assert
- Set Phase 4 Resume Trigger Source: None
- Set Pattern Delay 1 timer: 1s.
- In this sequence step, set the Resume Modifier: Pattern Delay 1.

Pause Notes

Since a Pattern can have multiple Phases (using a Phase Trigger Type set to System Clock) and a CCP>1, multiple handshakes can be performed within a pattern.

Since a Waveform can replace Phases 3 & 4, multiple, irregularly spaced Handshakes can be programmed within a Pattern.

The Resume signals edge detection logic is automatically cleared when not paused.

If the Resume signal is already satisfied, the Pause will not occur.

When paused, the CPU cannot access the Pattern, Record or Probe memories.

A pause based on a level can only be cleared by removing the level causing the pause. Manual Resume and Resume Modifiers are disabled.

The Phase Pause edge must occur at least 16ns before the end of the period (using a 500MHz Master Clock).

It is possible to record the correct results even when pausing. To do so, the Window decision edge must occur no later than 6ns after the Pause decision edge. Aux. outputs may be used to examine the timing relationship of the active Windows with respect to the Phase edge (or external signal) used to trigger a pause.

The Window decision edge in pattern “n” must occur before any pause in pattern “n+1” by at least the amount of record offset in ns, in order to capture results correctly.

The delay from a Phase or TTLTRG Pause to an actual pause is ~6-7ns. Likewise, the delay from a TTLTRG Resume is ~6-7ns.

Halt Operation

Halting an execution sequence can be triggered internally or externally.

Internal Halt

Internal Halts can be used to:

- Halt on Pass/Fail
- Halt on a pattern using a Sync pulse or external signal
- Establish a breakpoint
- Do single-stepping

The Internal Halt Mode is Programmed on the **Execute>Data Sequencer** panel.

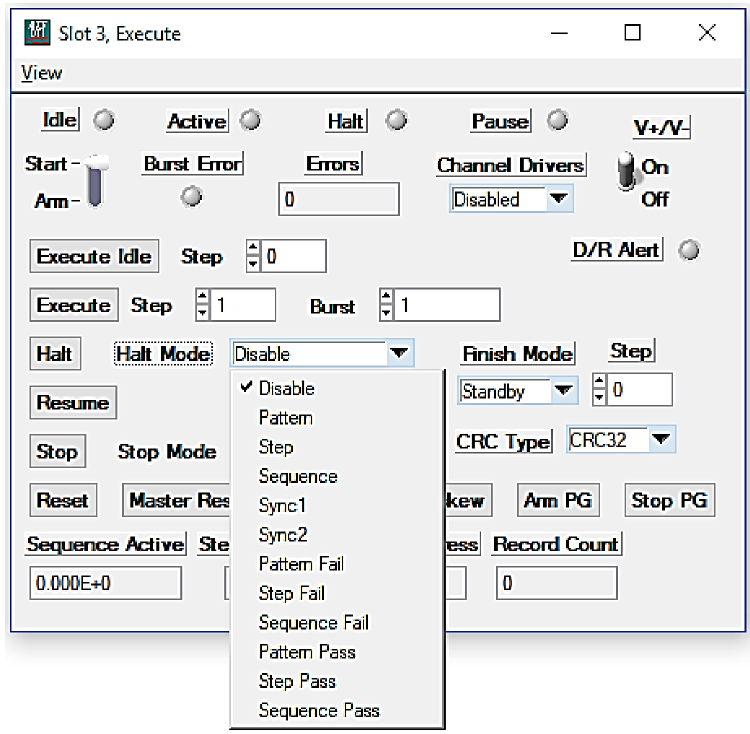


Figure 6-26 Setting the Halt Mode

Relevant API function(s):

- ats6943e_setHaltMode

The **Disabled** setting disables the internal halt mode.

The first five are typically used for single stepping:

- Pattern
- Step
- Sequence
- Sync 1
- Sync 2

The Sync selections are actually Pulses programmed on the Execute Panel by clicking “Set Sync”.

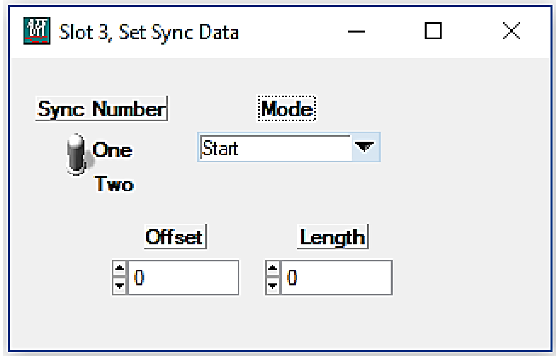


Figure 6-27 Setting the Sync Pulse

Relevant API function(s):

- ats6943e_setSyncEvent
- ats6943e_setSyncParameters

The Sync Pulse, can be set to start from the beginning of the sequence, a specific sequence step just once or a specific sequence step continuous. An Offset and a Length are specified in patterns.

If the Length of the Sync Pulse is N, then single pattern stepping will continue until N is exhausted.

To use these first five perform the following on the Exec>Data Sequencer panel:

1. Select the desired Halt Mode.
2. Depress the “Halt” command button.
3. Depress the “Execute” command button.
4. Each time “Halt” is subsequently clicked the Halt will re-occur on the next Pattern, Step, etc.

One can change the Halt Mode between clicks of “Halt”. For example, one may initially have a Sync Pulse halt and then change to pattern to single step one pattern at a time.

When finished doing single stepping, click Resume.

Relevant API function(s):

- ats6943e_resumeSequence

The last six types of Internal Halt Modes cover various types of Pass/Fail conditions. In these modes, set the desired condition and then click “Execute”.

The relevant VXI*plug&play* API function is:

- ats6943e_executeSequence

Do not click “Halt” before “Execute”. Click Resume when you want to proceed to the next conditional Halt, if more are expected. As before, the Halt Mode may be changed between “Resumes”. To finish the Primary Sequence without any further Halts, change the Halt Mode to Disable.

The “Pass Fail Basis” applies to conditional Halting. Thus, one can halt on all Pattern Pass or Fail conditions or only those qualified with CONDEN.

When pipelined, Halts occur the depth of the pipeline later.

In non-pipelined mode, there is a maximum data rate where that one can do conditional Halting without corrupting the counting and/or logging of Errors, see Zero Pipeline Operation.microsoft

External Halt

An external halt is initiated from the Halt Trigger.

Refer to the **Configure Triggers** section in chapter 4 for Halt Trigger settings and options.

An external Halt can only halt on a Pattern.

If the Halt Trigger test condition is set to a level (High or Low), then the sequence will resume on the opposite level, e.g., if the Halt Trigger condition is set to High Level on AUX5 then the sequence will halt when AUX5 goes high and resume when AUX5 goes low.

In order to halt on a specific pattern, the Halt Trigger must be provided 11 MCLKS + ~20-30ns before the end of the pattern otherwise it will halt on the next pattern.

Resume options:

- CPU Resume
- CPU Single-Step
- Opposite Halt Trigger Level (used for System Clutch...see example below).

If the Halt Trigger is set to Rising or Falling edge then the Edge Test Clear should be set to Event True to enable resume.

Halt Examples

The following lists common examples using Halts:

Halt on Error: Pause on Phase 4 falling edge (right after the output data is formatted and before the input data is to be captured) of Sequence Step 3 (a one pattern Sequence Step) and Resume on the Rising Edge of Aux. 5.

- Set the Halt Mode: Pattern Fail

Halt on Pattern: Halt on Pattern 6 in Sequence Step 4 (like a breakpoint)

- Set the Halt Mode: Sync1
- Set Sync1 Mode: Single Step
- Set Sync1 Step: 4
- Set Sync1 Offset: 6

- Set Sync1 Length: 1

Halt on End — Halt at the end of the Sequence.

- Set the Halt Mode: Sequence

Halt on Event — Halt on an external Rising Edge signal occurring on the Aux. 2 Input.

- Set Halt Trigger Source: AUX2
- Set Halt Trigger Test Condition: Rising Edge
- Set Halt Trigger Edge Test Clear: Event True

System Clutch — Simulate a system clutch function using the Aux. 8 input (an active high clutch).

- Set Halt Trigger Source: AUX8
- Set Halt Trigger Test Condition: High Level

A “high” on Aux. 8 causes a halt at the end of the Pattern and a “low” resumes (a Resume is not needed in this case). Thus the actual duration of the Halt will most likely be longer than the duration of the System Clutch.

Halt Notes

When halted, the CPU may access the Data, Record and Probe memories.

A Resume will be ignored while memory access is granted.

When the timing requirements are not met for completing the capture of the response data prior to the Halt, the response data and related error counting/logging may be corrupted.

Sequencer Operation

The “Sequencer” is a Mealy state machine that controls the flow and timing of the digital test patterns hereafter referred to as patterns.

A “Sequence Step” defines a subset of the total number of patterns to be applied to the UUT and defines the following properties:

- The location and the number of patterns to be output. The pattern data describes both the Stimulus to be applied to the UUT and how the response from the UUT is to be examined (includes expect data if applicable) for each channel.
- The timing to be used for the stimulus response (T0CLK period, phase and window timing).
- The Clocks per Pattern (CPP) to be used for each pattern in this sequence step from 1 to 256.

- Waveform selection control and Waveform Table to use (1 of 256).
- The Phase Trigger Type for each Phase (Pattern or System Clock). This is applicable when CPP is greater than 1.
- Sequence Flag state (2).
- Pattern Control Instructions.

The Sequencer is always running unless “Paused” or “Halted”.

The sequencer memory contains one or more of the following:

- A “Primary Sequence” is composed of one or more “Sequence Steps” and describes in total how all the Patterns will be applied to a UUT for a dynamic stimulus/response test.
- A “Standby Sequence” is a step which defines the power-up/reset state of the sequencer. It runs continuously and may output one pattern but response data is ignored. Step 0 is the default standby step for power on, sequence reset and master reset.
- An “Idle Sequence” is an optional step that may be run before and/or after the primary sequence. The Idle Sequence run after a Primary Sequence may be different than the one run before a Primary Sequence. An Idle Sequence always runs continuously and may output one or more Patterns, but response data is ignored.

A “Finishing Sequence” is the Seq. Step that is run after the Primary Sequence. It may be an Idle Sequence or a Standby Sequence.

One or more Sequence Steps may be designated as a Subroutine.

The Pattern Control Instructions allows for looping, jumping, pausing and halting.

Basic Sequence Step Flow

The basic sequence step flow is described by the following:

1. First pattern of the standby step executed continuously.
2. An optional idle step can be executed that will cause the idle step patterns to be executed continuously.
3. Primary step executed. After completing the standby (or Idle) step, jump to the specified primary step.
4. After all patterns of the primary step are executed, jump to the standby (or idle) step if the last step flag is set otherwise jump to the next higher step number and repeat step 4.

The following sequence step definitions will be used in the basic sequence step flow examples:

Step Number	Number of Patterns	Starting Pattern Address (PA)	Last Step Flag
0	1	0	Yes

Step Number	Number of Patterns	Starting Pattern Address (PA)	Last Step Flag
1	2	4	Yes
2	3	8	Yes
3	2	12	No
4	3	16	Yes

Table 6-9 Sequence Step Flow Example Definitions

Highlighted table cells indicate continuous output.

Sequence Flow Example 1

Execution Commands:

1. Sequence Reset.
2. Execute Step 1.

Command	Sequence Reset	Execute 1		Last Step Standby
Step	0	1		0
PA	0	4	5	0

Table 6-10 Sequence Flow Example 1

Sequence Flow Example 2

Execution Commands:

1. Sequence Reset.
2. Execute IDLE Step 2.
3. Execute Step 1.

Command	Sequence Reset	Execute Idle 2			Execute 1		Last Step Standby
Step	0	2			1		0
PA	0	8	9	10	4	5	0

Table 6-11 Sequence Flow Example 2

Sequence Flow Example 3

Execution Commands:

1. Sequence Reset.
2. Execute Step 3.

Command	Sequence Reset	Execute 3					Last Step Standby
Step	0	3		4			0
PA	0	12	13	16	17	18	0

Table 6-12 Sequence Flow Example 3

Sequence Flow Example 4

Execution Commands:

1. Set Finish Mode Idle Step 1.
2. Sequence Reset.
3. Execute Step 2.

Command	Sequence Reset		Execute 2			Last Step Idle	
Step	0		2			1	
PA	0		8	9	10	4	5

Table 6-13 Sequence Flow Example 4

Sequence Flow Example 5

Execution Commands:

1. Set Finish Mode Standby Step 1.
2. Sequence Reset.
3. Execute Step 2.

Command	Sequence Reset	Execute 2			Last Step Standby
Step	0	2			1
PA	0	8	9	10	4

Table 6-14 Sequence Flow Example 5

Pattern Control Instructions

The pattern control instructions allows the basic sequence step flow to be altered based on internal or external conditions. The pattern control instructions allows the following actions to a primary sequence step:

- Looping
- Jumping

Pattern Control Looping

There are two hardware structures available to loop a primary sequence, burst counter and step loop counter.

Burst Counter

The burst counter allows the primary step(s) to be looped from 0 (continuous) to 1048576 times.

Burst Loop Example

Using the sequence step definitions from Table 6-9:

Execution Commands:

1. Set Burst Count to 1000.
2. Sequence Reset.
3. Execute Step 1.

Command	Sequence Reset	Execute 1		Last Step Standby
Loop	Continuous	1000		Continuous
Step	0	1		0
PA	0	4	5	0

Table 6-15 Burst Loop Example

Step Loop Counter

There are sixteen step loop counters that are shared by all the sequence steps. The step loop counters have the following specifications:

- Can be set from 0 (disabled) to 65535
- Can be reloaded when terminal count is reached or disabled.
- Used as a qualifier for the jump pattern control.
- Loop count and counter selection is programmed per sequence step.

Step loop counters can be re-used when count complete (incomplete loop counters will continue where they left off when re-used).

There are two bits associated with each step loop counter. The first bit, Counter Active (CA), gets set when the loop counter is loaded with the specified count. The second bit, Use Counter Once (UCO), is programmed by the user for each loop counter. If UCO is set, the CA bit will **not** be reset when the loop counter completes, thus the counter cannot be re-loaded. If UCO is not set, the CA bit is reset when the count is complete and the next sequence step begins.

A step loop counter paired with an unconditional jump is called a counted loop. A counted loop will only jump until the selected counter is complete.

A step loop counter paired with a conditional jump is called a counted loop with termination. A counted loop with termination will only jump until the jump condition is true or the selected counter is complete.

If the sequence step designated as the last step has loops then upon completing the loops, execution will proceed to the Finishing Sequence.

Step loop counters can be nested but only one can end on a given Sequence Step.

Step loop counters can be used around one or more Sequence Steps and the group of sequence steps need not be consecutive i.e., one or more intermediate Jumps could have occurred.

Step Loop Example 1

Using the sequence step definitions from Table 6-9:

Execution Commands:

1. Set LC0 terminal count action to Reload
2. Set Step 3 Loop Count to 16 and Counter to 0, Jump Always to Step 3
3. Set Step 4 Loop Count to 31 and counter to 0, Jump Always to Step 4
4. Sequence Reset
5. Execute Step 3

Command	Sequence Reset	Execute 3					Last Step Standby
Loop	Continuous	17		32			Continuous
Step	0	3		4			0
PA	0	12	13	16	17	18	0

Table 6-16 Step Loop Example 1

Step Loop Example 2

Using the sequence step definitions from Table 6-9:

Execution Commands:

1. Set LC0 terminal count action to Disable
2. Set Step 3 Loop Count to 16 and Counter to 0, Jump Always to Step 3
3. Set Step 4 Loop Count to 31 and counter to 0, Jump Always to Step 4
4. Sequence Reset
5. Execute Step 3

Command	Sequence Reset	Execute 3					Last Step Standby
Loop	Continuous	17		1			Continuous
Step	0	3		4			0
PA	0	12	13	16	17	18	0

Table 6-17 Step Loop Example 2

Pattern Control Jumping

Jumping allows the normal sequential flow of sequence steps to be altered.

Pattern Control Instruction Details

Table 6-19 describes the pattern flow based on the jump type and the LAST STEP and RTN control flags. It's a flow chart in tabular form. The "Jump" column designates that the Test Condition was "True".

Table 6-18 lists the acronyms used in Table 6-19 to describe the pattern flow.

Acronym	Description
JSA	Jump Sequence Address
CA	Counter Active
LC	Loop Count
LCD	Loop Counter Done
UCO	Use Counter Once
INSUB	Gosub active flag
RSA	Return Sequence Address
JUMP	Jump condition true flag
LAST STEP	Last step flag
RTN	Gosub return flag
SUBRT	Subroutine

Table 6-18 Pattern Control Acronyms

JUMP	LAST STEP	RTN	SUBRT	Action/Comments
0	0	0	0	Proceed to the next sequential step.

1	0	0	0	If (LC=0)
				Jump to JSA.
				Else if (CA=0)
				Load the designated loop counter with LC, set LCD=0, set CA=1 and jump to JSA.
				Else if (LCD=0)
				Decrement the loop counter and set LCD=1 if loop counter equal to zero. Jump to JSA.
				Else
				Reset CA if UCO=0 and proceed to the next sequential step.

1	0	0	1	If (INSUB=1)
				Set multiple subroutine bit in the condition register and jump to finish sequence.
				If (LC=0)
				Set INSUB=1, save the next sequential step as RSA and jump to JSA.
				Else if (CA=0)

JUMP	LAST STEP	RTN	SUBRT	Action/Comments
				Load the designated Loop Counter, set CA=1, set INSUB=1, save the next sequential step as the RSA and jump to JSA. Else if (LCD=0) Decrement the loop counter and set LCD=1 if loop counter equal to zero. Jump to JSA. Else Reset CA if UCO=0 and proceed to the next sequential step.

0	0	1	0	If (INSUB=1)
				Jump to the RSA and set INSUB=0.
				Else
				Set return subroutine error bit in the condition register and proceed to the next sequential step.

1	0	1	0	If (LC=0)
				Jump to JSA.
				Else if (CA=0)
				Load the designated loop counter with LC, set LCD=0, set CA=1 and jump to JSA.
				Else if (LCD=0)
				Decrement the loop counter and set LCD=1 if loop counter equal to zero. Jump to JSA.
				Else if (INSUB=1)
				Reset CA if UCO=0, set INSUB=0 and jump to the RSA.
				Else
				Set return subroutine error bit in the condition register and proceed to the next sequential step.

1	0	1	1	If (INSUB=1)
				Set multiple subroutine bit in the condition register and jump to RSA and set INSUB=0.
				Else
				Set return subroutine error bit in the condition register.
				If (LC=0)
				Set INSUB=1, save the next sequential step as the return sequence address and jump to JSA.
				Else if (CA=0)
				Load the designated Loop Counter, set CA=1, set INSUB=1, save the next sequential step as the RSA and jump to JSA.
				Else if (LCD=0)
				Decrement the loop counter and set LCD=1 if loop counter equal to zero. Jump to JSA.
				Else
				Reset CA if UCO=0 and proceed to the next sequential step.

0	1	0	0	If (INSUB=1) Set return subroutine error bit in the condition register. Jump to finish sequence.
1	1	0	0	If (LC=0) Jump to JSA. Else if (CA=0) Load the designated loop counter with LC, set LCD=0, set CA=1 and jump to JSA. Else if (LCD=0) Decrement the loop counter and set LCD=1 if loop counter equal to zero. Jump to JSA.

JUMP	LAST STEP	RTN	SUBRT	Action/Comments
				Else Reset CA if UCO=0 and proceed to the finish sequence.

0	1	1	0	If (INSUB=1)
				Jump to the RSA and set INSUB=0.
				Else
				Set return subroutine error bit in the condition register and proceed to the finish step.

1	1	0	1	If (INSUB=1)
				Set multiple subroutine bit in the condition register and jump to finish sequence.
				If (LC=0)
				Set INSUB=1, save the next sequential step as the return sequence address and jump to JSA.
				Else if (CA=0)
				Load the designated Loop Counter, set CA=1, set INSUB=1, save the next sequential step as the return sequence address and jump to JSA.
				Else if (LCD=0)
				Decrement the loop counter and set LCD=1 if loop counter equal to zero. Jump to JSA.
				Else
				Reset CA if UCO=0 and proceed to the finish step.

1	1	1	0	If (LC=0)
				Jump to JSA.
				Else if (CA=0)
				Load the designated loop counter with LC, set LCD=0, set CA=1 and jump to JSA.
				Else if (LCD=0)
				Decrement the loop counter and set LCD=1 if loop counter equal to zero. Jump to JSA.
				Else if (INSUB=1)
				Reset CA if UCO=0, set INSUB=0 and jump to the RSA.
				Else
				Set return subroutine error bit in the condition register and proceed to the finish step.

1	1	1	1	If (INSUB=1)
				Set multiple subroutine bit in the condition register, INSUB=0 and jump to RSA.
				Else
				Set return subroutine error bit in the condition register.
				If (LC=0)
				Set INSUB=1, save the next sequential step as the return sequence address and jump to JSA.
				Else if (CA=0)
				Load the designated Loop Counter, set CA=1, set INSUB=1, save the next sequential step as the return sequence address and jump to JSA.
				Else if (LCD=0)
				Decrement the loop counter and set LCD=1 if loop counter equal to zero. Jump to JSA.
				Else
				Reset CA if UCO=0 and proceed to the finish step.

Table 6-19 Pattern Control Flow

PXI Backplane Trigger Bus

The PXI trigger bus is comprised of eight TTL signals that are shared across the PXIe chassis backplane. Larger chassis group the signals in to segments. Within each segment the signals operate as open collector with a pullup. Between the segments, the triggers can be connected by the PXIe backplane making them bi-polar uni-directional. This section discusses the open collector uses of the TTL triggers within a single segment.

Trigger Bus description

TTLTRG Bus (8 backplane signals): normally active low.

Trigger Bus Applications

Inter-module communications (for Sequencers configured in a DTS configuration):

- Communicating a Channel Test Trigger for a Conditional Jump
- Communicating a Synchronization Signal from the Primary Sequencer that all the coupled Synchronizers can check themselves against.
- Communicating a Sequence Reset to all coupled sequencers: primarily used for re-synchronizing coupled Sequencers
- Communicating a Master Reset to all coupled sequencers
- Communicating a Driver Disable to all coupled sequencers that can disable all the channel drivers at once.

Receive a signal from another instrument in the PXIe chassis (needs to go to the Primary Sequencer):

- External Start and/or Stop
- External Jump
- External Halt, Pause or Resume

Trigger another instrument in the PXIe chassis. Possible signal choices are:

- A sync pulse
- A Seq. Flag
- An Aux. Input
- Idle Active
- Seq. Active
- A Channel Test

Normal Operation

For inter-module communication, the “active high” and “active low” state of the backplane bus is handled automatically.

For communications with other instruments, the "active high" or "active low" state of the bus must be considered when:

- Receiving a signal from another instrument
- Providing a signal to another instrument

The signals driving out onto these buses or coming in from these buses can be inverted.

Normal Operation Example

To do a Jump Trigger on an Aux. Input located on a non-primary sequencer:

- Select the TRG Bus to be used and select the Aux. signal to drive it.
- Invert the output, if the Aux. signal is active low.
- On the primary, select the same TRG Bus signal as the Jump Trigger and use a "High" or "Rising Edge" test condition.

To do a Jump Trigger on a single input Channel (Channel Test):

- Select the Channel Test to be used (1 of 4) on the DTI which covers that channel.
- Pick the desired channel and unmask the channel test for that channel.
- Set the expect level for the channel to be the level desired for a trigger.
- Select the TRG Bus to be used and select the Channel Test signal to drive it.
- On the Primary, select the same TRG Bus signal as the Jump Trigger and use a "High" or "Rising Edge" test condition.

Advanced Operation Examples

To do a Jump Test on the OR of several Channels:

- Select the Channel Test to be used (1 of 4) on the sequencer(s) for the channels to be ORed.
- Unmask Channel Test for these channels.
- Set the expect level for each channel to be the level desired for a trigger.
- Select the TRG Bus to be used and select the Channel Test signal to drive it.
- On the Primary, select the same TRG Bus signal as the Jump Trigger and use a "High" or "Rising Edge" test condition.

To do a Jump Test on the AND of several Channels:

- Select the Channel Test to be used (1 of 4) on the sequencer(s) for the channels to be ANDed.

- Unmask Channel Test for these channels.
- Set the expect level for each channel to be the complement of the level desired for a trigger.
- Select the TRG Bus to be used and select the Channel Test signal to drive it.
- On the Primary, select the same TRG bus signal as the Jump Trigger and use a “Low” or “Falling Edge” test condition.

Notes

1. The TTLTRG Bus has a weak pullup, thus the rise time will be quite slow. As such, the trailing edge of an active low signal will be delayed up to 40ns more than the leading edge. Triggering on a falling edge is recommended when delay is a concern.
2. There is a way to do AND/OR or OR/AND channel tests between groups of channels on different sequencers.
3. Since Aux Inputs can drive TRG Bus lines, they can be ORed. ANDed or even combined with Channel Test signals in various ways. For example, an Aux. input could be a qualifier for a Channel test.
4. When a TRG Bus line is configured to drive out a “Synchronization Signal”, “Sequence Reset”, “Master Reset” or “Driver Disable” signal, the corresponding input of these signals to the Primary will be automatically configured.
5. The combination of up to 4 TRG Bus signals may be used to formulate a 1 of 16 “vector” to the sequencers so one can do a vectored jump to 1 of 16 locations based on the state of these four signals.

This page was left intentionally blank.

Appendix A
Specifications

The 32 DTI channels can operate as dynamic I/O controlled by the data sequencer or as PMU I/O. The following lists the specifications of each modes.

Data Sequencer

Master Clock

500MHz Accuracy	 50ppm
Frequency Synthesizer Range Accuracy Internal Reference External Reference	 40KHz – 500MHz 50ppm PXI CLK100 or CLK10 AUX1-12 (10MHz to 100MHz)

Timing Characteristics

Internal I/O Data Rate (using the 500 MHz master clock)	~15.256 kHz to 50 MHz (with CPP = 1) 59.6 Hz Min. (with CPP = 256)
Internal I/O Data Rate (using the Freq synthesizer at 40 kHz as the master clock)	~1.22 Hz Min. (with CPP = 1) ~0.048 Hz Min. (with CPP=256)
Timing Set Options Indexed Per Step Multiple Per Step Single	3 256 Timing Sets with 4 phases and 4 windows and 4K sequence steps 1K Timing Sets with 4 phases and 4 windows and 1K sequence steps (one for each sequence step) 4K Timing Sets with 1 phase and 1 window and 4K sequence steps (one for each sequence step)
T0Cycle Period Range (per Sequence step)	20 ns to ~65.5 µs (using the 500 MHz master clock)
T0Cycle Timing Resolution	1 ns (using the 500 MHz master clock)
Phase Programming Range	0 ns to ~65.5 µs (using the 500 MHz master clock)
Window Programming Range	0 ns to ~65.5 µs (using the 500 MHz master clock)
Phase/Window Timing Resolution	1 ns (using the 500 MHz master clock)
Minimum Phase/Window Pulse Width	8 ns (using the 500 MHz master clock)
Phase/Window Reference	Phases: System or Pattern Clock (selectable per sequence step) Windows: Pattern Clock only
Phase/Window Range	0 to Pattern Period – 8 counts
Window Dead Time	~13 ns at the end of the Pattern period
Clocks per Pattern (CPP)	1 to 256 (selectable per sequence step)

Pause/Pattern Clutch	Phases and Windows are frozen when asserted Can pause based on an external signal (levels or edges) Can pause based on a phase edge Can resume based on an external signal (levels or edges) or CPU Resume Can resume after a programmed delay (2 timers available). Useful to implement a Wait Pattern timeout can be programmed to generate an event if a pattern is paused too long. See the **Pause and Halt** section of Chapter 6 for additional details about the use of pause.
Halt/System Clutch	All phases will complete their action for the current pattern. Can halt based on an external signal (levels or edges) Can halt on error (at slower data rates) Can halt on a sync pulse (used as a breakpoint) Also used for single-stepping (The latter three require a CPU Resume: see spec for additional clarification.) See the **Pause and Halt** section of Chapter 6 for additional details about the use of halt.
Pause/Pattern and Halt/System Clutch Sources	TTLTrg0-7, F/P AUX I/O 1-12, CH 1-32 (with mask/expect), and Phase 1-4 (for Pause)
External T0Cycle Range	< 1 kHz to ~48 MHz
External T0Cycle Edge Selection	Can use either edge or both edges of a signal to define the T0 Cycle period. Can also divide the incoming clock by 2.
External T0Cycle Delay Adjustment	A programmable delay is provided to adjust the timing relationship of the T0Cycle with respect to the Ext. input (2 ns resolution; 0-64K ns range with the 500 MHz master clock).
External T0Cycle Clock Source	F/P AUX I/O 1-12
Clock/Waveform Outputs	Up to 4 waveforms can be output during a pattern (each sequencer). They are provided in lieu of certain phases and windows. They can be output on any AUXI/O Channel (two can actually be output on any data channel). They can be any arbitrary or repeating waveform. There are up to 16 waveform tables. Output resolution/step size is 1 ns with the 500 MHz master clock. The width (high or low) should not be too narrow with respect to the driver rise fall time capabilities of the Channel being used to output it.

Stimulus/Capture

Output Timing Signals Phase	1 to 4 based on timing mode
Input Timing Sources Window	1 to 4 based on timing mode
Data Output Formats	Force: Low, High, tristate Format: NR, RT, R0, R1, RC, Complement Surround Output the Phase or its complement (used to output waveforms on channels)

Capture Modes (per channel)	Mask Opening edge of window Closing edge of window Window (input data must match “expect” from open edge to close edge of the window)
Pattern Memory Depth	256K
Record Memory Depth	256K
Pattern (Stimulus/Expect) Data	Output: H, L, Tristate Expect: Good 1, Good 0, OK, between or mask Keep last Toggle last Accumulate a CRC16 (based on a Good 1 only)
Static Mode	Utilizes a Single Word Sequence Step Delay Range: 1 ns to ~65 µs (master clock @ 500 MHz) Delay Range: 100 ns to 6.5 ms (master clock @ 5 MHz) Resolution: 1 ns (for a 500 MHz master clock) Resolution: 100 ns (using a 5 MHz master clock) Note: Repeat pattern data is updated based on the static drive state
Static Mode	Utilizes an independent static stimulus/response path that doesn’t alter the Repeated pattern data of dynamic tests Static test is not run in parallel with a standby Sequence Step. Response Delay from 100ns to ~6.5ms in 100ns steps.

Recoding Mode

Recording Modes (per Sequence Step)	Record errors for programmable inputs that have a Good 1 and Good 0 Record errors for single-ended inputs that have only a Good 1 Record raw data based on NOT a Good 0 Record raw data based on a Good 1
Recording Type	Un-expanded: Record data at the same index as the stimulus (will overwrite data when looping) Expanded: Records data sequentially. A separate Record Index Memory stores information that allows the recorded data to be re-aligned with the original data.
Error Address Recording	Separate Error Address Record Memory records where errors occurred in the Record Memory. Limit: 1K errors.
Record Offset	Used to compensate for round-trip driver/receiver delay and also cabling delay to the UUT. Can also be used to allow windows to effectively close at the end of the T0Cycle. Resolution: 1 master clock Range: 2-63 master clocks

Sequencer

General	Channels: 32 per sequencer Modes: Static, Dynamic
Sequence Memory	Sequence Size: 1024 or 4096 Steps

Sequence Loop Counters	Loop Counters: 16 Loop Count can be different each time or continuous Loop counters may be nested Loop counters can be optionally re-loaded during a burst Only one can end on a sequence step
Loop Count Range	1-64K or continuous
Subroutine Characteristics	Output one or more Sequence Steps with or without looping. Cannot be nested. Has a designated “Return” Step.
Burst Count Range	1-1M or continuous
Jump Types	Conditional or unconditional Jumps at the end of a sequence step Vectored (1 of 16 destinations)
Conditional Jump Sources (per seq. step)	One of four Test Inputs Seq. Step PASS Seq. Step FAIL Seq. Step NOT a PASS (i.e. FAIL or indeterminate) Seq. Step NOT a FAIL (i.e. PASS or indeterminate) Burst PASS Burst FAIL
Conditional Jump Enable (CONDEN)	Per pattern
PASS/FAIL Pipeline	0-16 patterns
Burst Error Enable (BERREN)	Per pattern
Test Input Sources	TTLTrg0-7, F/P AUX I/O 1-12, Chan 1-32 (with mask/expect)
Test Input Sense	Rising edge, Falling edge, Hi-state or Low state
Sync Pulse Outputs	Outputs per Sequencer: 2 Modes: Start of Sequence, Start of Sequence Step Offset Range 0-1M patterns Pulse Width: 1-4095 patterns
AUX Outputs	Sync Pulses (2) Sequence Flags (2) Sequence/Idle Active T0Cycle Waveforms Phases/Windows A multitude of other signals
Sequence Standby Characteristics	A one word continuous sequence step that may be used to output “standby” data on power up or after a sequence reset. The CPU can access pattern data in this state.
Idle Sequence Characteristics	A continuous sequence step that may be used to output data before or after an active sequence. The CPU cannot access pattern data in this state. The Idle Sequence output after the active sequence may be different from the one output before.
Sequence Execution Control	Reset to Standby Sequence (CPU command) Run Idle Sequence (CPU or external command) Run Sequence (CPU or external command) Stop Sequence (CPU or external command) Single step by Pattern or Sequence Step. (see Halt function)

Burst Timeout Timer	A watchdog timer that limits the maximum execution time of a dynamic pattern set independently of pauses, halts and external clocks. On timeout, sets all outputs to tristate. Can be disabled. Range: 40 ns to ~86 seconds Resolution: 20 ns
Handshaking	See Pause function

Counter/Timer

Measurement Modes	Frequency Period Time Interval Totalize Timed Totalize Positive Pulse Negative Pulse
Input Source	CH1-32 (Uses Good 1) AUX1-12 Frequency Synthesizer CLK10 250 MHz Pulse Generator
Input Sense	Rising/Pos or Falling/Neg
Frequency/Period Measurement Source	Input 1
Frequency/Period Measurement Range	0.25 Hz to 250 MHz/4 ns to 4 s
Preset Aperture Windows	1 µs to 10 s in decade steps
Aperture Window Accuracy	0.1% +50 ppm
Frequency/Period Measurement Resolution	≥4 Digits with a 1 ms Aperture ≥5 Digits with a 100 ms Aperture ≥6 Digits with a 10 s Aperture
Time Interval Functions	Between Inputs 1 & 2; Positive/Negative Pulse Width of Input 1
Time Interval Range	~2 ns to ~4.29 s
Time Interval Resolution	1 ns
Time Interval accuracy	1 count + input comparator threshold uncertainty
Time Interval Reference Accuracy	50 ppm
Totalize (2 modes)	Timed with a Preset Aperture. Aperture defined by Input 3.
Preset Aperture accuracy	50 ppm
Max. Count	2^32-1
Max. Input Data Rate	250 MHz Note: CH and AUX input technology may limit the max. data rate that can be supported.
Input Trigger	Input 3
Trigger functions for Freq./Period, Time Interval & Totalize (mode 1 only)	Manual External (Input 3) Continuous
Events provided	Indicates when the data is ready to be read (may also generate an interrupt)

Pulse Generator

Description	Characteristics
Signal Routing	System Clock TTL Triggers Any Aux channel Counter Input
Pulse Resolution	10 ns, 20 ns
Run Mode	Continuous Continuous Start Single Start Single Step
Period	10 ns Resolution: Min: 20 ns Max: 42.94967297 s 20 ns Resolution: Min: 40 ns Max: 85.899345960 s
Delay	10 ns Resolution: Min: 20ns Max: 42.94967297 s 20 ns Resolution: Min: 20ns Max: 85.899345960 s
Width	10 ns Resolution: Min: 0ns Max: 42.94967297 s 20 ns Resolution: Min: 0ns Max: 85.899345960 s

I/O Channels

Description	Characteristics
I/O Type	Variable Voltage
Channels	32 SE or 16 DIFF Per channel relay isolation
Output Voltage Swing	500 mV to 9 V
Output Resolution	< 300 μV
Output Accuracy (DVH, DVH, VTT)	± 30mV
Output Range DVH DVL VTT	 -1.5V to +7V -2V to +6V -2V to +7V
Output Drive Current	± 50 mA typical (Source/Sink)
Output Impedance	Series (50 Ω), ± 4 Ω
Slew Rate (Selectable/Channel or custom)	0.2 V/ns 0.7 V/ns, 1.0 V/ns or 1.3 V/ns: typical
Input Threshold Resolution	< 300 μV

Description	Characteristics
Input Threshold Accuracy (CVH, CVL)	± 20mV
Input Threshold Range (CVH, CVL)	-2V to +7V
Skew (Chan. to Chan.)	< 3 ns (drive and compare)
Active Load Current Resolution	100 µA
Active Load Current Accuracy	± 0.5 mA
Active Load Current Range	-24mA to +24mA
Active Load V_{com} Resolution (CMH and CML)	< 300 µV
Active Load V_{com} Accuracy (CMH and CML)	± 20mV
Active Load V_{com} Range (CMH and CML)	0V to +5V
V_{com} Voltage for Isource = 12 mA	Channel pin – 2.5V
V_{com} Voltage for Isink = 12 mA	Channel pin + 2.5V
PMU Modes	Force Voltage (FV) Measure Voltage (MV) Force Current (FI) Measure Current (MI)
PMU Current Ranges (I_{max})	50mA 5mA 500µA 50µA 5µA
PMU FV Range	-2V to +7V
PMU FV Accuracy	± 5 mV
PMU MV Range	-2V to +7V
PMU MV Accuracy	± 5 mV
PMU FI Range	± I_{max}, Usable to ± 2 * I_{max}
PMU FI Accuracy	± 0.5% of I_{max}
PMU MI Range	± I_{max}, Usable to ± 2 * I_{max}
PMU MI Accuracy	± 1% of I_{max}
PMU Voltage Clamp Range	-2V to +7V
PMU Voltage Clamp Accuracy	± 100 mV
PMU Current Clamp Range	± I_{max}, Usable to ± 2 * I_{max}
PMU Current Clamp Accuracy	± 5% of I_{max}
DUT_GND Reference Input (per Driver/Receiver board)	Offset range: ±3 V Resistive load: 100 kΩ Bypass Relay: On or Off
Pin Electronics Monitoring (per channel)	All programmed levels Output and Input levels Temperature
Channel Over-voltage Protection	Clamped to 0.4 V beyond V+ or V- Max current 200mA for < 10ms Auto Shutdown: DC level within 1 V of V+ or V- A 5 µs spike exceeding V+ or V-
Channel Capacitance	<120 pF
Channel Crosstalk	<250 mV_{pk-pk}

Description	Characteristics
Voltage Monitoring (per Driver/Receiver board)	V+, V- and Front Panel DUT_GND
Auxiliary I/O Channels (per Driver/Receiver board)	LVTTL (8) LVDS (4) Differential AUX I/O is bi-directional

Power Supply

Peak and Dynamic Module Current Contributions

Supply Rail	I_{Pm} (A)	I_{Dm} (A)
+12 V	1.4	1.2
+3.3 V	0.5	0.45

Power Absorbed

Max. power absorbed	31 W

Environmental Conditions

All environmental conditions tested to MIL-PRF-28800F, Class 3

Temperature Operating Non-operating	 0° C to +50° C -40° C to +71° C
Relative Humidity	5% to 95% RH non condensing ≤ 30° C 5% to 75% RH above 30° C 5% to 45% RH above 40° C
Altitude Operating Non-Operating	 15,000 ft. maximum 15,000 ft. maximum
Shock	30 g peak, half sine, 11 ms pulse
Vibration	5 to 500 Hz
Bench Handling	4-inch drop at 45°
Cooling Requirements (10°C temp. rise) Min. Flow Rate	 4 l/s

Appendix B
Glossary of Terms and Acronyms

This appendix includes a list of many of the terms and acronyms used in this manual.

ADE	Application Development Environment
Assert	Rising edge of a Phase
AUX	Auxiliary
Bipolar	Sources and sinks current (single-ended)
CPP	Clocks per Pattern
CH	Channel (signal)
Channel Test	Allows any channel to be used as a test input (TEST1 or TEST2). It can also be used with other Channel tests to form a Vector Jump Index. It can even be used to start or stop a sequence.
Close	The falling edge of a Window
Comparator	Compares an input signal with a voltage reference level
Coupled	Used to describe a DTI that is included in a DTS chain
CMH	Commutating Voltage High
CML	Commutating Voltage Low
CVH	Compare Voltage High
CVL	Compare Voltage Low
DB	Digital Board
Differential	A pair of signals representing a state when one is at a high level the other is at a low level.
DR	Driver/Receiver Board
DTI	Digital Test Instrument
DRS/DTS	Digital Resource/Test Suite. Two or more adjacent DTIs synchronized together to form a digital test system with more than 32 channels.
DUT	Device Under Test
DVH	Drive Voltage High
DVL	Drive Voltage Low
EN	Enable
Error	A channel error is determined by comparing the channel response to the expect/mask conditions of the Pattern data.
ETB	External Timing Bus
GND_REF	Ground reference output from the pin electronics devices
Good “0”	A signal generated when an input signal is less than CVL
Good “1”	A signal generated when an input signal is greater than CVH
Idle	An execution state that outputs the entire pattern set of a specified step after a sequence burst. Pattern and record memory cannot be accessed by the user.
Indeterminate	An “indeterminate” PASS/FAIL condition occurs if there is neither a valid PASS nor a FAIL. This is discussed in more detail in the This is discussed in more detail in the **Pass Valid Mode** section of Chapter 6.
I/O	Input/Output

Jump	Used to "Jump" out of the normal sequential flow of Sequence Steps to another Sequence Step. The jump occurs at the end of the sequence step after all of the patterns have been output.
JTAG	Joint Test Action Group, IEEE 1149.1: serial interface that allows the serial PROM to be reloaded for in-field system upgrades.
CBUS	An internal Control Bus connecting the PXI Bridge to the Data Sequencers and the Driver/Receiver board's Control Logic
l/s	Liters per second (flow rate measurement)
LED	Light Emitting Diode
LVDS	Low-Voltage Differential Signaling
LVTTL	Low-Voltage TTL
MCLK	Master Clock
Open	The rising edge of a Window
PAT_CLK	Pattern Clock
Pass Valid	A signal which conveys a Pass Valid Mode setting. This is discussed in more detail in the **Pass Valid Mode** section of Chapter 6. See also "Valid Pass", below.
Pattern	One stimulus applied to and/or one response received from the UUT. Sometimes called a Word or Vector.
Pattern Set	A Pattern Set is one or more consecutive channel patterns
PBUT	Probe button input signal to the Sequencer for support of remote probe operations
Primary	Used to describe the DTI that provides all the timing for the sequencers that are part of the DTS chain. The primary module must be located in the rightmost slot position in the chassis relative to the DTIs that will be coupled.
PXIe	PCI Express Extensions for Instrumentation
Reference	A programmable DC voltage
Return	Falling edge of a Phase
Standby	An execution state that outputs the first pattern of a specified step after a sequence burst. Pattern and record memory can be accessed by the user.
Secondary	Used to describe the DTIs located between the primary and terminating modules. Individual sequencers can either be coupled or run independently from the primary module.
Sequence	A sequence is an ordered list of stimulus/response actions consisting of one or more sequence steps.
Sequence Burst	An execution of one or more patterns.
Sequence Step	A sequence step is a single element of a sequence. A sequence step selects a timing set, pattern set, loop count, jump condition and control flags.
Slew Rate	Rate of change of an output transition (typically in V/ns)
Terminator	Used to describe the DTI in the leftmost position of the DTS chain
Timing Set	A timing set is the structure that is created that defines the stimulus/response timing.
TO_CLK	System Clock
TPS	Test Program Set
TTL TRG	TTL Trigger
UUT	Unit Under Test
Valid Pass	A Valid Pass is one where no channel errors were detected but there must be at least one valid pattern expect code for each pattern in the sequence step. This is discussed in more detail in the **Pass Valid Mode** section of Chapter 6.
VIH	Voltage Input High Level (min.)
VIL	Voltage Input Low Level (max.)
VOH	Voltage Output High Level (min.)
VOL	Voltage Output Low Level (max.)