DAQ

NI 660*X* Register-Level Programmer Manual

November 2002 Edition
Part Number 370505B-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 55 11 3262 3599,
Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085, China 86 21 6555 7838,
Czech Republic 02 2423 5774, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Greece 01 42 96 427, Hong Kong 2645 3186, India 91 80 4190000,
Israel 03 6393737, Italy 02 413091, Japan 03 5472 2970, Korea 02 3451 3400, Malaysia 603 9596711,
Mexico 001 800 010 0793, Netherlands 0348 433466, New Zealand 09 914 0488, Norway 32 27 73 00,
Poland 22 3390 150, Portugal 210 311 210, Russia 095 238 7139, Singapore 65 6 226 5886,
Slovenia 3 425 4200, South Africa 11 805 8197, Spain 91 640 0085, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2528 7227, United Kingdom 01635 523545

For further support information, see the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

Important Information

Warranty

The NI PCI-6601, NI PCI-6602, NI PXI-6602, and NI PXI-6608 devices are warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

About This Manual

Chapter 1
About the NI 660*X* Devices

Chapter 2
Programming General-Purpose Counter/Timers

Chapter 3
Register Maps

About This Manual

The *NI 660X Register-Level Programmer Manual* describes the programmable features of the National Instruments 660*X* devices. This document is necessary for programmers who use platforms not supported by NI-DAQ.

This manual contains information necessary for the register-level programming of NI 660*X* devices. This manual includes the following information:

- Address and function registers for reading and writing data, and for configuring the input lines and the general-purpose counter/timers (GPCTs)
- Overview of several common GPCT operations
- Guidelines for configuring GPCTs

Note Use the change notification feature *only* if you are familiar with writing, installing, and uninstalling interrupt service routines (ISRs).

NI recommends programming the NI 660*X* device using NI-DAQ with application development environment software, such as LabVIEW, LabWindows™/CVI™, or Measurement Studio. Application software provides easier programming with the same flexibility as register-level programming and much faster development time.

Using the Manual Set

The *NI 660X Register-Level Programmer Manual* is one piece of the documentation set for the data acquisition (DAQ) system. For information about a specific NI 660*X* device, refer to the documentation for that device. The user manual provides installation procedures, connection requirements, programming options, specifications, and guidelines for operating the NI 660*X* devices.

Consult the accessory installation guides or manuals for information about accessory products. The terminal block and cable assembly installation guides or accessory user manuals explain how to connect system parts.

Conventions

This manual uses the following conventions:

<>	Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, DI0<3..0>.
»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
	This icon denotes a note, which alerts you to important information.
	This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
CompactPCI	CompactPCI refers to the core specification defined by the PCI Industrial Computer Manufacturer's Group (PICMG).
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes sections of code, programming examples, and syntax examples. This font is also used for the proper names of programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.
NI 660*X*	This phrase refers to all devices in the NI 660*X* family, including the NI PCI-6601 (NI 6601), NI PCI/PXI-6602 (NI 6602), and the NI PXI-6608 (NI 6608).
PCI	Peripheral Component Interconnect—PCI is a high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA.
PXI	A rugged, open system for modular instrumentation based on CompactPCI, with special mechanical, electrical, and software features.

Related Documentation

The following documents contain information that you might find helpful as you read this manual:

- *6601/6602 User Manual*, available at `ni.com/manuals`
- *Note to Users: About Your 6608 Device*, available at `ni.com/manuals`
- *PCI Local Bus Specification Rev 2.1*, available from `www.pcisig.com`
- *PICMG 2.0 R3.0, CompactPCI Specification*, available at `www.picmg.org`
- *PXI Specification*, Revision 1.0, available at `www.pxisa.org`

1

About the NI 660*X* Devices

This chapter provides a summary of the NI 660*X* devices and includes a brief overview of the MXI Interface to Everything (MITE) and NI-TIO components.

The NI 660*X* devices are digital I/O (DIO) and timing I/O (TIO) devices for the PCI bus in PC-compatible computers and PXI or CompactPCI chassis.

The NI 6601 features four 32-bit counter channels and eight lines of individually configurable, TTL/CMOS-compatible DIO. The NI 6602 and NI 6608 have the same capabilities as the NI 6601, but feature eight 32-bit counter channels and an 80 MHz clock. The NI 6608 features an oven-controlled crystal oscillator (OCXO) for greater timing accuracy.

Refer to the *6601/6602 User Manual* or the *Note to Users: About Your 6608 Device* for more information about device functionality, installation, connections, and safety guidelines.

Cautions Using the NI 660*X* devices in a way inconsistent with the *6601/6602 User Manual* and the *Note to Users: About Your 6608 Device* can cause injury or equipment damage. NI is *not* liable for damage or injury resulting from incorrect use.

Incorrectly programming NI 660*X* devices can cause permanent damage to the devices. For example, some I/O pins can be driven from several outputs. Two sources driving the same I/O pin could burn out the I/O pin and destroy one of the ASICs. Because the NI 6602 and NI 6608 have two NI-TIO application-specific integrated circuits (ASICs), they are particularly subject to this kind of damage. If using both ASICs, configure the second NI-TIO to use the I/O pins for the second NI-TIO. Doing so prevents both NI-TIOs from connecting to the same I/O pins.

Structural Overview

The following functional groups make up the NI 660*X* circuitry:

- MITE PCI interface ASIC
- NI-TIO (0) GPCT ASIC
- NI-TIO (1) GPCT ASIC, available only on the NI 6602 and NI 6608

Figure 1-1 illustrates the data flow of the NI 660*X* devices.

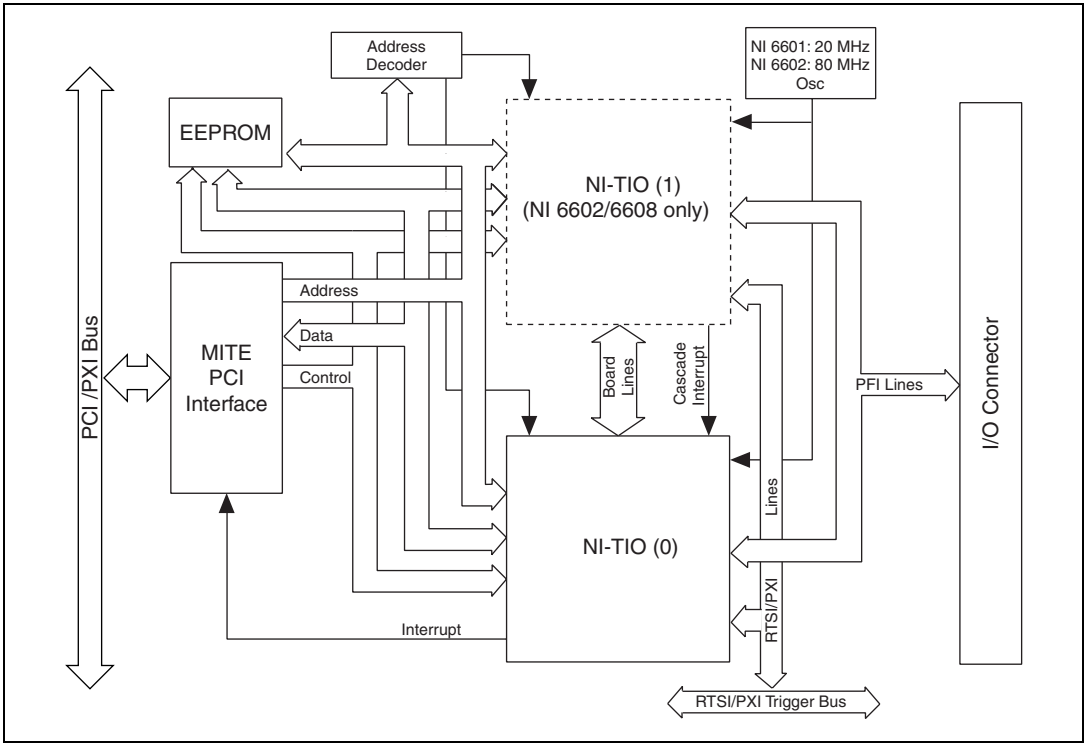

Figure 1-1. NI 660*X* Device Data Flow

MITE PCI Interface ASIC

The MITE is the interface for register accesses, interrupts, and direct memory access (DMA). The MITE enables communication between the NI 660*X* device and the PCI or PXI bus. Once initialized, the MITE enables communication with other NI 660*X* components.

NI-TIO General-Purpose Counter/Timer ASIC

Each NI-TIO contains four independent general-purpose counters. Each counter has a source, two gates, an output, and multiple load registers. The NI 6601 has one NI-TIO—NI-TIO (0)—and the NI 6602 and NI 6608 each have two NI-TIOs—NI-TIO (0) and NI-TIO (1).

Caution To avoid double-driving the counter outputs on the NI 6602 and NI 6608, set the Counter_Swap bit in the *Clock Config Register* to configure the second NI-TIO to use counters 4 through 7. If incorrectly configured, both NI-TIOs can use the I/O connectors for counters 0 through 3, which could permanently damage the device.

NI-TIO GPCTs

Each NI-TIO consists of four independent 32-bit up/down counters. These counters are identical, except for the internal routing of the counter inputs and outputs. Each counter has associated load and save registers and a control structure for implementing some common counting and TIO functions. The timing functions include period measurement, pulse width measurement, event counting, single-pulse generation, and pulse-train generation with programmable frequency and duty cycle.

Note Most functions can operate using only one general-purpose counter.

Measurement functions have two operational modes: single mode and buffered mode. Single-mode functions obtain only one measurement, and buffered mode functions obtain a series of consecutive, gap-free measurements.

You can select GPCT input signals from the external TIO pins on the NI 660*X* devices. Available input options depend on the GPCT input type. Output generation and timing measurement are the two primary GPCT modes.

Features

The NI-TIO has the following features:

- Four independent 32-bit binary up/down counters
- Count-up/count-down control through hardware or software
- Programmable counter source and gate selection from 19 signal sources
- Programmable input and output signal polarities
- Single-pulse or continuous pulse generation output

- Inter-event (relative) timestamping
- Two sets of save registers so that you can save the counter value through an external control signal or through software command
- Current count value readings that do not affect circuit operation
- Interrupts based on terminal count (TC)—rising edge, falling edge, or any edge
- Quadrature encoder position measurements

Note The NI 660*X* devices do *not* support binary-coded decimal (BCD) counting and time-of-day counting.

Simplified Model

Each GPCT contains four identical 32-bit binary up/down counters. Figure 1-2 shows a simplified model of the GPCT.

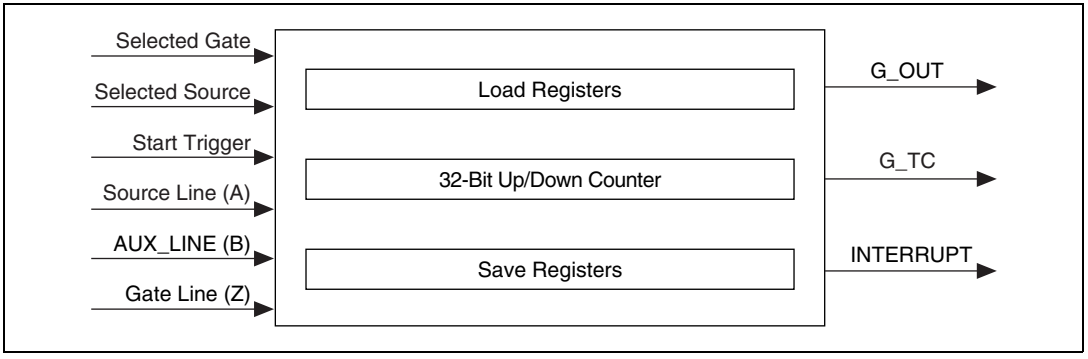

Figure 1-2. General-Purpose Counter/Timer Simplified Model

Each GPCT has a source input (Selected Source), a gate input (Selected Gate), and a second gate input (AUX_LINE) that doubles as an up/down control input. When the counter is enabled, rising edges on the Selected Source input cause the counter to increment or decrement. The Selected Gate input acts as a general-purpose control signal and can operate as a counter trigger signal, a counter enable, a save signal, a reload signal, an interrupt, an output control signal, a load register select signal, or a counter disarm.

The counter outputs are the signals labeled G_OUT and INTERRUPT. G_OUT is a TC-related counter output signal that can toggle on every counter TC or directly output the counter TC signal. INTERRUPT is a

counter output interrupt signal routed to the MITE. The load registers reload the counter with new count values, and the save registers save the counter contents until the software can read them.

Pulse Generation

In pulse generation mode, the counter decrements until reaching terminal count, the output of the counter changes, and a load register reloads the counter. The source may be a known frequency, in which case both the duty cycle and the output frequency are known. The source can also be an input that must be divided by an integer. The operation can be single-shot, repetitive, or triggered from the gate signal.

The NI-TIO has four available load registers—Load A and Load B in two X and Y bank cells. With this flexibility, you can generate any duty cycle and reprogram the load registers to change the properties on the fly. The output is typically programmed to alternate on each terminal count, but it can be programmed to toggle on the terminal count signal to create the smallest pulse or highest frequency output.

Figure 1-3 illustrates the pulse generation data flow.

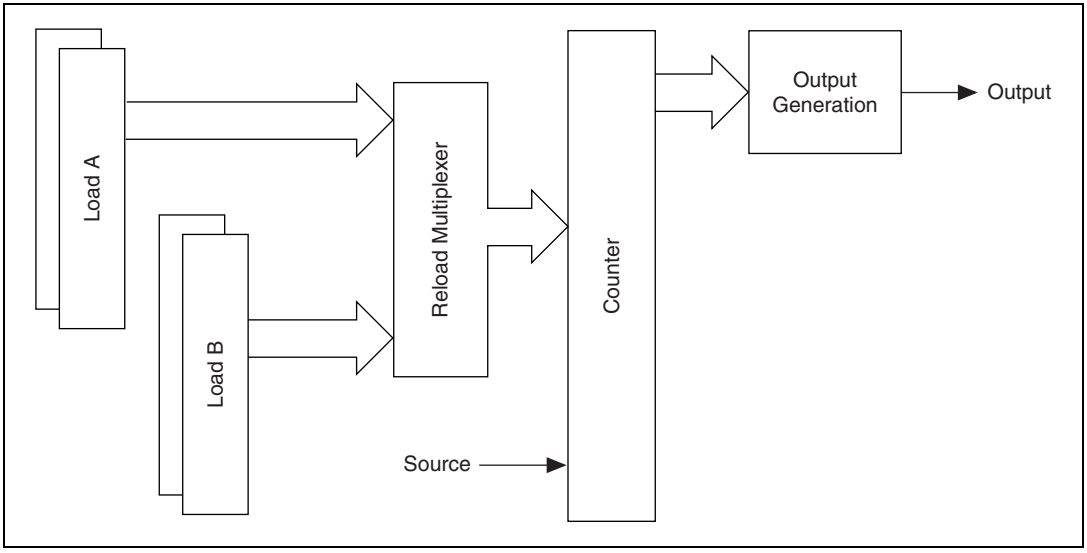

Figure 1-3. NI-TIO Pulse Generation Data Flow

Measurement

In measurement modes, the counter measures the source or the gate properties. Source edges increment or decrement the counter, and the gate signal stops and starts the counter. During buffered acquisitions, the gate signal also latches the counter state for either a software or DMA reading of the measurement. For measurements such as period, semiperiod, or pulse width, the counter uses a known timebase (usually the fastest internal timebase) for clocking, and the gate signal starts and stops the counter while latching the measurement. With event counting and frequency measurements, an unknown signal applied to the source and the gate is of a fixed time to create a window of measurement.

Figure 1-4 illustrates the measurement data flow.

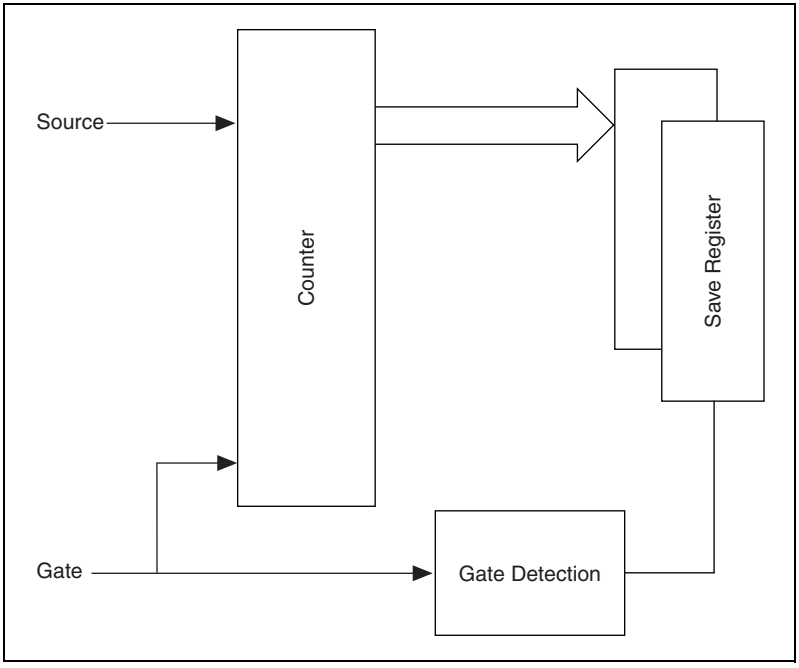

Figure 1-4. Measurement Data Flow

Oscillator

The signal from the oscillator is the internal timebase for the NI 660*X* devices. The NI 6601 uses a 20 MHz oscillator, and the NI 6602 and NI 6608 each use a 80 MHz oscillator.

EEPROM

The EEPROM stores PCI Plug and Play configuration information. The MITE reads this memory during system startup.

Real-Time System Integration (RTSI)

The NI-TIO ASICs can access the RTSI bus to share triggers or to clock signals between DAQ devices. The RTSI bus on the NI 6602 and NI 6608 can share triggers between the two NI-TIOs.

Caution Only one NI-TIO or NI 660*X* device should drive a RTSI line at a time. Double-driving a RTSI line can damage the NI 660*X* device.

Other Features

The NI-TIO includes several other features for specific measurement or generation cases, including an interface to quadrature encoders for position and velocity measurement of rotating devices. Additionally, the NI-TIO second gate allows edge-separation measurements.

Using PXI with CompactPCI

Using PXI-compatible products with standard CompactPCI products is an important feature of the *PXI Specification*, Revision 1.0. Using a PXI-compatible plug-in device in a standard CompactPCI chassis enables you to use basic plug-in device functions, but PXI-specific functions are not available. For example, the RTSI bus on PXI-bus NI 660*X* devices is available in a PXI chassis, but not in a CompactPCI chassis.

The CompactPCI specification enables vendors to develop sub-buses that coexist with the basic PCI interface on the CompactPCI bus. Compatible operation is not guaranteed between CompactPCI devices with different sub-buses or between CompactPCI devices with sub-buses and PXI. The standard implementation for CompactPCI does not include these sub-buses. The NI PXI-6602 and NI PXI-6608 work in any standard CompactPCI chassis adhering to the *PICMG 2.0 R3.0, CompactPCI Specification*.

The J2 connector of the CompactPCI bus implements the PXI-specific features. The PXI device is compatible with any CompactPCI chassis whose sub-bus does not drive these lines.

Caution Driving these lines with the sub-bus may cause damage to the NI 660*X* device.

Even if the sub-bus is capable of driving these lines, the PXI device is still compatible with the CompactPCI chassis if the pins on the sub-bus are disabled by default. Refer to Table 1-1 for a list of the J2 pins the NI PXI-6602 and NI PXI-6608 use.

Table 1-1. Pins Used by NI PXI-660*X* Devices

NI PXI-6602/6608 Signal	PXI Pin Name	PXI J2 Pin Number
RTSI Trigger (0..5)	PXI Trigger (0..5)	B16, A16, A17, A18, B18, C18
RTSI Trigger (6)	PXI Star	D17
RTSI Clock	PXI Trigger (7)	E16
Reserved	LBR (7, 8, 10, 11, 12)	A3, C3, E3, A2, B2

2

Programming General-Purpose Counter/Timers

This chapter contains programming instructions for operating the circuitry on the NI 660*X* devices. The application descriptions presented in this chapter are broken into different application classes by functionality. Each application description example discusses the pertinent registers and bitfields used to program the application.

Programming the NI 660*X* devices involves writing to and reading from registers on the device. For a list of all device registers refer to Chapter 3, *Register Maps*.

The programming steps for counters and timing I/O and specific applications are explained later on in this chapter and in register descriptions. These operations are explained in terms of bitfields. A bitfield is defined as a group of contiguous bits that jointly perform a function. Using bitfields establishes an efficient mapping technique to the underlying hardware. Also, the programming style becomes very modular, so you can easily understand the functional description of every programming step.

PCI Local Bus

The NI 660*X* devices are fully compatible with the *PCI Local Bus Specification*, Revision 2.1. The PCI local bus is a high-performance, 32-bit bus with multiplexed address and data lines. The PCI system arbitrates and assigns resources through software, freeing you from manually setting switches and jumpers. You must configure the bus-related resources before attempting to execute a register-level program. Both the bus-related and data acquisition-related PCI configurations are described in the *6601/6602 User Manual*.

PCI local bus devices can be used on any PCI bus system; however, this manual only discusses IBM-compatible systems.

PCI Initialization for the IBM Compatible System

The NI 660*X* devices use the MITE Application Specific Integrated Circuit (ASIC) chip as the PCI bus interface. NI designed this ASIC specifically for data acquisition. In order for the device to operate properly this chip must be configured. Ordinarily, NI-DAQ performs this function, but if you are not using NI-DAQ, as is the case with register-level programming, then you must configure the MITE.

The initialization is done by performing the following functions:

1. Detect whether the PCI bus is present using the appropriate PCI BIOS calls. This step verifies that the PCI bus is present.
2. Using the appropriate function, scan the PCI bus for all NI 660*X* devices to access the list of available PCI devices. PCI BIOS calls are used to find PCI devices that contain the NI Vendor ID (0x1093) and a valid NI 660*X* device ID. Refer to Table 2-1 for the list of device IDs. If a device is found matching these requirements, the pertinent information should be stored in a data structure.

Table 2-1. NI 660*X* Device IDs

Device	Device ID
NI PCI-6601	0x2C60
NI PXI-6601	0x2C70
DAQCard-6601	0x2880
NI PCI-6602	0x1310
NI PXI-6602	0x1360
NI PCI-6608	0x2DB0
NI PXI-6608	0x2CC0

3. Configure the device windows of the MITE and remap the device under 1 MB in the memory map. Refer to the *Remapping the NI 660X Device* section for more information.

Remapping the NI 660*X* Device

The NI 660*X* device uses two base address registers (BARs). BAR0 points to the base address for the MITE registers, while BAR1 points to the base address of the device registers such as the TIO counter/timer chip.

The size of each BAR is 4 KB and both are mapped into memory space. Therefore, you cannot use the C language I/O space read and write functions `inp()` and `outp()` to communicate with the NI 660*X* device. Both BARs usually map above 1 MB in the memory map, so you must know how to perform memory cycles to extended memory, which is difficult under operating systems such as Windows.

The following procedure provides you with the capability to remap the device under 1 MB making the communication much simpler. If you know how to perform memory calls under Windows or your operating system, remapping the device is both unnecessary and undesirable.

The pseudocode shown below demonstrates how to remap the device below 1 MB. If you do not want to remap the device, you must still complete steps 4 and 5 to enable the device window of the MITE. All values and bit masks are 32 bits and all pseudocode functions are of the format *`function_name (address, data)`*. This example assumes the new BAR0 address is 0xd000 and BAR1 address is 0xd1000. If you are using other addresses, you must make the appropriate changes.

1. Write the address where you want to remap the MITE (BAR0) to PCI configuration space offset 0x10. Using 0xd0000 for the new BAR0 address, make the following function call.

   ```
   PCI_Config_Write (0x10, 0xd0000)
   ```

2. Write the value 0xaeae to offset 0x340 from the new MITE address.

   ```
   Mem_Write (0xd0340, 0xaeae)
   ```

3. Write the address where you want to remap the device registers (BAR1) to configuration space offset (0x14) using 0xd1000 for the new BAR1 address.

   ```
   PCI_Config_Write (0x14, 0xd1000)
   ```

4. Create the window_data_value by masking the device address. If you are not remapping the device, use the following code.

   ```
   window_data_value = ((0xFFFFFF00 AND BAR1) OR (0x8C))
   ```

   If you are remapping the device, use the following code.

   ```
   window_data_value = ((0xFFFFFF00 AND 0xd1000) OR
   (0x8C))
   ```

   window_data_value = 0xd108C in this example.

5. Write the window data value to offset 0xc4 from the MITE (BAR0) address. If you are not remapping the device, use the following code.

   ```
   Mem_Write (BAR0 + 0xc4, window_data_value)
   ```

If you are remapping the device, use the following code.

```
Mem_Write (0xd00c4, window_data_value)
```

6. Write 0x0 to offset 0xf4 from the MITE (BAR0) address. If you are not remapping the device, use the following code.

```
Mem_Write (BAR0 + 0xf4, 0)
```

If you are remapping the device, use the following code.

```
Mem_Write (0xd00f4, 0)
```

Functional Overview

You can use the NI 660*X* device in the counter-based applications outlined in Table 2-2. For more detailed descriptions of each application, refer to Chapter 3, *Device Overview*, of the *6601/6602 User Manual*.

Table 2-2. Counter-Based Applications

Application Class	Application
Simple Counting	Simple event counting Gated event counting
Time Measurement	Single-period measurement Single pulse width measurement Two-signal edge-separation measurement
Simple-Pulse and Pulse-Train Generation	Single pulse generation Single triggered-pulse generation Retriggerable single-pulse generation Continuous pulse-train generation
Buffered Counting and Time Measurement	Buffered event counting (continuous) Buffered period measurement (continuous) Buffered semiperiod measurement (continuous) Buffered pulse width measurement (continuous) Buffered two-signal edge-separation measurement (continuous)
Position Measurement	Quadrature encoders

Reading Counter Values

This section explains how to read the values for disarmed and armed counters, and how to take buffered readings from counters.

Disarmed Counters

Read the *Gi SW Save Register* to determine the counter value on a disarmed counter. Use this method for single-period and single-pulse measurements.

Armed Counters

Values for armed counters can change during the register read. If disarming the counter is impractical, as with simple event counting or position measurement, read the *Gi SW Save Register* twice. If both reads are the same, their value indicates the counter value. If the two reads are different, the counter value changed during one of the register reads. In this case, read *Gi SW Save Register* a third time. The third read is the correct counter value.

Buffered Readings

During a buffered measurement, the counter must save a value and continue counting. Setting the G*i*_DMA_Enable bit in the *Gi DMA Config Register* tells the counter to save the counter contents in either the G*i* HW Save Register or the *Gi SW Save Register*. The counter uses the G*i* DMA Readbank bit in the *Gi DMA Status Register* to tell the software which register to read. When G*i* DMA Read bit is set to 0, the most recent buffered counter value is in the G*i* HW Save Register. When G*i*_DMA_Readbank is set to 1, the most recent buffered counter value is in the *Gi SW Save Register*.

Counter/Timer Functions and Examples

The GPCTs provide counter/timer functions that are improved over those available on Am9513-based DAQ devices. Event counting, period measurement, pulse generation, and pulse-train generation are examples of existing counter/timer functions the NI-TIO supports. Enhancements to the existing counter/timer functions include quadrature encoder support and the ability to perform buffered-mode operations.

Event Counting

In event-counting functions, the counter counts events on the Selected Source input following the software arm. The software arm occurs when software sets the counter arm bit in the command register. The following actions are available in event counting:

- The Selected Source increments or decrements the counter.
- The Selected Gate indicates when to start and stop counting intervals or when to save the counter contents to the save register.
- The software either reads the counter value asynchronously from the *Gi SW Save Register*, or it reads the G*i* HW Save Register or the *Gi SW Save Register* each time the hardware latches the counter value. In the latter case, interrupts notify the software that a save has occurred.
- AUX_LINE/UP_DOWN controls the direction of the counting.

Simple Event Counting

In simple event counting, the counter counts the number of pulses occurring on the Selected Source signal after the software arm. Software can read the counter contents at any time without disturbing the counting process. Figure 2-1 shows an example of simple event counting in which the counter counts five events on the Selected Source. The counter configuration is as follows:

- Source: external signal
- Gate: disabled
- Read from: *Gi SW Save Register*
- Direction: up

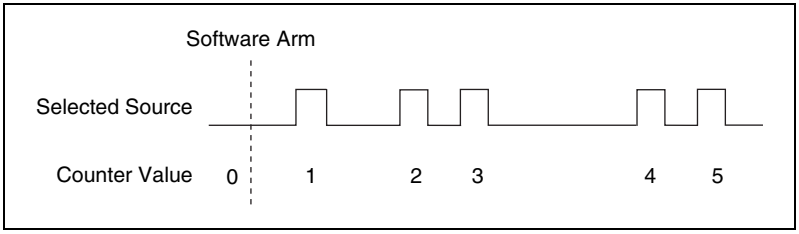

Figure 2-1. Simple Event Counting

Simple Gated-Event Counting

Simple gated-event counting is similar to simple event counting, but the counting process in simple gated-event counting is gated—halted and resumed—through the Selected Gate. When the Selected Gate is active, the counter counts pulses occurring on the Selected Source signal after the software arm. When the Selected Gate is inactive, the counter retains the current count value. Figure 2-2 shows an example of simple gated-event counting in which the gate action allows the counter to count only five of the pulses on the Selected Source. The counter configuration is as follows:

- Source: external signal
- Gate: external signal
- Read from: *Gi SW Save Register*
- Direction: up

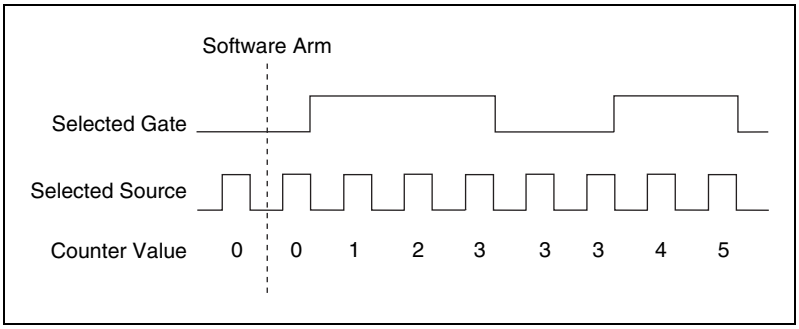

Figure 2-2. Simple Gated-Event Counting

Buffered Cumulative Event Counting

Buffered cumulative event counting is similar to simple event counting, but the Selected Gate signal in buffered cumulative event counting indicates when to save the counter value to the G*i* HW Save Register. The active Selected Gate edge latches the count value into the *Gi SW Save Register*. Counting continues uninterrupted regardless of the Selected Gate activity. An interrupt notifies the CPU after each active Selected Gate edge so that the ISR can read the result from the G*i* HW Save Register. Figure 2-3 shows cumulative event counting in which the gate action causes the G*i* HW Save Register to save the counter contents twice. The counter configuration is as follows:

- Source: external signal
- Gate: external signal
- Trigger mode: gate ignored

- Interrupt on: DMA interrupt
- Read from: determined by *Gi DMA Status Register*
- Direction: up
- Load on gate: do not reload on gate

Caution Reading the value of a counter outside the ISR may result in data loss.

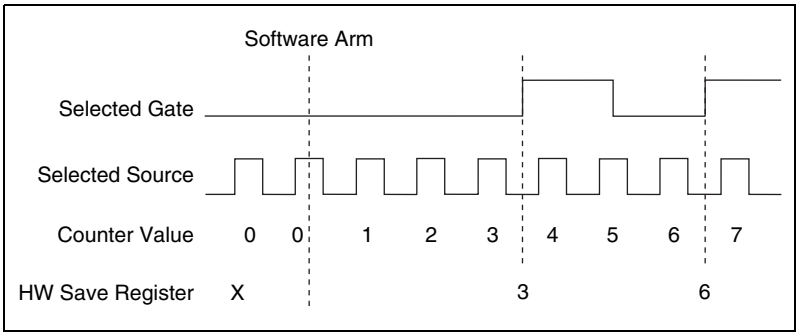

Figure 2-3. Cumulative Event Counting

Buffered Noncumulative Event Counting

Buffered noncumulative event counting is similar to simple event counting, but buffered noncumulative event counting has multiple counting intervals. The Selected Gate signal indicates the boundary between consecutive counting intervals. The counter counts the number of pulses occurring on the Selected Source signal after the software arm. Each active edge of the Selected Gate signal latches the count value for the current counting interval into either the G*i* HW Save Register or the *Gi SW Save Register* and reloads the counter with the initial value to begin the next counting interval. An interrupt notifies the CPU after each counting interval so that the interrupt software can read the result from the G*i* HW Save Register. Figure 2-4 shows buffered noncumulative event counting with two counting intervals. Three events are counted in each of the two counting intervals. The counter configuration is as follows:

- Source: external signal
- Gate: external signal
- Trigger mode: gate ignored
- Load on: gate
- Interrupt: on gate

- Read from: determined by *Gi DMA Status Register*
- Direction: up
- Load on gate: reload on gate

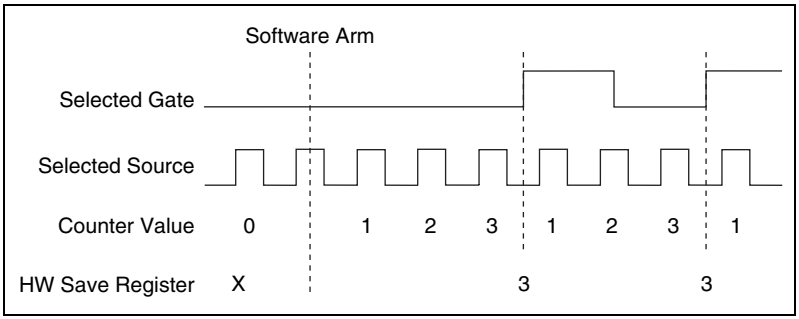

Figure 2-4. Buffered Noncumulative Event Counting

Position Measurement

In position measurement functions, the counter tracks the position of a quadrature encoder.

Relative Position Measurement

In relative position sensing, the counter tracks the relative position of the position encoder. Two types of events are possible: movement in the positive direction and movement in the negative direction. When Channel A leads Channel B, the counter increments. When Channel B leads Channel A, the counter decrements. The number of increments and decrements per cycle depends on the type of encoding: X1, X2, or X4. On NI-TIO counters, Channel A is hardwired to the default counter source pin, Channel B is hardwired to the AUX_LINE pin, and Channel Z is hardwired to the gate pin.

The software initially loads the counter with a value corresponding to the initial position of the object. Upon reaching terminal count (TC), the counter rolls over. You can obtain the relative object position at any time by asynchronously reading the counter value. Figure 2-5 shows an example of relative position sensing. The counter configuration is as follows:

- Encoder counting mode: X4
- Source select: internal timebase
- Read from: *Gi SW Save Register*

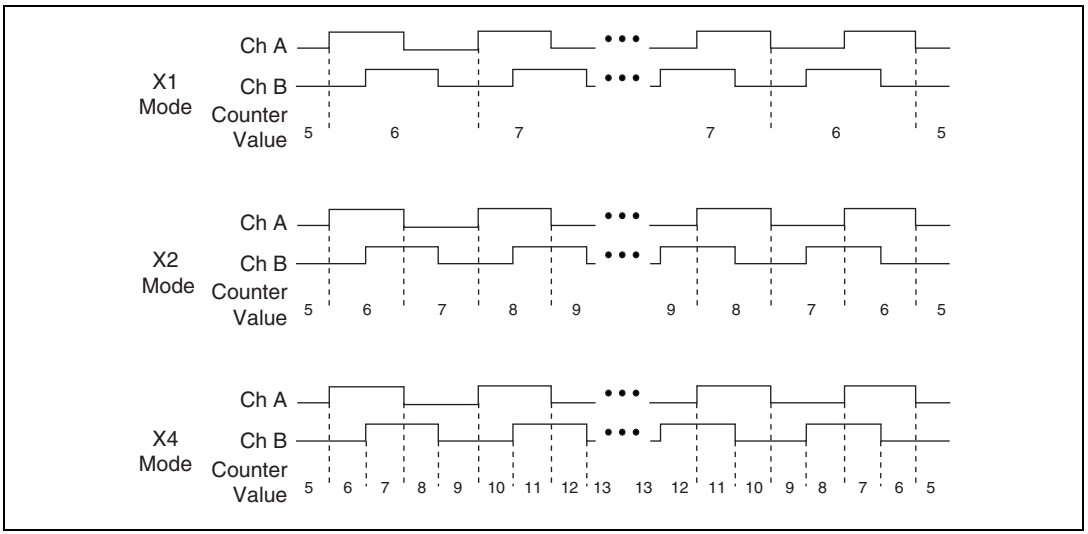

Figure 2-5. Relative Position Sensing

Time Measurement

In time measurement functions, the counter uses the Selected Source as a timebase to measure the time interval between events on the Selected Gate signal. The following actions are available in time measurement:

- Rising edges on the Selected Source can increment or decrement the counter during the measurement interval.
- Counting can begin and end on any two of the Selected Gate edges—rising, falling, or both.
- The G*i* HW Save Register or the *Gi SW Save Register* can save the counter value upon completion of the measurement.

Single-Period Measurement

In single-period measurement, the counter uses the Selected Source to measure the period of the signal present on the Selected Gate input. The counter counts the number of rising edges occurring on the Selected Source between two active edges of the Selected Gate. At the completion of the period interval for the Selected Gate, the *Gi SW Save Register* contains the last counter value. Figure 2-6 shows a single-period measurement in which the period of the Selected Gate is five Selected Source rising edges. The counter configuration is as follows:

- Source: internal clock
- Gate: external signal

- Gate mode: falling edge
- Second gate: selected gate
- Second gate mode: on
- Second gate polarity: noninverted
- Count once: true

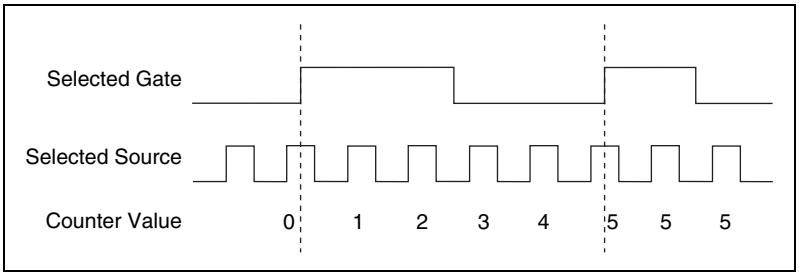

Figure 2-6. Single-Period Measurement

Single Pulse Width Measurement

In single pulse-width measurement, the counter uses the Selected Source to measure the pulse width of the signal present on the Selected Gate input. The counter counts the number of rising edges occurring on the Selected Source while the Selected Gate signal remains in an active state. At the completion of the pulse-width interval for the Selected Gate, the *Gi SW Save Register* retains the counter value for software read. Figure 2-7 shows a single pulse-width measurement in which the pulse width of the Selected Gate is five Selected Source rising edges. The counter configuration is as follows:

- Source: internal
- Gate: external signal
- Gate mode: level
- Gate polarity: inverted
- Second gate: selected gate
- Second gate mode: on
- Second gate polarity: inverted
- Count once: true

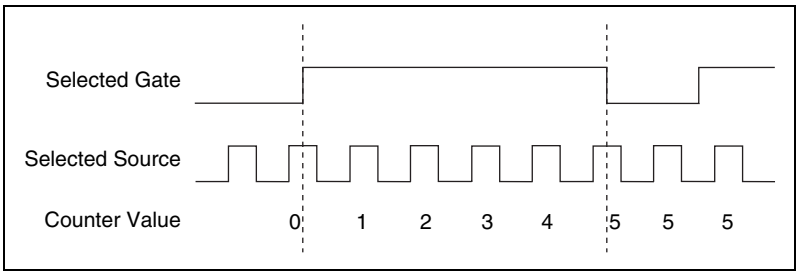

Figure 2-7. Single Pulse-Width Measurement

Buffered Period Measurement

Buffered period measurement is similar to single-period measurement, except that buffered period measurement takes measurements for multiple periods. The counter uses the Selected Source to measure the time interval between two active edges of the signal present on the Selected Gate input, counting the number of rising edges that occur on the Selected Source between each pair of active edges of the Selected Gate. At the completion of each period interval for the Selected Gate, the G*i* HW Save Register or the *Gi SW Save Register* latches the counter value for software read. An interrupt notifies the CPU after each period so that the interrupt software can read the value in the G*i* HW Save Register. Figure 2-8 shows two periods of a buffered period measurement in which the period is three Selected Source rising edges. The counter configuration is as follows:

- Source: internal clock
- Gate: external signal
- Gate mode: falling edge
- Second gate: selected gate
- Second gate mode: on
- Second gate polarity: non-inverted
- Read from: determined by *Gi DMA Status Register*
- Interrupt on: DMA interrupt
- Count once: false

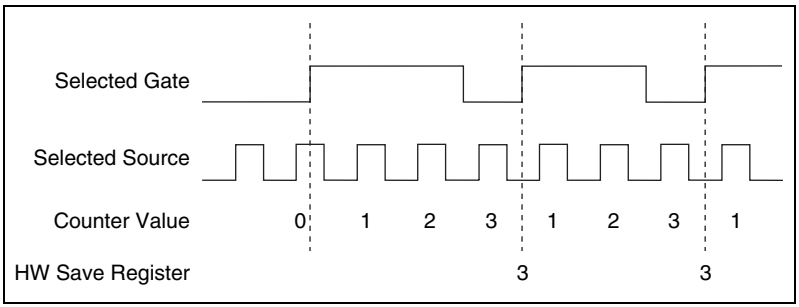

Figure 2-8. Buffered Period Measurement

Buffered Pulse Width Measurement

Buffered pulse-width measurement is similar to single pulse-width measurement, but buffered pulse-width measurement takes measurements over multiple pulses. The counter uses the Selected Source to measure the pulse width of the signal present on the Selected Gate input, counting the number of rising edges that occur on the Selected Source while the Selected Gate remains in an active state. At the completion of each pulse width interval for the Selected Gate, the G*i* HW Save Register or the *Gi SW Save Register* latches the counter value for software read. An interrupt notifies the CPU after each period so that the interrupt software can read the value in the G*i* HW Save Register. Figure 2-9 shows two pulse widths of a buffered pulse-width measurement in which the first pulse width is three Selected Source rising edges and the second pulse width is two Selected Source rising edges. The counter configuration is as follows:

- Source: internal
- Gate: external signal
- Gate mode: level
- Gate polarity: inverted
- Second gate: selected gate
- Second gate mode: on
- Second gate polarity: inverted
- Read from: determined by *Gi DMA Status Register*
- Interrupt on: DMA interrupt
- Count once: false

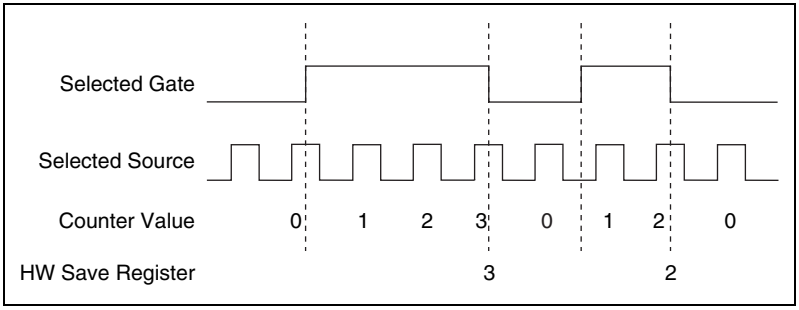

Figure 2-9. Buffered Pulse-Width Measurement

Pulse Generation

In pulse generation functions, the counter generates a single pulse of specified duration following the software arm. The software arm occurs when software sets the counter arm bit in the command register. The following actions are available in pulse generation:

- The counter uses the Selected Source as a timebase to generate the pulse.
- The user specifies the pulse parameters in terms of periods of the Selected Source input.
- The Selected Gate can serve as a trigger signal to generate a pulse after the first active gate edge or after each active gate edge.

Single Pulse Generation

The single pulse generation function generates a single pulse with programmable delay and programmable pulse width following the software arm. Because the counter uses the Selected Source as a timebase to generate the pulse, specify the pulse parameters in terms of periods of the Selected Source input. Software implements pulse generation by loading the delay value into the counter, loading the pulse-width value into the load register, and programming the counter output G_OUT to change states on counter TC. Figure 2-10 shows the generation of a single pulse with a pulse delay of four and a pulse width of three. The counter configuration is as follows:

- Source: internal
- Gate: none
- Output mode: toggle on TC
- Direction: down

- Count once: true
- Load on: TC

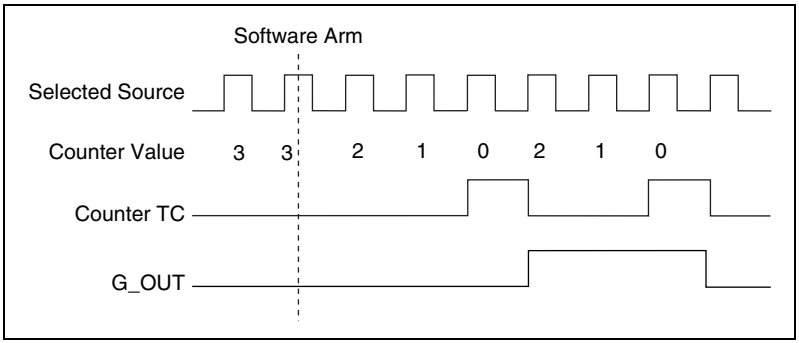

Figure 2-10. Single Pulse Generation

Pulse-Train Generation

In the pulse-train generation functions, the counter generates a continuous stream of pulses of specified interval and duration following the software arm and an optional hardware trigger. The software arm occurs when software sets the counter arm bit in the NI-TIO register. The following actions are available in pulse-train generation:

- The pulse parameters can be specified in terms of periods of the Selected Source input.
- The Selected Gate input can serve as a trigger signal to generate a stream of pulses only after the active gate edge occurs.

Continuous Pulse-Train Generation

This function generates a sequence of pulses with programmable pulse interval and pulse width. Because the counter uses the Selected Source as a timebase to generate the pulses, specify the programmable parameters in terms of periods of the Selected Source input. Pulse-train generation is implemented in software by loading the pulse parameters into the counter and load registers, and by programming the counter to switch load registers on every counter TC. Figure 2-11 shows the generation of three pulses with a delay from trigger of three, a pulse interval of four, and a pulse width of three. The counter configuration is as follows:

- Source: internal
- Gate: none
- Output mode: toggle on TC

- Direction: down
- Count once: false
- Load on: TC
- Trigger mode: only gate edge starts
- Reload source switching: true

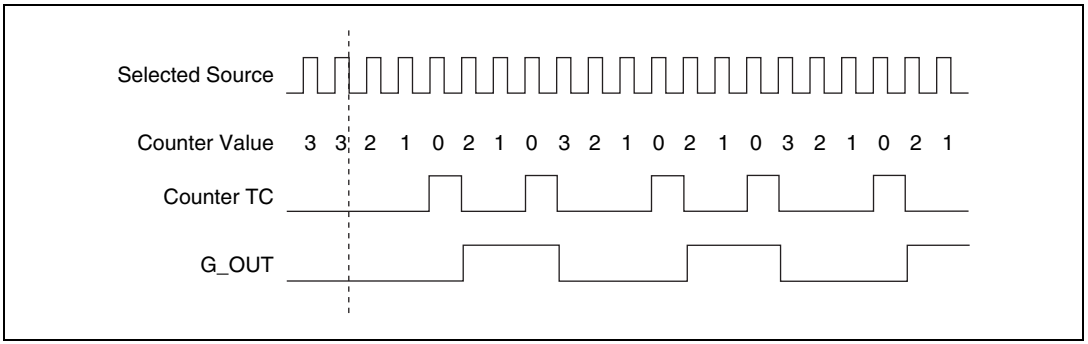

Figure 2-11. Continuous Pulse-Train Generation

Single Triggered Pulse Generation

Single triggered pulse generation is similar to single pulse generation; the Selected Gate in single triggered pulse generation provides a trigger function. An active Selected Gate edge following the software arm causes the counter to generate a single pulse with programmable delay and programmable pulse width. Specify the programmable parameters in terms of periods of the Selected Source input. Single triggered pulse generation is implemented in software by loading the delay value into the counter, loading the pulse-width value into the load register, programming the counter output G_OUT to change states on counter TC, and configuring the Selected Gate as the trigger signal. Figure 2-12 shows the generation of a single pulse with a pulse delay of four and a pulse width of three. The counter configuration is as follows:

- Source: internal
- Gate: external signal
- Output mode: toggle on TC
- Direction: down
- Count once: true
- Load on: TC
- Trigger mode: gate edge only starts

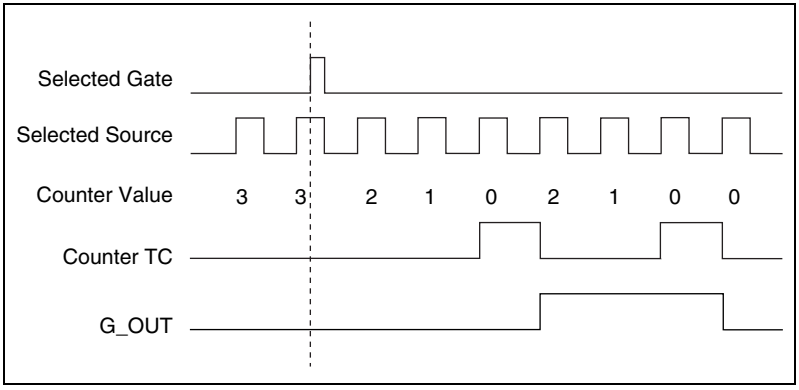

Figure 2-12. Single Triggered Pulse Generation

3

Register Maps

This chapter contains NI-TIO and MITE register maps and register descriptions.

NI-TIO

The NI-TIO (0) location is PCI BAR1, offset 0x0. The NI-TIO (1) location, which is only on the NI 6602 and NI 6608 devices, is PCI BAR1, offset 0x800.

Register Map and Descriptions

This section includes a register map and a bit-by-bit description of each register in the NI-TIO. Table 3-1 provides the register name, the register address offset from the device base address, BAR1, the type of register—read-only or write-only, and the size of the register in bits. Registers are grouped in the table by offset, and the register descriptions following the table are organized alphabetically.

Table 3-1. TIO Register Address Map

Register Name	Offset (Hex)	Type	Size
G0 Interrupt Acknowledge	0x004	Write-only	16-bit
G0 Status Register	0x004	Read-only	16-bit
G1 Interrupt Acknowledge	0x006	Write-only	16-bit
G1 Status Register	0x006	Read-only	16-bit
G01 Status Register	0x008	Read-only	16-bit
G0 Command Register	0x00C	Write-only	16-bit
G1 Command Register	0x00E	Write-only	16-bit
G0 HW Save Register	0x010	Read-only	32-bit
G1 HW Save Register	0x014	Read-only	32-bit
G0 SW Save Register	0x018	Read-only	32-bit

Table 3-1. TIO Register Address Map (Continued)

Register Name	Offset (Hex)	Type	Size
G1 SW Save Register	0x01C	Read-only	32-bit
G0 Mode Register	0x034	Write-only	16-bit
G01 Joint Status 1 Register	0x036	Read-only	16-bit
G1 Mode Register	0x036	Write-only	16-bit
G0 Load A Register	0x038	Write-only	32-bit
G01 Joint Status 2 Register	0x03A	Read-only	16-bit
G0 Load B Register	0x03C	Write-only	32-bit
G1 Load A Register	0x040	Write-only	32-bit
G1 Load B Register	0x044	Write-only	32-bit
G0 Input Select Register	0x048	Write-only	16-bit
G1 Input Select Register	0x04A	Write-only	16-bit
G0 Autoincrement Register	0x088	Write-only	16-bit
G1 Autoincrement Register	0x08A	Write-only	16-bit
G01 Joint Reset Register	0x090	Write-only	16-bit
G0 Interrupt Enable	0x092	Write-only	16-bit
G1 Interrupt Enable	0x096	Write-only	16-bit
G0 Counting Mode Register	0x0B0	Write-only	16-bit
G1 Counting Mode Register	0x0B2	Write-only	16-bit
G0 Second Gate Register	0x0B4	Write-only	16-bit
G1 Second Gate Register	0x0B6	Write-only	16-bit
G0 DMA Config Register	0x0B8	Write-only	16-bit
G0 DMA Status Register	0x0B8	Read-only	16-bit
G1 DMA Config Register	0x0BA	Write-only	16-bit
G1 DMA Status Register	0x0BA	Read-only	16-bit
G2 Interrupt Acknowledge	0x104	Write-only	16-bit
G2 Status Register	0x104	Read-only	16-bit

Table 3-1. TIO Register Address Map (Continued)

Register Name	Offset (Hex)	Type	Size
G3 Interrupt Acknowledge	0x106	Write-only	16-bit
G3 Status Register	0x106	Read-only	16-bit
G23 Status Register	0x108	Read-only	16-bit
G2 Command Register	0x10C	Write-only	16-bit
G3 Command Register	0x10E	Write-only	16-bit
G2 HW Save Register	0x110	Read-only	32-bit
G3 HW Save Register	0x114	Read-only	32-bit
G2 SW Save Register	0x118	Read-only	32-bit
G3 SW Save Register	0x11C	Read-only	32-bit
G2 Mode Register	0x134	Write-only	16-bit
G23 Joint Status 1 Register	0x136	Read-only	16-bit
G3 Mode Register	0x136	Write-only	16-bit
G2 Load A Register	0x138	Write-only	32-bit
G23 Joint Status 2 Register	0x13A	Read-only	16-bit
G2 Load B Register	0x13C	Write-only	32-bit
G3 Load A Register	0x140	Write-only	32-bit
G3 Load B Register	0x144	Write-only	32-bit
G2 Input Select Register	0x148	Write-only	16-bit
G3 Input Select Register	0x14A	Write-only	16-bit
G2 Autoincrement Register	0x188	Write-only	16-bit
G3 Autoincrement Register	0x18A	Write-only	16-bit
G23 Joint Reset Register	0x190	Write-only	16-bit
G2 Interrupt Enable	0x192	Write-only	16-bit
G3 Interrupt Enable	0x196	Write-only	16-bit
G2 Counting Mode Register	0x1B0	Write-only	16-bit
G3 Counting Mode Register	0x1B2	Write-only	16-bit

Table 3-1. TIO Register Address Map (Continued)

Register Name	Offset (Hex)	Type	Size
G3 Second Gate Register	0x1B6	Write-only	16-bit
G2 Second Gate Register	0x1B4	Write-only	16-bit
G2 DMA Config Register	0x1B8	Write-only	16-bit
G2 DMA Status Register	0x1B8	Read-only	16-bit
G3 DMA Config Register	0x1BA	Write-only	16-bit
G3 DMA Status Register	0x1BA	Read-only	16-bit
Reset Control Register	0x700	Write-only	32-bit
Chip Signature Register	0x700	Read-only	32-bit
Clock Config Register	0x73C	Write-only	32-bit
Global Interrupt Status Register	0x754	Read-only	32-bit
DMA Configuration Register	0x76C	Write-only	32-bit
Global Interrupt Config Register	0x770	Write-only	32-bit

Chip Signature Register

Address Offset: 0x73C

Type: Read

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
0	0	0	0	Version3	Version2	Version1	Version0

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0

Bit	Name	Description
27–24	Version<3..0>	The revision of the chip.

Clock Config Register

Address Offset: 0x73C

Type: Write

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
0	0	Counter_ Swap	0	0	0	0	0

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Bit	Name	Description
21	Counter_Swap	Setting this bit swaps the dedicated counter pins (SRC, GATE, UP/DOWN, OUT) between counter locations 0 through 3 and 4 through 7. This setting allows two NI-TIO ASICs to be configured to use all eight dedicated counter locations. The addressing of the counters remains the same. This setting also applies to dedicated pins, such as the Quadrature and Output. The mapping is as follows: 0 <-> 4 1 <-> 5 2 <-> 6 3 <-> 7

DMA Configuration Register

Address Offset: 0x76C

Type: Write

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
DMA_3_ Reset	0	0	DMA_3_ Select_4	DMA_3_ Select_3	DMA_3_ Select_2	DMA_3_ Select_1	DMA_3_ Select_0

23	22	21	20	19	18	17	16
DMA_2_ Reset	0	0	DMA_2_ Select_4	DMA_2_ Select_3	DMA_2_ Select_2	DMA_2_ Select_1	DMA_2_ Select_0

15	14	13	12	11	10	9	8
DMA_1_ Reset	0	0	DMA_1_ Select_4	DMA_1_ Select_3	DMA_1_ Select_2	DMA_1_ Select_1	DMA_1_ Select_0

7	6	5	4	3	2	1	0
DMA_0_ Reset	0	0	DMA_0_ Select_4	DMA_0_ Select_3	DMA_0_ Select_2	DMA_0_ Select_1	DMA_0_ Select_0

Bits	Name	Description
31, 23, 15, 7	DMA_<3..0>_Reset	Direct Memory Access Reset. Writing a one to this bit resets the toggle flip flop used to track 16-bit DMA accesses. The toggle flip flop points the DACK to the proper data lane. This bit should be set as part of initializing DMA. The bit clears automatically after a write.
28–24	DMA_3_Select_<4..0>	
20–16	DMA_2_Select_<4..0>	

12–8 DMA_1_Select_<4..0>

4–0 DMA_0_Select_<4..0>

These fields are used to select which internal DMA resource is routed to the corresponding DRQ pin. The mapping is as follows:

0—Counter 0
1—Counter 1
2—Counter 2
3—Counter 3
31—Nothing selected; DRQ pin goes to a high-impedance state; reset state

G0 Interrupt Enable Register, G2 Interrupt Enable Register

Address Offsets: 0x092 (G0), 0x192 (G2)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	G0_TC_ Interrupt_ Enable	0	0	0	0	0	0

Bit	Name	Description
6	G0_TC_Interrupt_Enable	G0 Terminal Count Interrupt Enable. Setting this bit allows a terminal count interrupt to assert the counter interrupt.

G01 Joint Reset Register, G23 Joint Reset Register

Address Offsets: 0x090 (G01), 0x190 (G23)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	G1_Reset	G0_Reset	0	0

Bits	Name	Description
3, 2	G*i*_Reset	Writing this strobe bit to 1 resets the counter. This bit automatically clears, and the counter is disarmed.

G01 Joint Status 1 Register

Address Offset: 0x036

Type: Read

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
X	X	X	Serial_In_ Progress	X	X	X	X

7	6	5	4	3	2	1	0
X	X	X	X	G1_Gate	G0_Gate	G1_Bank	G0_Bank

Bits	Name	Description
12	Serial_In_Progress	This bit indicates that hardware serial-to-parallel conversion is in progress.
3, 2	G*i*_Gate	This bit reads the state of the selected gate signal after polarity selection.
1, 0	G*i*_Bank	This bit indicates which bank of A and B registers the counter is using. 0—Bank X 1—Bank Y

G01 Joint Status 2 Register, G23 Joint Status 2 Register

Address Offsets: 0x03A (G01), 0x13A (G23)

Type: Read

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
G1_ Permanent_ Stale_Data	G0_ Permanent_ Stale_Data	G1_HW_ Save	G0_HW_ Save	X	X	X	X

7	6	5	4	3	2	1	0
X	X	X	X	X	X	G1_Output	G0_Output

Bits	Name	Description
15, 14	G*i*_Permanent_Stale_Data	This bit is set if the G*i*_Stale_Data bit, located on the *G01 Status Register, G23 Status Register*, is set at any time during counter operation. A counter reset clears this bit.
13, 12	G*i*_HW_Save	Hardware Save. This bit indicates that it latched valid data and is ready to read.
1, 0	G*i*_Output	This bit allows the counter output state to be read back. The read occurs after the polarity selection.

G01 Status Register, G23 Status Register

Address Offset: 0x008

Type: Read

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
X	X	G1_TC_ Error	G0_TC_ Error	G1_No_ Load_ Between_ Gates	G0_No_ Load_ Between_ Gates	G1_Armed	G0_Armed

7	6	5	4	3	2	1	0
G1_Stale_ Data	G0_Stale_ Data	G1_Next_ Load_ Source	G0_Next_ Load_ Source	G1_ Counting	G0_ Counting	G1_Save	G0_Save

Bits	Name	Description
13, 12	G*i*_TC_Error	Terminal Count Error. This bit sets when two terminal count interrupts occur without an acknowledgment. The G*i*_TC_Error_Confirm bit, located on the *G0 Interrupt Acknowledge Register, G2 Interrupt Acknowledge Register*, clears this bit.
11, 10	G*i*_No_Load_Between_Gates	This bit indicates that a load did not occur between the selected gate edges. The G*i*_Load, located on the *Gi Command Register*, or G*i*_Reset bit, located on the *G01 Joint Reset Register, G23 Joint Reset Register*, clears this bit.
9, 8	G*i*_Armed	This bit sets when the counter is armed.
7, 6	G*i*_Stale_Data	This error bit sets if two relevant GATE edges occur without a SRC edge between them. Because reaction to GATE edges depends on the SRC edges, it may have generated incorrect data. The synchronous counting mode is one way to prevent this error. In this mode, the Stale bit updates on each GATE edge and reflects the state of the G*i* HW Save Register. The G*i*_Permanent_Stale_Data bit,

		located in the *G01 Joint Status 2 Register, G23 Joint Status 2 Register*, records the error until it is cleared.
5, 4	G*i*_Next_Load_Source	This bit indicates which load bank to load from next. 0—Load register A 1—Load register B
3, 2	G*i*_Counting	G*i*_Counting is asserted when the counter is armed and enabled for counting. It reflects the state of the terms used for GATE or TC disabling the counter. When the counter is not armed, the bit is ignored.
1, 0	G*i*_Save	G*i*_Save indicates the state of the *Gi SW Save Register*. It sets when the *Gi SW Save Register* is latched for reading.

G1 Interrupt Enable Register, G3 Interrupt Enable Register

Address Offsets: 0x096 (G1), 0x196 (G3)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	0	0	0	0	G1_TC_ Interrupt_ Enable	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Bit	Name	Description
9	G1_TC_Interrupt_Enable	G1 Terminal Count Interrupt Enable. Setting this bit allows a terminal count interrupt to assert the counter interrupt.

G23 Joint Status 1 Register

Address Offset: 0x136

Type: Read

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
X	X	X	X	G3_Gate	G2_Gate	G3_Bank	G2_Bank

Bits	Name	Description
3, 2	G*i*_Gate	This bit reads the state of the selected gate signal after polarity selection.
1, 0	G*i*_Bank	This bit indicates which bank of A and B registers the counter is using. 0—Bank X 1—Bank Y

G*i* Autoincrement Register

Address Offset: 0x088, 0x0A, 0x188, 0x18A

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
G*i*_Autoincrement_7	G*i*_Autoincrment_6	G*i*_Autoincrement_5	G*i*_Autoincrement_4	G*i*_Autoincrement_3	G*i*_Autoincrement_2	G*i*_Autoincrement_1	G*i*_Autoincrement_0

Bits	Name	Description
7–0	G*i*_Autoincrement_<7..0>	This field holds a fixed value that is added to the Load A register after a counter reload, so the load value is increased each time. This functionality is included to support equivalent time sampling through pulse generation.

Note The *G01 Joint Reset Register, G23 Joint Reset Register* did not reset this register, so it should be programmed to zero.

G*i* Command Register

Address Offsets: 0x00C (G0), 0x00E (G1), 0x10C (G2), 0x10E (G3)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
G*i*_Disarm_Copy	G*i* _Save_Trace_Copy	G*i*_Arm_Copy	G*i*_Bank_Switch_Enable	G*i*_Bank_Switch_Mode	G*i*_Bank_Switch_Start	G*i*_Little_Big_Endian	G*i*_Synchronized_Gate

7	6	5	4	3	2	1	0
G*i*_Write_Switch	G*i*_Up/Down_1	G*i* _Up/Down_0	G*i*_Disarm	0	G*i*_Load	G*i*_Save_Trace	G*i*_Arm

Bits	Name	Description
15	G*i*_Disarm_Copy	Setting this bit disarms the other counter in the pair. The copy bits allow a single write to perform a command on both counters in the pair.
14	G*i*_Save_Trace_Copy	This bit controls the *Gi SW Save Register*. When G*i*_Save_Trace_Copy is clear, the *Gi SW Save Register* is open and tracing the counter. When set, the *Gi SW Save Register* goes into the latched mode after the next clock edge to assure valid data was latched. The latching process is complete when the G*i*_Save bit, located on the *G01 Status Register, G23 Status Register*, is set. The copy bits allow a single write to perform a command on both counters in the pair.
13	G*i*_Arm_Copy	Setting this bit arms the other counter in the pair. The counter remains armed until disabled in hardware or by the G*i*_Disarm_Copy bit.

12 G*i*_Bank_Switch_Enable

Setting this bit enables bank switching. When the counter is armed, this bit enables switching between the two banks of A and B Load registers, referred to as X and Y. If the counter is disarmed, this bit selects the bank written when accessing the Load A and Load B registers.

0—Writes Bank X
1—Writes Bank Y

11 G*i*_Bank_Switch_Mode

If bank switching is enabled, this bit selects the source that causes a bank switch to occur during operation.

0—Gate
1—Software

10 G*i*_Bank_Switch_Start

Setting this bit causes a bank switch on the condition determined by the G*i*_Bank_Switch_Mode bit. The bit automatically clears after the bank switch.

9 G*i*_Little_Big_Endian

When using automatic interrupt acknowledgement, this bit determines which segment of the load or save register triggers the acknowledgement.

0—Low register, bits 15–0
1—High register, bits 31–16

8 G*i*_Synchronize_Gate

Setting this bit synchronizes the GATE signal to the counter clock. This bit should be enabled, unless the gate signal is definitely synchronized to the SRC and there is sufficient setup time.

7 G*i*_Write_Switch

This bit controls load register writes to the inactive bank. When the bit is clear, writes to Load A always change A. When the bit is set, writes to Load A are directed to the inactive bank.

6, 5	G*i*_Up/Down_<0..1>	This bitfield determines the counting direction. 0—Always down 1—Always up 2—Hardware selected based on the UP/DOWN I/O pin: 0 = down, 1 = up 3—Hardware selected based on the internal GATE, 0 = down, 1 = up
4	G*i*_Disarm	Setting this bit disarms the counter.
2	G*i*_Load	This strobe bit uses software to load the initial counter value. When the bit is asserted, the selected load register (A or B) transfers to the counter.
1	G*i*_Save_Trace	This bit controls the *Gi SW Save Register*. When G*i*_Save_Trace is clear, the *Gi SW Save Register* is open and tracing the counter. When set, the register goes into the latched mode after the next clock edge to assure valid data was latched. The latching process is complete when the G*i*_Save bit, located on the *G01 Status Register, G23 Status Register*, is set.
0	G*i*_Arm	Setting this bit arms the counter. The counter remains armed until disabled in hardware or by the G*i*_Disarm bit.

G*i* Counting Mode Register

Address Offsets: 0x0B0 (G0), 0x0B2 (G1), 0x1B0 (G2), 0x1B2 (G3)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	G*i*_ Alternate_ Sync	G*i*_Prescale	0	0	0	0

7	6	5	4	3	2	1	0
0	G*i*_Index_ Phase_1	G*i*_Index_ Phase_0	G*i*_Index_ Mode	0	G*i*_ Counting_ Mode_2	G*i*_ Counting_ Mode_1	G*i*_ Counting_ Mode_0

Bits	Name	Description
13	G*i*_Alternate_Sync	Alternate Synchronization. When synchronizing a signal, the NI-TIO uses the falling edge of the clock when an internal timebase is chosen. For other signals, the clock may free run, in which case the selected source synchronizes the signal, and a delayed version clocks the counter. Setting the G*i*_Alternate_Sync bit adds an additional synchronization bit to the NI-TIO. When set, the rising source edge synchronizes the signal and clocks the counter, which gives a full state of setup and delays the reaction one clock. This mode must be enabled if the counter is clocked above 40 MHz, including cases in which TIMEBASE 3 is implicitly selected, such as in synchronous counting and quadrature modes.
12	G*i*_Prescale	When this bit is enabled, the high-speed counter divides the selected source by eight before clocking the counter. This mode allows signal frequency to be measured, even when the frequency is above the 80 MHz limit for the counters.

6, 5 G*i*_Index_Phase_<0..1>

The index phase field determines the state of the A and B quadrature signals and is where the index, or Z signal, is acted upon. The Z index can span multiple phases of the quadrature, but it is recommended that you reload the counter in only one of the phases.

0—A low, B low
1—A low, B high
2—A high, B low
3—A high, B high

4 G*i*_Index_Mode

G*i*_Index_Mode affects the new counting modes in the following ways:

Quadrature mode—The Z input signal, or index, can reload the counter in a particular phase. The counter reloads when this bit is set, A and B match the index phase, and the Z input is high.

Two-pulse mode—The counter reloads while Z is loaded.

Source synchronous mode—Setting this bit disables the edge detector on the source signal that requires it to be synchronous.

2, 0 G*i*_Counting_Mode_<0..2>

This bitfield adds counting modes to the NI-TIO GPCT for interfacing to encoders and other new applications.

0—One. Normal counting mode.

1—Quadrature Mode X1. This mode allows direct interfacing to quadrature encoders for position and velocity measurement. The encoder has two and sometimes three signals: signal A connects to the dedicated source pin of the counter; signal B connects to the UP/DOWN pin; and signal Z connects to the GATE pin. Signals A and B pulse with each movement and are 90 degrees out of phase to indicate direction. The Z signal reloads the counter when the G*i*_Index_Mode bit is enabled. The X1 mode increments or decrements the counter once per total phase.

2—Quadrature Mode X2. This mode counts on both edges of the A signal. Two counts per cycle are recorded.

3—Quadrature Mode X4. This mode counts on both edges of the A and B signals. Four counts per cycle are recorded.

4—Two-Pulse Mode. In this mode, a rising edge on the A signal increments the counter, and a rising edge on the B signal decrements the counter. The counter reloads while the Z signal is high and enabled.

6—Synchronous Source Mode. Several counter features, such as hardware save and load, depend on source edges to operate. However, it is sometimes necessary to measure event rates down to 0 Hz. Synchronous Source Mode runs the counter at maximum timebase, but enables it on a source edge to mimic running off that source while providing a free-running clock.

G*i* DMA Config Register

Address Offsets: 0x0B8 (G0), 0x0BA (G1), 0x1B8 (G2), 0x1BA (G3)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	G*i*_DMA_ Int	G*i*_DMA_ Write	G*i*_DMA_ Enable

Bits	Name	Description
2	G*i*_DMA_Int	Direct Memory Access Interrupt. This bit uses DMA mode to implement programmed I/O (PIO) operations. When set, the counter interrupt asserts with the counter DMA request (DRQ) signal, indicating that service is needed. If reading data, the software reads either the G*i* HW Save Register or the *Gi SW Save Register*, as determined by the G*i*_DMA_Readbank bit, located on the *Gi DMA Status Register*. Write data to the appropriate A or B register.
1	G*i*_DMA_Write	Direct Memory Access Write. This bit indicates the direction of the DMA operation. 0—DMA controller reads data from the NI-TIO save registers 1—DMA controller writes data to the NI-TIO load registers
0	G*i*_DMA_Enable	Direct Memory Access Enable. When this bit is set, the NI-TIO adds a new DMA mode for streaming counts into or out of the counters. The read mode uses both the hardware and the *Gi SW Save Register* as a two-element FIFO in order to increase the rate at which pulse or period measurements are made. When using a DMA controller (asserted DACK), the DACK reads are automatically

routed to the proper register, and data is alternately stored on gate edges. When using PIO, the G*i*_DMA_Readbank bit, located on the *Gi DMA Status Register*, indicates the register at the head of the FIFO.

For pulse generation, DMA can reload the load register when a bank switch occurs. DRQ asserts on the bank switch and clears when the last load register is written. The G*i*_Write_Switch and G*i*_Little_Big_Endian bits, located on the , determine the last load register. If G*i*_Write_Switch is clear, writes are directed to Load B, and DRQ clears on a write to the half of Load B as determined by G*i*_Little_Big_Endian. If G*i*_Write_Switch is true, the first write goes to Load B, DRQ remains asserted, and the second write goes to Load A, which clears DRQ.

G*i* DMA Status Register

Address Offsets: 0x0B8 (G0), 0xBA (G1), 0x1B8 (G2), 0x1BA (G3)

Type: Read

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
G*i*_DRQ_ Status	G*i*_DRQ_ Error	G*i*_DMA_ Readbank	X	X	X	X	X

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X

Bits	Name	Description
15	G*i*_DRQ_Status	DMA Request Status. This bit is set when the counter needs DMA service and clears automatically when the request is serviced.
14	G*i*_DRQ_Error	DMA Request Error. This bit is set when a DMA overflow error occurs on a read operation and when an underflow error occurs on a write operation. When reading, this bit sets if a save request occurs when the FIFO is full and another DMA read follows. The save request does not corrupt the data when the FIFO is received after completion. When writing, an error is set if the bank switches while DRQ is still set, which indicates that the two switches occurred before the load registers were serviced.
13	G*i*_DMA_Readbank	When implementing the PIO for read DMA, the G*i*_DMA_Readbank bit indicates which save register contains the next data to be transferred. This bit alternates with each read. 0—HW save 1—SW save

G*i* Input Select Register

Address Offsets: 0x048 (G0), 0x04A (G1), 0x148 (G2), 0x14A (G3)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
G*i*_Source_ Polarity	G*i*_Output_ Polarity	G*i*_OR_ Gate	G*i*_Gate_ Select_Load_ Source	G*i*_Gate_ Select_4	G*i*_Gate_ Select_3	G*i*_Gate_ Select_2	G*i*_Gate_ Select_1

7	6	5	4	3	2	1	0
G*i*_Gate_ Select_0	G*i*_Source_ Select_4	G*i*_Source_ Select_3	G*i*_Source_ Select_2	G*i*_Source_ Select_1	G*i*_Source_ Select_0	0	0

Bits	Name	Description
15	G*i*_Source_Polarity	Set this bit to invert the selected source signal, changing the polarity of the active edge.
14	G*i*_Output_Polarity	Set this bit to invert the counter output signal.
13	G*i*_OR_Gate	Set this bit to OR the output of the adjacent counter with the selected gate. When the counter is not counting, only the adjacent counter output is the gate. Clearing this bit removes the effect on the gate.
12	G*i*_Gate_Select_Load_Source	When this bit is set, the selected gate signal chooses the load source for the counter. An active gate level chooses Load A, while an inactive level chooses Load B. The G*i*_Gating_Mode bit, located on the *Gi Mode Register*, must be set to Level mode.
11–7	G*i*_Gate_Select_<4..0>	This bit selects the signal to serve as the gate for the counter.

0—Source Pin *i*; the source pin dedicated to this counter
1—Gate Pin *i*; the gate pin dedicated to this counter
2—Gate Pin 0, I/O Pin 38
3—Gate Pin 1, I/O Pin 34
4—Gate Pin 2, I/O Pin 30
5—Gate Pin 3, I/O Pin 26
6—Gate Pin 4, I/O Pin 22
7—Gate Pin 5, I/O Pin 18
8—Gate Pin 6, I/O Pin 14
9—Gate Pin 7, I/O Pin 10
10—Next SRC; the selected source of the adjacent counter
11—RTSI 0
12—RTSI 1
13—RTSI 2
14—RTSI 3
15—RTSI 4
16—RTSI 5
17—RTSI 6
20—Next Out; the counter of the adjacent counter
30—Logic low
31—Logic low

6–2 G*i*_Source_Select_<4..0>

This bit selects the signal the counter uses as the source.

0—Timebase 1; the internal timebase (20 MHz)
1—Source Pin *i*; the source pin dedicated to this counter
2—Source Pin 0, I/O Pin 39
3—Source Pin 1, I/O Pin 35
4—Source Pin 2, I/O Pin 31
5—Source Pin 3, I/O Pin 27
6—Source Pin 4, I/O Pin 23
7—Source Pin 5, I/O Pin 19
8—Source Pin 6, I/O Pin 15
9—Source Pin 7, I/O Pin 11
10—Next Gate; the selected gate of the adjacent counter
11—RTSI 0
12—RTSI 1
13—RTSI 2
14—RTSI 3
15—RTSI 4
16—RTSI 5
17—RTSI 6
18—Timebase 2; the internal timebase (100 KHz)

19—Next TC; the adjacent counter's terminal count
30—Timebase 3; the internal timebase (the maximum frequency)
31—Logic low

G0 Interrupt Acknowledge Register, G2 Interrupt Acknowledge Register

Address Offsets: 0x004 (G0), 0x104 (G2)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	G*i*_TC_ Interrupt_ Ack	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	G*i*_TC_ Error_ Confirm	0	0	0	0	0	0

Bits	Name	Description
14	G*i*_TC_Interrupt_Ack	Write a 1 to G*i*_TC_Interrupt_Ack to clear the *Gi Status Register* G*i*_TC_Status bit, and the TC status interrupt, if it was enabled in the *G0 Interrupt Enable Register, G2 Interrupt Enable Register* or the *G1 Interrupt Enable Register, G3 Interrupt Enable Register*. This bit returns to 0, immediately after the write.
6	G*i*_TC_Error_Confirm	Write a 1 to G*i*_TC_Error_Confirm to clear the TC error condition associated with the *G01 Status Register, G23 Status Register* G*i*_TC_Error bit. This bit returns to 0 immediately after the write.

G1 Interrupt Acknowledge Register, G3 Interrupt Acknowledge Register

Address Offsets: 0x006 (G1), 0x106 (G3)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	G*i*_TC_ Interrupt_ Ack	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0		0	0	0	G*i*_TC_ Error_ Confirm	0	0

Bits	Name	Description
14	G*i*_TC_Interrupt_Ack	Write a 1 to G*i*_TC_Interrupt_Ack to clear the *Gi Status Register* G*i*_TC_Status bit, and the TC status interrupt, if it was enabled in the *G0 Interrupt Enable Register, G2 Interrupt Enable Register* or the *G1 Interrupt Enable Register, G3 Interrupt Enable Register*. This bit returns to 0, immediately after the write.
6	G*i*_TC_Error_Confirm	Write a 1 to G*i*_TC_Error_Confirm to clear the TC error condition associated with the *G01 Status Register, G23 Status Register* G*i*_TC_Error bit. This bit returns to 0 immediately after the write.

G*i* Load A Register

Address Offsets: 0x038 (G0), 0x040 (G1), 0x138 (G2), 0x140 (G3)

Type: Write

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
Load_A.31	Load_A.30	Load_A.29	Load_A.28	Load_A.27	Load_A.26	Load_A.25	Load_A.24

23	22	21	20	19	18	17	16
Load_A.23	Load_A.22	Load_A.21	Load_A.20	Load_A.19	Load_A.18	Load_A.17	Load_A.16

15	14	13	12	11	10	9	8
Load_A.15	Load_A.14	Load_A.13	Load_A.12	Load_A.11	Load_A.10	Load_A.9	Load_A.8

7	6	5	4	3	2	1	0
Load_A.7	Load_A.6	Load_A.5	Load_A.4	Load_A.3	Load_A.2	Load_A.1	Load_A.0

Bits	Name	Description
31–0	Load_A.<31..0>	The Load A field is for loading the counter with G*i*_Load, located on the *Gi Command Register*, or for reloading on gate or TC conditions. This field can alternate with Load B by using the G*i*_Reload_Source_Switching bit, located on the *Gi Mode Register*. The A and B registers are actually two banks of load registers, X and Y, which are controlled by the bank switching attributes in the *Gi Command Register*.

G*i* Load B Register

Address Offsets: 0x03C (G0), 0x044 (G1), 0x13C (G2), 0x144 (G3)

Type: Write

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
Load_B.31	Load_B.30	Load_B.29	Load_B.28	Load_B.27	Load_B.26	Load_B.25	Load_B.24

23	22	21	20	19	18	17	16
Load_B.23	Load_B.22	Load_B.21	Load_B.20	Load_B.19	Load_B.18	Load_B.17	Load_B.16

15	14	13	12	11	10	9	8
Load_B.15	Load_B.14	Load_B.13	Load_B.12	Load_B.11	Load_B.10	Load_B.9	Load_B.8

7	6	5	4	3	2	1	0
Load_B.7	Load_B.6	Load_B.5	Load_B.4	Load_B.3	Load_B.2	Load_B.1	Load_B.0

Bits	Name	Description
31–0	Load_B.<31..0>	The Load B field is for loading the counter with G*i*_Load, located on the *Gi Command Register*, or for reloading on gate or TC conditions. It can alternate with Load B by using the G*i*_Reload_Source_Switching bit in the *Gi Mode Register*. The A and B registers are actually two banks of load registers, X and Y, which are controlled by the bank switching attributes in the *Gi Command Register*.

G*i* Mode Register

Address Offsets: 0x034 (G0), 0x036 (G1), 0x134 (G2), 0x136 (G3)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
G*i*_Reload_ Source_ Switching	G*i*_Loading on_Gate	G*i*_Gate_ Polarity	G*i*_Loading_ on_TC	G*i*_Counting _Once_1	G*i*_Counting _Once_0	G*i*_Output_ Mode_1	G*i*_Output_ Mode_0

7	6	5	4	3	2	1	0
G*i*_Load_ Source_ Select	G*i*_Stop_ Mode_1	G*i*_Stop_ Mode_0	G*i*_Trigger_ Mode_for Edge_ Gate_1	G*i*_Trigger_ Mode_for_ Edge_ Gate_0	G*i*_Gate_ on_Both_ Edges	G*i*_Gating_ Mode	G*i*_Gating_ Mode

Bits	Name	Description
15	G*i*_Reload_Source_Switching	When G*i*_Gate_Select_Load_Source, located on the *Gi Input Select Register*, is 0, the G*i*_Reload_Source_Switching bit determines if the reloading always comes from the same load register (0), or if it alternates between the two (1).
14	G*i*_Loading_on_Gate	When this bit is set, the gate edge reloads the counter on the next source edge. Set the G*i*_Trigger_Mode_for_Edge_Gate to 3 to reload the counter on each selected edge. Otherwise, the counter reloads on the gate that stops the counter. You also can use G*i*_Loading_on_Gate with G*i*_Loading_on_TC.
13	G*i*_Gate_Polarity	Setting the bit inverts the gate signal, making it active low rather than the normal active high.

12 G*i*_Loading_on_TC

Loading on Terminal Count. If this bit is clear, the counter rolls over when it reaches TC. When set, the counter reloads from a Load register on TC. You also can use G*i*_Loading_on_TC with G*i*_Loading_on_Gate.

11, 10 G*i*_Counting_Once_<1..0>

This field determines whether the counter disarms when the counter stops for a hardware condition.

0—No hardware disarm
1—Disarm at the TC that stops counting
2—Disarm at the gate that stops counting
3—Disarm at the first TC or gate that stops counting

9, 8 G*i*_Output_Mode_<1..0>

G*i* Output Mode determines the counter output type.

0—Reserved
1—The counter TC is the output; it asserts for one clock at TC
2—Output toggles value on each TC
3—Output toggles on TC or gate

7 G*i*_Load_Source_Select

When the counter is disarmed, this bit determines which load register—Load A (0) or Load B (1)—loads the counter in response to a G*i*_Load. Once the counter is armed, the G*i*_Reload_Source_Switching bit determines the load source. After the counter is armed, writing this bit has no effect.

6, 5 G*i*_Stop_Mode_<1..0>

This bit determines the mode in which the hardware stops the counter.

0—Stop on gate condition
1—Stop on gate or first TC
2—Stop on gate or second TC
3—Reserved

Note To make these conditions TC only, use G*i*_Trigger_Mode_for_Edge_Gate to turn off the edge gate stopping.

4, 3 G*i*_Trigger_Mode_for_Edge_Gate_<1..0>

This field determines how a gate edge affects the counting when gating is enabled.

0—The first gate edge starts, and the next gate edge stops the counting.
1—The first gate edge stops, and the second gate edge starts the counting.
2—The gate edge that always starts the counting. TC normally stops this mode.
3—The gate not used for starting or stopping. This gate may still save, reload, or load select. Selections 0 and 1 are valid when G*i*_Stop_Mode is 0, and selection 0 through 2 are only valid for edge gating and level gating.

2 G*i*_Gate_on_Both_Edges

When G*i*_Gate_on_Both_Edges is set and G*i*_Gating_Mode is set to rising mode, the edge gating modes operate on both the rising and falling edges.

1, 0 G*i*_Gating_Mode

This field determines how the gate signal is interpreted for gating operations.

0—Gating disabled
1—Level gating
2—Rising edge gating
3—Falling edge gating

Note G*i*_Gate_Polarity can change the edge for modes 2 and 3. When gating is disabled, you can use the gate for counting direction only.

G*i* Second Gate Register

Address Offsets: 0x0B4 (G0), 0x0B6 (G1), 0x1B4 (G2), 0x1B6 (G3)

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	G*i*_Second_Gate_Polarity	0	G*i*_Second_Gate_Select_4	G*i*_Second_Gate_Select_3	G*i*_Second_Gate_Select_2	G*i*_Second_Gate_Select_1

7	6	5	4	3	2	1	0
G*i*_Second_Gate_Select_0	0	0	0	0	0	0	G*i*_Second_Gate_Mode

Bits	Name	Description
13	G*i*_Second_Gate_Polarity	Setting this bit inverts the selected second gate, changing the polarity.
11–7	G*i*_Second_Gate_Select_<4..0>	This field selects the signal used as the second gate for the counter. 0—Source Pin *i*; the source pin dedicated to this counter 1—Up/Down Pin *i*; the Up/Down pin dedicated to this counter 2—Up/Down Pin 0, I/O Pin 37 3—Up/Down Pin 1, I/O Pin 33 4—Up/Down Pin 2, I/O Pin 29 5—Up/Down Pin 3, I/O Pin 25 6—Up/Down Pin 4, I/O Pin 21 7—Up/Down Pin 5, I/O Pin 17 8—Up/Down Pin 6, I/O Pin 13 9—Up/Down Pin 7, I/O Pin 9 10—Next SRC; the selected source of the adjacent counter 11—RTSI 0 12—RTSI 1 13—RTSI 2

14—RTSI 3
15—RTSI 4
16—RTSI 5
17—RTSI 6
20—Next Out; the adjacent counter's output
30—Selected gate; the output of the gate selection, and not the selected gate I/O pin. For example, if the gate is set to (1) Gate Pin *i* and if G*i*_Output_Polarity bit, located on the *Gi Input Select Register*, is set, the alternate gate is an inverted Gate Pin *i* signal. This gate is useful for pulse measurement.
31—Logic low

0 G*i*_Second_Gate_Mode

The second gate feature allows one signal to start the counter and another to stop it for two-edge separation measurements. When set, G*i*_Second_Gate_Mode modifies the actual gate signal with a combination of the first and second gate. An assertion of the second gate signals asserts the counter gate, and an assertion of the gate signal deasserts the gate. You can then use this new gate signal for start and stop operations.

G*i*_Second_Gate_Mode is also useful in pulse-width measurements. The level gating mode starts counting if the counter is armed while the gate is high, so the first measurement may be too short. Using the selected gate input to the second gate and inverting both gate and second gate polarity forces a rising edge to occur before the gate asserts. As a result, the first pulse is measured.

G*i*_Second_Gate_Mode is also used for two-edge separation measurements.

G*i* Status Register

Address Offsets: 0x004 (G0), 0x006 (G1), 0x104 (G2), 0x106 (G3)

Type: Read

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
G*i*_Interrupt	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
X	X	X	X	G*i*_TC_ Status	X	X	X

Bits	Name	Description
15	G*i*_Interrupt	This counter asserts an interrupt. TC or DMA Interrupt is true and enabled.
3	G*i*_TC_Status	Terminal Count Status. This bit indicates that the counter has reached terminal count. Setting G*i*_TC_Interrupt_Ack, located on the *G0 Interrupt Acknowledge Register, G2 Interrupt Acknowledge Register* and the *G1 Interrupt Acknowledge Register, G3 Interrupt Acknowledge Register*, clears this bit. G*i*_TC_Status is useful for generating interrupts at a constant frequency.

G*i* SW Save Register

Address Offsets: 0x018 (G0), 0x01C (G1), 0x118 (G2), 0x11C (G3)

Type: Read

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
SW_ Save.31	SW_ Save.30	SW_ Save.29	SW_ Save.28	SW_ Save.27	SW_ Save.26	SW_ Save.25	SW_ Save.24

23	22	21	20	19	18	17	16
SW_ Save.23	SW_ Save.22	SW_ Save.21	SW_ Save.20	SW_ Save.19	SW_ Save.18	SW_ Save.17	SW_ Save.16

15	14	13	12	11	10	9	8
SW_ Save.15	SW_ Save.14	SW_ Save.13	SW_ Save.12	SW_ Save.11	SW_ Save.10	SW_Save.9	SW_Save.8

7	6	5	4	3	2	1	0
SW_Save.7	SW_Save.6	SW_Save.5	SW_Save.4	SW_Save.3	SW_Save.2	SW_Save.1	SW_Save.0

Bits	Name	Description
31–0	SW_Save_<31..0>	Software Save. This value is latched on the next source edge following the assertion of the G*i*_Save_Trace bit, located on the *Gi Command Register*. When the bit is clear, the counter value is transparent through the *Global Interrupt Status Register*. This register is also part of the buffer for DMA operations. When G*i* DMA Enable, located on the *Gi DMA Config Register* is set, the *Gi SW Save Register* is used as a G*i* HW Save Register. In DMA mode, the save value alternates between the G*i* HW Save Register and the *Gi SW Save Register*.

Global Interrupt Config Register

Address Offset: 0x770

Type: Write

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
Global_Int_ Enable	Global_Int_ Polarity	Cascade_ Int_Enable	0	0	0	0	0

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Bit	Name	Description
31	Global_Int_Enable	Global Interrupt Enable. Clearing this bit disables the assertion of the INT pin. When set, it allows the INT pin to assert when a condition is true.
30	Global_Int_Polarity	Global Interrupt Polarity. Setting this bit flips the polarity of the INT pin, causing it to go low on an interrupt assertion.
29	Cascade_Int_Enable	Cascade Interrupt Enable. Setting this bit allows the CAS INT pin to be routed to the INT pin in combination with the other sources. Clearing the bit prevents any

cascade interrupts. On devices with two NI-TIO ASICS, such as the NI 6602, the CAS_INT pin is connected to the second NI-TIO.

Global Interrupt Status Register

Address Offset: 0x754

Type: Read

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
Global_Int	X	Cascade_ Int	X	X	X	X	X

23	22	21	20	19	18	17	16
X	X	X	X	Counter_3_ Int	Counter_2_ Int	Counter_1_ Int	Counter_0_ Int

15	14	13	12	11	10	9	8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X

Bits	Name	Description
31	Global_Int	Global Interrupt. This bit is set when the NI-TIO is generating an interrupt.
29	Cascade_Int	Cascade Interrupt. This bit reflects the state of the CAS_INT pin. On devices with two NI-TIO ASICS, such as the NI 6602, the CAS_INT pin is connected to the second NI-TIO.
19–16	Counter_<3..0>_Int	Counter Interrupts. These bits are set when the counter is generating an interrupt. Refer to the appropriate counter register—*Gi Status Register* or *Gi DMA Status Register*—for more information.

Reset Control Register

Address Offset: 0x700

Type: Write

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
0	0	0	0				

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0
							Soft_Reset

Bit	Name	Description
0	Soft_Reset	Writing a 1 to this bit resets the chip. This bit clears automatically.

Simple Digital I/O

An 8-bit DIO port is available for simple unstrobed read and write operations.

STC Digital I/O

The 8-bit digital port provides individual control over the direction and state of each bit. This port is located on pins 0 through 7.

STC Digital I/O Registers

Table 3-2 shows the STC digital I/O register map. The table provides the register name, the register address offset from the device base address (BAR1), the type of register—read-only and write-only, and the size of the register in bits.

Registers are grouped in the table by offset, and the register descriptions following the table are organized alphabetically.

Table 3-2. STC Digital I/O Register Address Map

Register Name	Offset (Hex)	Type	Size
STC DIO Parallel Input	0x00E	Read-only	16-bit
STC DIO Output	0x014	Write-only	16-bit
STC DIO Control	0x016	Write-only	16-bit
STC DIO Serial Input	0x038	Read-only	16-bit

STC DIO Control

Address Offset: 0x016

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
0	0	0	0	Serial_SW_ Strobe	Serial_ Timebase	Serial_HW_ Enable	Serial_Start

7	6	5	4	3	2	1	0
OE_7	OE_6	OE_5	OE_4	OE_3	OE_2	OE_1	OE_0

Bits	Name	Description
11	Serial_SW_Strobe	Serial Software Strobe. When the Serial_HW_Enable bit is disabled, the Serial_SW_Strobe bit can toggle the serial clock.
10	Serial_Timebase	0—Serial clock is Timebase 1 / 24 (1.2 µs) 1—Serial clock is Timebase 2 (100 KHz)
9	Serial_HW_Enable	Serial Hardware Enable. Setting this bit enables hardware parallel-to-serial conversion. Setting this bit causes serial clock to be driven on STC DOUT 4 and serial data out to be driven on STC DOUT 0.
8	Serial_Start	Setting this bit initiates parallel-to-serial conversion of the STC DIO Output Register, starting with bit 0. It also captures serial input into the serial input register. This bit automatically clears when the conversion completes. The Serial_In_Progress bit, located on the *G01 Joint Status 1 Register*, can monitor the conversion progress.
7–0	OE_<7..0>	Output Enable. These bits control the output enable for the STC DIO lines when using I/O pins 7 through 0 if these pins are programmed as inputs in the I/O configuration registers. This setting makes the port

compatible with the DAQ-STC. The I/O configuration registers also control the output enables, in which case these bits are ignored.

STC DIO Output

Address Offset: 0x014

Type: Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
Serial_ DOUT.7	Serial_ DOUT.6	Serial_ DOUT.5	Serial_ DOUT.4	Serial_ DOUT.3	Serial_ DOUT.2	Serial_ DOUT.1	Serial_ DOUT.0

7	6	5	4	3	2	1	0
Parallel_ DOUT.7	Parallel_ DOUT.6	Parallel_ DOUT.5	Parallel_ DOUT.4	Parallel_ DOUT.3	Parallel_ DOUT.2	Parallel_ DOUT.1	Parallel_ DOUT.0

Bits	Name	Description
15–8	Serial_DOUT.<7..0>	Serial Digital Output. This field loads the output register to be shifted on the next serial start.
7–0	Parallel_DOUT.<7..0>	Parallel Digital Output. This field immediately updates the STC DOUT pins when they are enabled for output drive.

STC DIO Parallel Input

Address Offset: 0x00E

Type: Read

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
DIN.7	DIN.6	DIN.5	DIN.4	DIN.3	DIN.2	DIN.1	DIN.0

Bits	Name	Description
7–0	DIN.<7..0>	Digital Input. This field reads the state of the DAQ-STC DIO pins.

STC DIO Serial Input

Address Offset: 0x038

Type: Read

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
Serial_ DIN.7	Serial_ DIN.6	Serial_ DIN.5	Serial_ DIN.4	Serial_ DIN.3	Serial_ DIN.2	Serial_ DIN.1	Serial_ DIN.0

Bits	Name	Description
7–0	Serial_DIN.<7..0>	Serial Digital Input. These bits store the serial input data after hardware serial-to-parallel conversion completes.

I/O Connection Registers

The NI-TIO has a 40-pin I/O connector interface. These 40 bidirectional signals control most of the NI-TIO operations. The direction and output signals are programmable.

Note When the NI-TIO powers up, all I/O pins are in the common-input usage state.

Pin Usage

Table 3-3 lists output sources and common inputs for each I/O connection pin.

Table 3-3. I/O Connection Pin Usage

Pin Number	Common Input Usage	Counter Output
PFI 0	STC DIN 0	STC DOUT 0
PFI 1	STC DIN 1	STC DOUT 1
PFI 2	STC DIN 2	STC DOUT 2
PFI 3	STC DIN 3	STC DOUT 3
PFI 4	STC DIN 4	STC DOUT 4
PFI 5	STC DIN 5	STC DOUT 5
PFI 6	STC DIN 6	STC DOUT 6
PFI 7	STC DIN 7	STC DOUT 7
PFI 8	—	Counter 7 Output
PFI 9	Up/Down 7	—
PFI 10	Gate 7	Counter 7 Selected Gate
PFI 11	Source 7	Counter 7 Selected Source
PFI 12	—	Counter 6 Output
PFI 13	Up/Down 6	—
PFI 14	Gate 6	Counter 6 Selected Gate
PFI 15	Source 6	Counter 6 Selected Source
PFI 16	—	Counter 5 Output

Table 3-3. I/O Connection Pin Usage (Continued)

Pin Number	Common Input Usage	Counter Output
PFI 17	Up/Down 5	—
PFI 18	Gate 5	Counter 5 Selected Gate
PFI 19	Source 5	Counter 5 Selected Src
PFI 20	—	Counter 4 Output
PFI 21	Up/Down 4	—
PFI 22	Gate 4	Counter 4 Selected Gate
PFI 23	Source 4	Counter 4 Selected Source
PFI 24	—	Counter 3 Output
PFI 25	Up/Down 3	—
PFI 26	Gate 3	Counter 3 Selected Gate
PFI 27	Source 3	Counter 3 Selected Source
PFI 28	—	Counter 2 Output
PFI 29	Up/Down 2	—
PFI 30	Gate 2	Counter 2 Selected Gate
PFI 31	Source 2	Counter 2 Selected Source
PFI 32	—	Counter 1 Output
PFI 33	Up/Down 1	—
PFI 34	Gate 1	Counter 1 Selected Gate
PFI 35	Source 1	Counter 1 Selected Source
PFI 36	—	Counter 0 Output
PFI 37	Up/Down 0	—
PFI 38	Gate 0	Counter 0 Selected Gate
PFI 39	Source 0	Counter 0 Selected Source

I/O Registers

Table 3-4. I/O Connection Register Address Map

Register Name	Offset (Hex)	Type	Size
I/O Config Reg 0–1	0x77C	Read and Write	16-bit
I/O Config Reg 2–3	0x77E	Read and Write	16-bit
I/O Config Reg 4–5	0x780	Read and Write	16-bit
I/O Config Reg 6–7	0x782	Read and Write	16-bit
I/O Config Reg 8–9	0x784	Read and Write	16-bit
I/O Config Reg 10–11	0x786	Read and Write	16-bit
I/O Config Reg 12–13	0x788	Read and Write	16-bit
I/O Config Reg 14–15	0x78A	Read and Write	16-bit
I/O Config Reg 16–17	0x78C	Read and Write	16-bit
I/O Config Reg 18–19	0x78E	Read and Write	16-bit
I/O Config Reg 20–21	0x790	Read and Write	16-bit
I/O Config Reg 22–23	0x792	Read and Write	16-bit
I/O Config Reg 24–25	0x794	Read and Write	16-bit
I/O Config Reg 26–27	0x796	Read and Write	16-bit
I/O Config Reg 28–29	0x798	Read and Write	16-bit
I/O Config Reg 30–31	0x79A	Read and Write	16-bit
I/O Config Reg 32–33	0x79C	Read and Write	16-bit
I/O Config Reg 34–35	0x79E	Read and Write	16-bit
I/O Config Reg 36–37	0x7A0	Read and Write	16-bit

I/O Config Register, Pins A through B

Address Offset: 0x77C, 0x77E, 0x780, 0x782, 0x784, 0x786, 0x788, 0x78A, 0x78C, 0x78E, 0x790, 0x792, 0x794, 0x79A, 0x79C, 0x79E, 0x7A0

Type: Read and Write

Size: 16-bit

Bit Map:

15	14	13	12	11	10	9	8
X	A_Input_ Select_2	A_Input_ Select_1	A_Input_ Select_0	X	X	A_Output_ Select_1	A_Output_ Select_0

7	6	5	4	3	2	1	0
X	B_Input_ Select_2	B_Input_ Select_1	B_Input_ Select_0	X	X	B_Output_ Select_1	B_Output_ Select_0

Bits	Name	Description
9, 8	A_Output_Select_<1..0>	
1, 0	B_Output_Select_<1..0>	This field determines if the I/O pin is driven and what signal is driven into the pin. 0—Pin is input only; driver goes to a high-impedance state 1—Normal counter output
14–12	A_Input_Select_<2..0>	
6–4	B_Input_Select_<2..0>	This field allows each input pin to be passed through or digitally filtered to remove noise. The digital filters sample the signal using Timebase 3, and upon detecting a state change, require it to be in the same state for a minimum pulse width either internally generated or selected from a RTSI line. 0—Input signal unchanged 1—Input synchronized to Timebase 3 2—Digital filter; minimum pulse width is 100 assertions of Timebase 1 3—Digital filter; minimum pulse width is 20 assertions of Timebase 1 4—Digital filter; minimum pulse width is 10 assertions of Timebase 1

5—Digital filter; minimum pulse width is 2 assertions of Timebase 1
6—Digital filter; minimum pulse width is 2 assertions of Timebase 3

MITE

The MITE is located on PCI BAR0, offset 0x0.

To enable the BAR1 address space where the NI-TIO ASICs are located, write to the I/O Window Base Size Register 1 (IODWBSR1) on the MITE. The IODWBSR1 determines the location and size of the I/O window in the CPU address space. This window is used to access the device address space on the I/O port.

For example, if the PCI/PXI device base address range 0 is BAR0, and the card base address range 1 is BAR1, enable BAR1 with the following function calls:

```
Bar0.write32(0xC4, (physicalBAR1 & 0xFFFFFF00L) | 0x8C);

Bar0.write32(0xF4, 0);
```

I/O Window Base Size Register 1 (IOWBSR1)

Address Offset: 0xC4

Type: Read and Write

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24

23	22	21	20	19	18	17	16
BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16

15	14	13	12	11	10	9	8
BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8

7	6	5	4	3	2	1	0
WENAB	X	X	WSIZE4	WSIZE3	WSIZE2	WSIZE1	WSIZE0

Bits	Name	Description
31–12	BA_<31..8>	Base Address. These bits indicate the base address of the 8 KB device window in CPU address space. The number of bits compared is determined by the GSIZE and WSIZE bits.
7	WENAB	Window Enable. Setting this bit enables the window. Clearing this bit disables the window.
6–5	X	X should always be written with a 0, and ignored when read.
0	WSIZE	Window Size. The WSIZE bits determine the number of address bits to compare. The minimum window size is 256 bytes. The size in bytes is equal to $2^{(WSIZE + 1)}$ for WSIZE > 7.

I/O Window Control Register 1 (IOWCR1)

Address Offset: 0xF4

Type: Read and Write

Size: 32-bit

Bit Map:

31	30	29	28	27	26	25	24
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X

Bits	Name	Description
31–0	X	These bits should always be written to with a 0 and will always read back as an X.

A

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Online technical support resources include the following:
  - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at ni.com/support. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, hardware schematics and conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.
  - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting ni.com/ask. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.
- **Training**—Visit ni.com/custed for self-paced tutorials, videos, and interactive CDs. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

Prefix	Meaning	Value
μ-	micro-	10^{-6}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{9}

Symbols

–	negative of, or minus
/	per
%	percent
±	plus or minus
+	positive of, or plus

A

arm	to enable a counter to start an operation. If the application requires a trigger, an armed counter waits for the trigger to begin the operation.
ASIC	application specific integrated circuit
asynchronous	a property of an event that occurs at an arbitrary time, without synchronization to a reference clock
AUX_LINE	the second gate input for the general-purpose counter/timers on the NI-TIO. This input doubles as an up/down control input.

B

b	bit—one binary digit, either 0 or 1
B	byte—eight related bits of data, an eight-bit binary number. Also used to denote the amount of memory required to store one byte of data.
BAR	base address memory—a memory address that serves as the starting address for programmable registers. All other addresses are located by adding to the base address.
BCD	binary-coded decimal—a method of encoding numbers where each decimal digit is encoded separately
buffer	a block of memory used to store measurement results
buffered	a type of measurement in which multiple measurements are made consecutively and measurement results are stored in a buffer
bus	the group of conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected. Examples of PC buses are the AT, EISA, and PCI bus.

C

CAS_INT	cascade interrupt
clock	hardware component that provides timing for various device operations
CMOS	complementary metal-oxide semiconductor
CompactPCI	an electrical superset of the PCI bus architecture with a mechanical form factor suited for industrial applications
counter/timer	a circuit that counts external pulses or clock pulses (timing)
CPU	central processing unit—the computational and control unit of a computer that interprets and executes instructions

D

DACK	DMA controller
DAQ	data acquisition—(1) collecting and measuring electrical signals from sensors, transducers, and test probes or fixtures and inputting them to a computer for processing; (2) collecting and measuring the same kinds of electrical signals with A/D and/or DIO boards plugged into a computer, and possibly generating control signals with D/A and/or DIO boards in the same computer
DAQ-STC	Data Acquisition System Timing Controller Chip
decode	used in the context of motion encoders. The two channels of a motion encoder indicate information about movement and direction of movement of an external device. Decoding refers to extracting this information from the signals on these channels.
device	a plug-in data acquisition board, card, or pad that can contain multiple channels and conversion devices. Plug-in boards, PCMCIA cards, and DAQ devices that connect to your computer parallel port, are all examples of DAQ devices.
DIN	Digital Input
DIO	Digital Input/Output
DMA	Direct Memory Access—a method by which data can be transferred to/from computer memory from/to a device or memory on the bus while the processor does something else. DMA is the fastest method of transferring data to/from computer memory.
driver	software that controls a specific hardware device such as a DAQ device
DRQ	DMA Request
duty cycle	the percentage of the cycle in which the pulse is high

E

EEPROM	Electrically Erasable Programmable Read-Only Memory—ROM that can be erased with an electrical signal and reprogrammed
EISA	Extended Industry Standard Architecture

encode	used in the context of motion encoders. Motion encoders provide information about movement and direction of movement of an external device. The process of producing the pulses that contain this information is called encoding.

F

FIFO	First-In-First-Out—a data buffering technique that functions like a shift register where the oldest values (first in) come out first

G

G_OUT	a TC-related counter output signal that can toggle on every counter TC or directly output the counter TC signal
GATE	the signal that controls the operation of a counter. This signal may start or stop the operation of a counter, reload the counter, or save the results of a counter.
GND	ground
GPCT	General-Purpose Counter/Timer

H

hardware	the physical components of a computer system, such as the circuit boards, plug-in boards, chassis, enclosures, peripherals, cables, and so on
high-impedance	a third output state, other than high or low, in which the output is undriven
HW	hardware
HW Save Register	a register inside the NI-TIO ASIC that stores the result of a measurement
Hz	hertz—a unit of frequency. One hertz corresponds to one cycle or event per second

I

I/O	Input/Output—the transfer of data to/from a computer system involving communications channels, operator interface devices, and/or data acquisition and control interfaces
interrupt	a computer signal indicating that the CPU should suspend its current task to service a designated activity
INTERRUPT	a counter output interrupt signal routed to the MITE
interrupt level	the relative priority at which a device can interrupt
IODWBSR	I/O Device Window Base Size Register
IRQ	Interrupt Request Signal
ISA	Industry Standard Architecture
ISR	Interrupt Service Routine

K

K	denotes 1,024, or 2^{10}, used with B (Bytes) in quantifying data or computer memory

L

LabVIEW	Laboratory Virtual Instrument Engineering Workbench—an NI graphical programming application

M

m	meters
max	maximum
maximum timebase	the fastest internal timebase available on a device. For the NI 6601, the maximum timebase is 20 MHz. For the NI 6602 and NI 6608, the maximum timebase is 80 MHz.
min	minimum

MITE	a custom ASIC designed by National Instruments that implements the PCI bus interface. The MITE supports bus mastering for high speed data transfers over the PCI bus.
motion encoders	transducers that generate pulses to indicate the physical motion of a device. The most common type of motion encoders are quadrature encoders. Two-pulse encoders (also referred to as up/down encoders) are another example.
MXI	Multisystem eXtension Interface—a high-performance communication link that interconnects devices using round, flexible cables

N

NI-DAQ	NI driver software for DAQ hardware
NI-TIO	a custom ASIC developed by National Instruments that provides counter and digital I/O functionality
noise	an undesirable electrical signal—noise comes from external sources such as the AC power line, motors, generators, transformers, fluorescent lights, soldering irons, CRT displays, computers, electrical storms, welders, radio transmitters, and internal sources such as semiconductors, resistors, and capacitors. Noise corrupts signals you are trying to send or receive.

O

operating system	base-level software that controls a computer, runs programs, interacts with users, and communicates with installed hardware or peripheral devices

P

PCI	Peripheral Component Interconnect—a high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA. It is achieving widespread acceptance as a standard for PCs and work-stations; it offers a theoretical maximum transfer rate of 132 Mbytes/s.
PFI	Programmable Function Input
PIO	Programmed I/O

port	(1) a communications connection on a computer or a remote controller; (2) a digital port, consisting of lines of digital input and/or output
prescaling	the division of frequency of an input signal that is to be used as SOURCE of a counter
programmed I/O	a data transfer method in which the CPU reads or writes data as prompted by software
PXI	modular instrumentation standard based on CompactPCI developed by National Instruments with enhancements for instrumentation

Q

quadrature encoders	a motion encoder that has two channels: channels A and B. Pulses and phases of pulses on channels A and B carry information about degree and direction of movement. A third channel—channel Z—may also exist that provides a reference point for position.

R

reflection	a high-speed signal transition behaves like a wave and is reflected like a wave at an inadequately terminated endpoint. This phenomenon is referred to as reflection.
RG	Reserved Ground—pins that are marked RG on the I/O connector are no-connects if you use the SH6868-D1 shielded cable, while they are ground pins if you use the R6868 unshielded ribbon cable.
ribbon cable	a flat cable in which the conductors are side by side
RMA	Return Material Authorization
RTSI Bus	Real-Time System Integration Bus—the NI timing bus that directly connects DAQ devices, by means of connectors on top of the boards, for precise synchronization of functions

S

s	seconds
Selected Gate	the gate input on the four general-purpose counter/timers on the NI-TIO

Selected Source	the source input on the four general-purpose counter/timers on the NI-TIO
source	in the counter context, source refers to the signal that causes the counter to increment or decrement. In the context of signals, source refers to the device that drives a signal.
SRC	Source—the signal that causes the counter to increment or decrement
start trigger	a TTL level signal having two discrete levels—a high and a low level—that starts an operation
synchronous	a property of an event that is synchronized to a reference clock

T

TC	Terminal Count—a strobe that occurs when a counter reaches zero from either direction
termination	matching of impedances at the end of a signal path to minimize reflections
timebase	another term used for the SOURCE of a counter. Usually indicates an internal SOURCE provided by or derived from an onboard oscillator.
TIO	Timing I/O
trigger	any event that causes, starts, or stops some form of data capture
TTL	Transistor-Transistor Logic
two-pulse encoder	a motion encoder that has two channels: channels A and B. Pulses on channel A indicate movement in one direction while pulses on channel B indicate movement in the opposite direction. This type of encoder is also referred to as up down encoder.

U

unstrobed digital I/O	a type of digital input or output in which software reads or writes the digital line or port states directly, without using any handshaking or hardware-controlled timing functions. Also called *immediate*, *nonhandshaking*, or *unlatched* digital I/O.
UP_DOWN	the signal that determines whether a counter increments or decrements

V

VI	Virtual Instrument—a LabVIEW software module which consists of a front panel user interface and a block diagram program.

W

wire	data path between nodes