USER GUIDE AND SPECIFICATIONS

NI cDAQ-9178/9174

For NI cDAQ-9178 (8-Slot) and NI cDAQ-9174 (4-Slot) Chassis

Contents

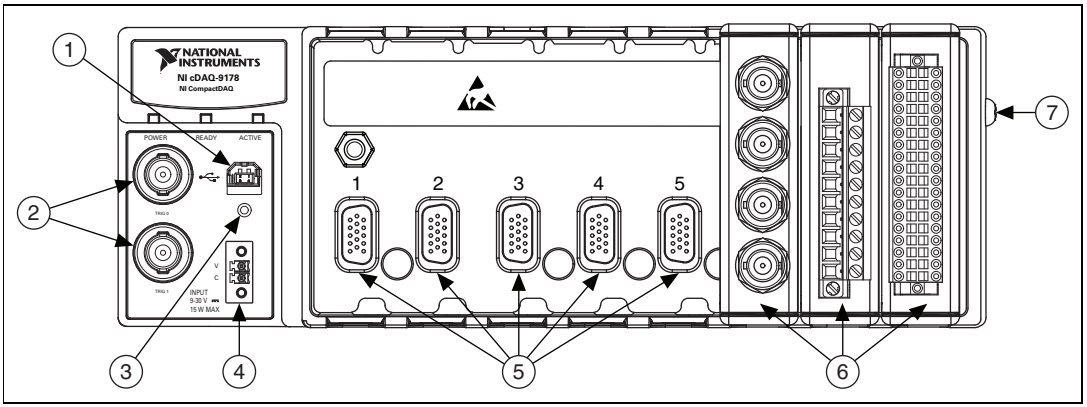
NATIONAL INSTRUMENTS

Introduction

This user guide describes how to use the National Instruments cDAQ-9178/9174 chassis and lists specifications. For an interactive demonstration of how to install the NI cDAQ-9178/9174, go to `ni.com/info` and enter `daqinstall`.

The NI cDAQ-9178 (eight slots) and NI cDAQ-9174 (four slots) USB chassis are designed for use with C Series I/O modules. The NI cDAQ-9178/9174 chassis is capable of measuring a broad range of analog and digital I/O signals and sensors using a Hi-Speed USB 2.0 interface. For module specifications, refer to the documentation included with your C Series I/O module(s) or go to `ni.com/manuals`.

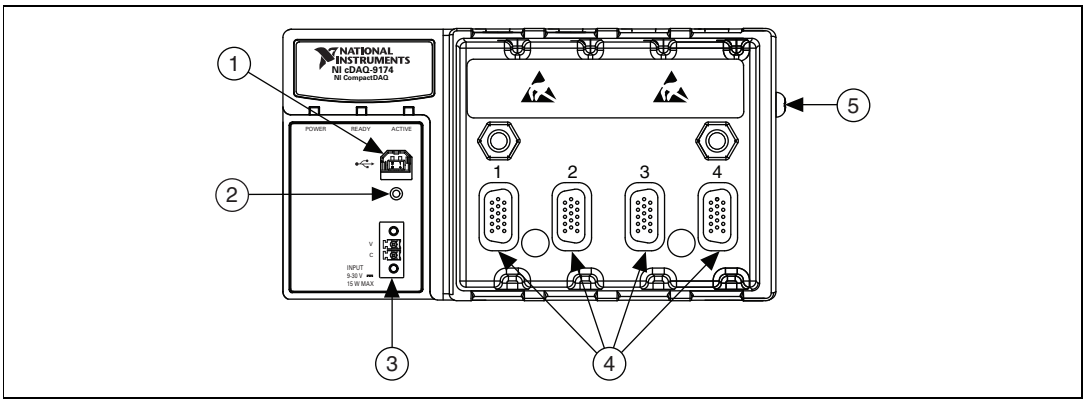
Figure 1 shows the NI cDAQ-9178 chassis.



1	USB Connector	4	9–30 VDC Power Connector
2	TRIG 0 (PFI 0) BNC Connector, TRIG 1 (PFI 1) BNC Connector	5	Module Slots
		6	Installed C Series Modules
3	USB Cable Strain Relief	7	Grounding Screw

Figure 1. NI cDAQ-9178 Chassis

Figure 2 shows the NI cDAQ-9174 chassis.



1	USB Connector	4	Module Slots
2	USB Cable Strain Relief	5	Grounding Screw
3	9–30 VDC Power Connector		

Figure 2. NI cDAQ-9174 Chassis

Safety Guidelines

Operate the NI cDAQ-9178/9174 chassis only as described in this user guide.

Note Because some C Series I/O modules may have more stringent certification standards than the NI cDAQ-9178/9174 chassis, the combined system may be limited by individual component restrictions. Refer to the *Using the NI cDAQ-9178/9174* section of this document for more details.

Caution The NI cDAQ-9178/9174 chassis is *not* certified for use in hazardous locations.

Hot Surface This icon denotes that the component may be hot. Touching this component may result in bodily injury.

Safety Guidelines for Hazardous Voltages

If *hazardous voltages* are connected to the module, take the following precautions. A hazardous voltage is a voltage greater than 42.4 V_{pk} or 60 VDC to earth ground.

Caution Ensure that hazardous voltage wiring is performed only by qualified personnel adhering to local electrical standards.

Caution Do *not* mix hazardous voltage circuits and human-accessible circuits on the same module.

Caution Make sure that chassis and circuits connected to the module are properly insulated from human contact.

Caution The NI cDAQ-9178/9174 chassis provides no isolation, but some modules offer isolation. Follow the safety guidelines for each module when using hazardous voltage.

Information Resources

In addition to the documentation included with your products, check `ni.com/manuals` for the most recent hardware and software documentation.

Technical Support on the Web

For additional support, refer to `ni.com/support` or `zone.ni.com`.

Training Courses

If you need more help developing an application with NI products, NI offers training courses. For more information, refer to `ni.com/training`.

Installing the Software

Install NI-DAQmx. For more information, download the *DAQ Getting Started Guide* from `ni.com/manuals`.

Installing Other Software

If you are using other software, refer to the installation instructions that accompany your software.

NI cDAQ-9178/9174 Dimensions

Figure 3 shows the dimensions of the NI cDAQ-9178/9174 chassis.

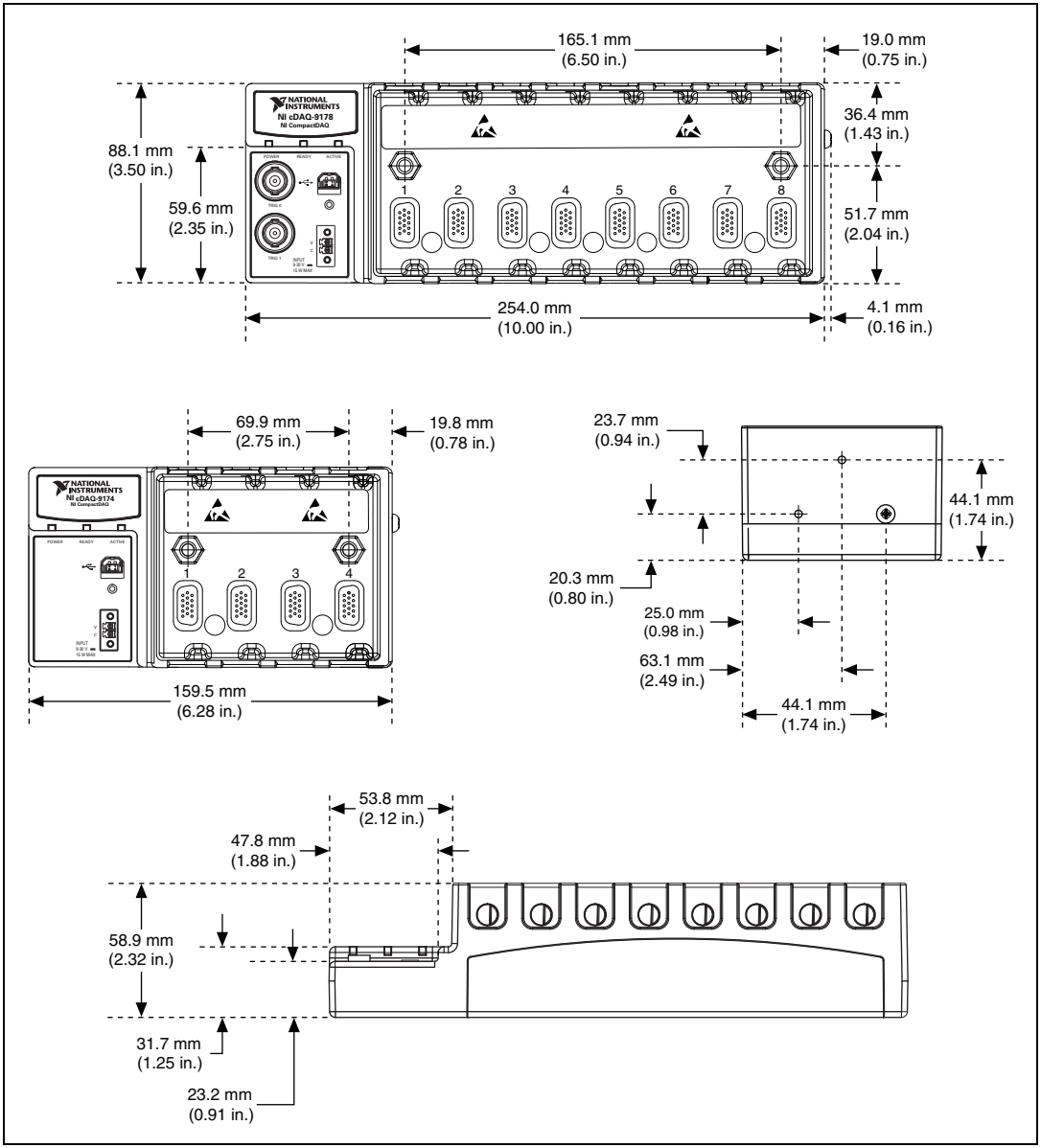


Figure 3. NI cDAQ-9178/9174 with Dimensions in Millimeters (Inches)

Mounting the NI cDAQ-9178/9174

You can mount the NI cDAQ-9178/9174 chassis using a desktop, 35 mm DIN-Rail, or panel mount accessory kit. For accessory ordering information, refer to `ni.com`.

Caution Your installation must meet the following requirements:

- Allows 25.4 mm (1 in.) of clearance above and below the NI cDAQ-9178/9174 chassis for air circulation.
- Allows at least 50.8 mm (2 in.) of clearance in front of the modules for common connector cabling such as the 10-terminal detachable screw terminal connector and, as needed, up to 88.9 mm (3.5 in.) of clearance in front of the modules for other types of cabling. For more information about mechanical dimensions, refer to `ni.com/info` and enter the info code `rdcrioconn`.

NI 9901 Desktop Mounting Kit

The NI 9901 desktop mounting kit includes two metal feet you can install on the sides of the NI cDAQ-9178/9174 chassis for desktop use. With this kit, you can tilt the NI cDAQ-9178/9174 chassis for convenient access to the I/O module connectors. When you install the two metal feet, the two existing screws on the back side and I/O end of the chassis must be removed, as shown in Figure 4. After removing the screws, replace them with the two longer screws included in the NI 9901 desktop mounting kit.

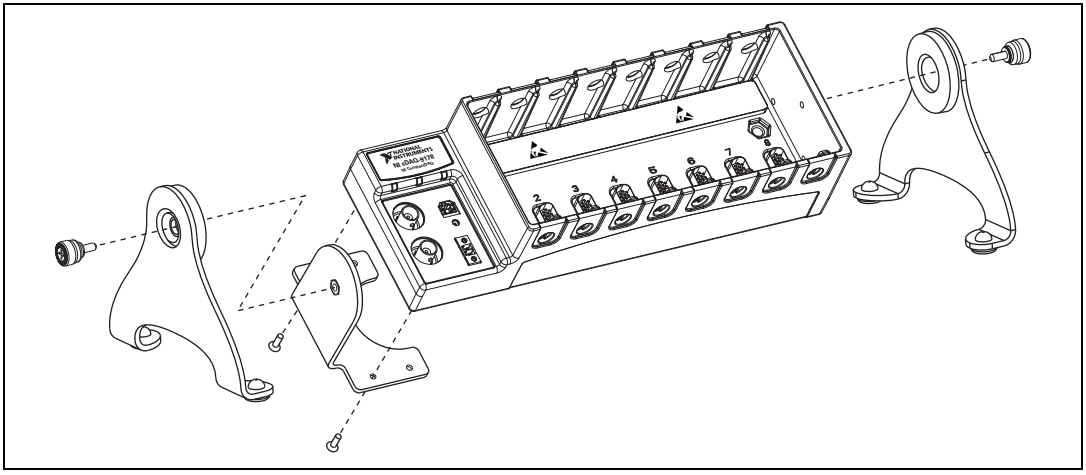


Figure 4. NI 9901 Desktop Mounting Kit

DIN-Rail Mounting Kits

- **NI 9915 DIN-Rail Kit**—For the NI cDAQ-9178.
- **NI 9912 DIN-Rail Kit**—For the NI cDAQ-9174.

Each DIN-Rail kit contains one clip for mounting the chassis on a standard 35 mm DIN-Rail. To mount the chassis on a DIN-Rail, fasten the DIN-Rail clip to the chassis using a number 2 Phillips screwdriver and two M4 × 17 screws. The screws are included in the DIN-Rail kit. Make sure the DIN-Rail kit is installed as illustrated in Figure 5, with the larger lip of the DIN-Rail positioned up. When the DIN-Rail kit is properly installed, the NI cDAQ-9178/9174 chassis is centered on the DIN-Rail.

Caution Remove the I/O modules before removing the chassis from the DIN-Rail.

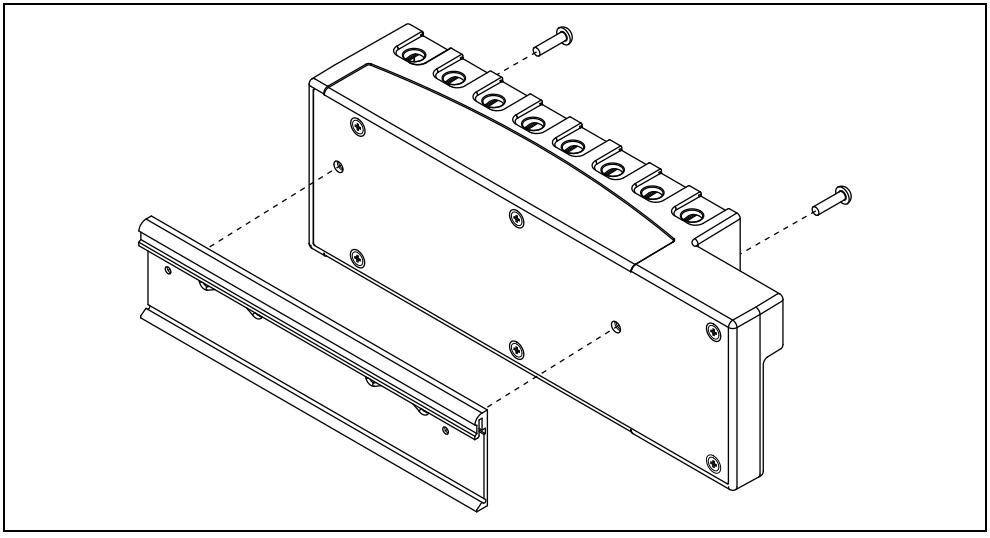
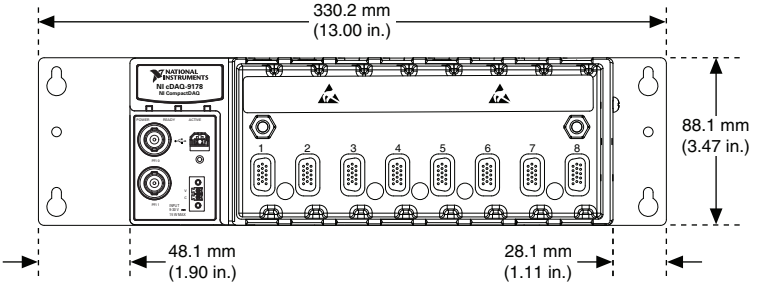


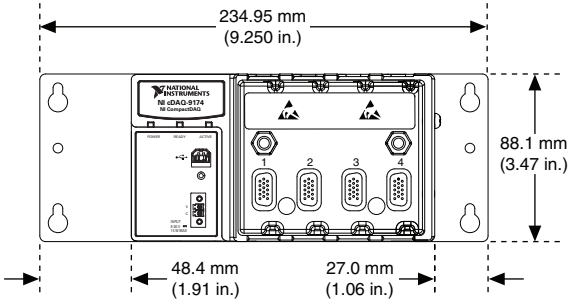
Figure 5. DIN-Rail Installation on the NI cDAQ-9178

Panel Mount Kits

- **NI 9905 Panel Mount Kit**—For the NI cDAQ-9178
- **NI 9904 Panel Mount Kit**—For the NI cDAQ-9174

To mount the chassis on a panel, align the chassis on the panel mount accessory. Attach the chassis to the panel mount kit using two M4 × 17 screws as shown in Figure 6. National Instruments provides these screws with the panel mount kit. You *must* use these screws because they are the correct depth and thread for the panel. These slots in the panel mount kit can be used with M4, M5, No. 8, or No.10 panhead screws. Figure 6 illustrates the panel dimensions and installation on the NI cDAQ-9178/9174 chassis. Refer to the documentation included with the panel mount kit for more detailed dimensions.





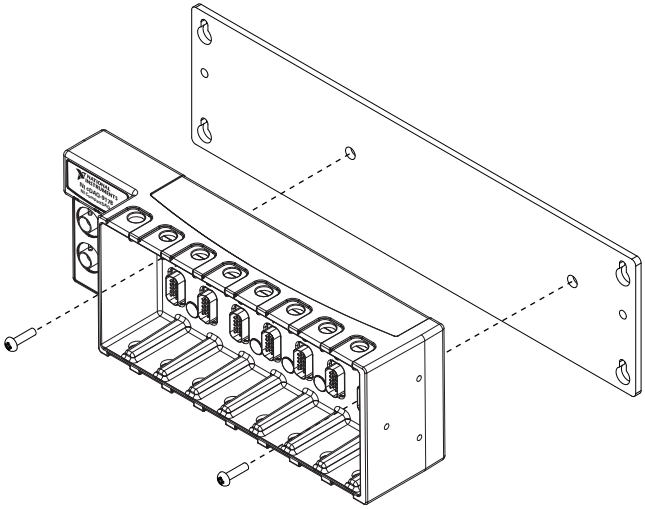


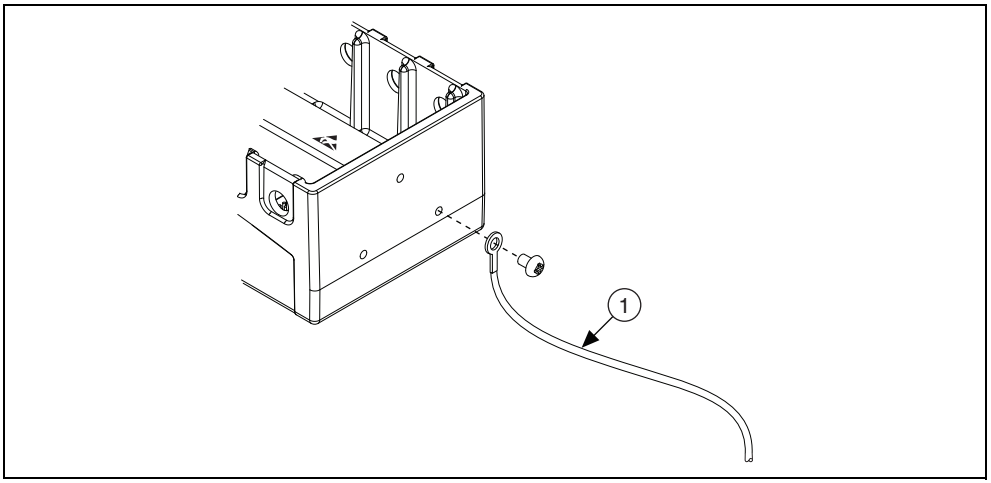
Figure 6. Panel Mount Dimensions and Installation on the NI cDAQ-9178/9174

Complete the following steps to prepare the NI cDAQ-9178/9174 chassis for use:

1. Before connecting the hardware, install the NI-DAQmx software.

Note The NI-DAQmx software is included on the CD shipped with your kit and is available for download at `ni.com/support`. The documentation for NI-DAQmx is available after installation from **Start»All Programs»National Instruments»NI-DAQ**. Other NI documentation is available from `ni.com/manuals`.

2. Make sure the NI cDAQ-9178/9174 power source is not connected.
3. Attach a ring lug to a 14 AWG (1.6 mm) wire. Connect the ring lug to the ground terminal on the side of the chassis using the ground screw. Attach the other end of the wire to the system safety ground as shown in Figure 7.



1	Attached to System Ground

Figure 7. Ring Lug Attached to Ground Terminal

Note Additionally, attach a wire with a ring lug to all other C Series I/O module cable shields. You *must* connect this wire to the ground terminal of the chassis using the ground screw.

4. Squeeze both C Series I/O module latches, insert the I/O module into the module slot, and press until both latches lock the module in place.

Note In order to meet EMC compliance standards, you must apply the ferrite included with your chassis to the power and USB cables. For more information about attaching the ferrite to the cables, refer to the *Ferrite Installation for NI cDAQ-9178/9174 Cables* document, included with your chassis.

5. Connect the NI cDAQ-9178/9174 chassis with the supplied USB cable to any available USB port on your computer. Use the jackscrew on the locking USB cable to securely attach the cable to the chassis.
6. Connect the supplied power source to the NI cDAQ-9178/9174 chassis. The NI cDAQ-9178/9174 chassis requires an external power supply that meets the specifications in the *Power Requirements* section.

7. Double-click the **Measurement & Automation** icon, shown at left, on the desktop to open MAX.
8. Expand **Devices and Interfaces**.
9. Check that your device appears under **Devices and Interfaces**. If your device does not appear, press <F5> to refresh the view in MAX. If your device is still not recognized, refer to `ni.com/support/install` for troubleshooting information.
10. Right-click your device and select **Self-Test**. If you need help during the self-test, select **Help» Help Topics»NI-DAQmx** and click **MAX Help for NI-DAQmx**.

    When the self-test finishes, a message indicates successful verification or an error. If an error occurs, refer to `ni.com/support/install` for troubleshooting information.

Note When in use, the NI cDAQ-9178/9174 chassis may become warm to the touch. This is normal.

Understanding LED Indications

Power LED

The Power LED indicates whether the NI cDAQ-9178/9174 is receiving power.

Table 1. Power LED

LED	Definition
Green	Power supplied
Off	No power supplied

Ready LED

The Ready LED is lit when the NI cDAQ-9178/9174 chassis is ready for use. The color indicates whether the USB connection is Full-Speed or Hi-Speed.

Table 2. Ready LED

LED	Definition
Amber	Hi-Speed (480 Mbit/sec)
Green	Full-Speed (12 Mbit/sec)
Off	USB connection is not established

Active LED

The Active LED indicates whether the NI cDAQ-9178/9174 chassis is communicating over the USB bus.

Table 3. Active LED

LED	Definition
Amber	Power is applied, but USB connection is not established
Green	USB traffic present
Off	No USB traffic present

The cDAQ system consists of three parts: C Series I/O modules, the cDAQ module interface, and the USB-STC3, as shown in Figure 8. These components digitize signals, perform D/A conversions to generate analog output signals, measure and control digital I/O signals, and provide signal conditioning.

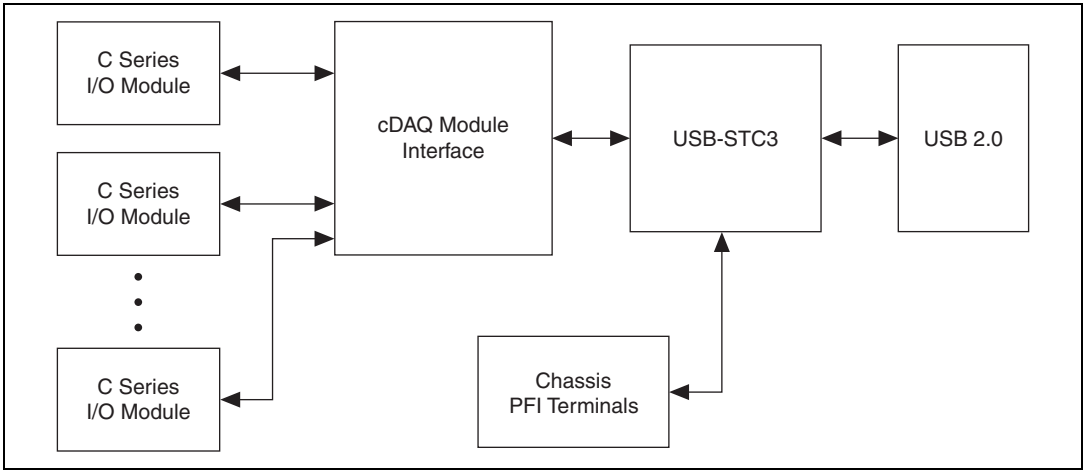


Figure 8. NI cDAQ-9178/9174 Block Diagram

C Series I/O Modules

National Instruments C Series I/O modules provide built-in signal conditioning and screw terminal, spring terminal, BNC, D-SUB, or RJ-50 connectors. A wide variety of I/O types are available, allowing you to customize the cDAQ system to meet your application needs.

C Series I/O modules are hot-swappable and automatically detected by the NI cDAQ-9178/9174 chassis. I/O channels are accessible using the NI-DAQmx driver software.

Because the modules contain built-in signal conditioning for extended voltage ranges or industrial signal types, you can usually make your wiring connections directly from the C Series I/O modules to your sensors/actuators. In most cases, the C Series I/O modules provide isolation from channel-to-earth ground.

For more information about which C Series I/O modules are compatible with the NI cDAQ-9178/9174 chassis, refer to the KnowledgeBase document, *C Series Modules Supported in the NI cDAQ-9178/9174 CompactDAQ*. To access this KnowledgeBase, go to `ni.com/info` and enter the info code `rdcdaq`.

Hardware-Timed Versus Static DIO Modules

Digital I/O module capabilities are determined by the type of digital signals that the module is capable of measuring or generating. Static digital I/O modules are designed for signals that change slowly and are accessed by software-timed reads and writes. Hardware-timed digital I/O modules are for signals that change rapidly and are updated by either software-timed or hardware-timed reads and writes. For more information about digital I/O modules, refer to the *Digital I/O* section.

cDAQ Module Interface

The cDAQ Module Interface manages data transfers between the USB-STC3 and the C Series I/O modules. The interface also handles autodetection, signal routing, and synchronization.

USB-STC3

The USB-STC3 features independent high-speed data streams; flexible AI, AO, and DIO sample timing; triggering; PFI signals for multi-device synchronization; flexible counter/timers with hardware gating; digital waveform acquisition and generation; and static DIO.

AI, AO, and DIO Sample Timing

The USB-STC3 contains advanced AI, AO, and DIO timing engines. A wide range of timing and synchronization signals are available through the PFI lines. Refer to the *Analog Input Timing Signals*, *Analog Output Timing Signals*, *Digital Input Timing Signals*, and *Digital Output Timing Signals* sections for more information about the configuration of these signals.

Triggering Modes

The NI cDAQ-9178/9174 supports different trigger modes, such as start trigger, reference trigger, and pause trigger with analog, digital, or software sources. Refer to the *Analog Input Triggering*, *Analog Output Triggering*, and *Digital Output Triggering* sections for more information.

Independent Data Streams

The NI cDAQ-9178/9174 supports seven independent high-speed data streams, allowing for up to seven simultaneous hardware timed tasks, such as analog input, analog output, buffered counter/timers, and hardware-timed digital input/output.

PFI Signals

The PFI signals provide access to advanced features such as triggering, synchronization, and counter/timers. PFI signals are available through hardware-timed digital input and output modules installed in up to two chassis slots and through the two PFI terminals provided on the NI cDAQ-9178 chassis. Refer to the *PFI* section for more information.

Flexible Counter/Timers

The NI cDAQ-9178/9174 includes four general-purpose 32-bit counter/timers that can be used to count edges, measure pulse-widths, measure periods and frequencies, and perform position measurements (encoding). In addition, the counter/timers can generate pulses, pulse trains, and square waves with adjustable frequencies. You can access the counter inputs and outputs using hardware-timed digital I/O modules installed in up to two slots, or by using the two chassis PFI terminals (NI cDAQ-9178 only). Refer to the *Counters* section for more information.

Analog Input

To perform analog input measurements, insert a supported analog input C Series I/O module into any slot on the cDAQ chassis. The measurement specifications, such as number of channels, channel configuration, sample rate, and gain, are determined by the type of C Series I/O module used. For more information and wiring diagrams, refer to the documentation included with your C Series I/O modules.

The NI cDAQ-9178/9174 has three AI timing engines, which means that three analog input tasks can be running at a time on a chassis. An analog input task can include channels from multiple analog input modules. However, channels from a single module cannot be used in multiple tasks.

Multiple timing engines allow the NI cDAQ-9178/9174 to run up to three analog input tasks simultaneously, each using independent timing and triggering configurations. The three AI timing engines are ai, te0, and te1.

Analog Input Triggering

A trigger is a signal that causes an action, such as starting or stopping the acquisition of data. When you configure a trigger, you must decide how you want to produce the trigger and the action you want the

trigger to cause. The NI cDAQ-9178/9174 chassis supports internal software, external digital triggering, and analog triggering.

Three triggers are available: Start Trigger, Reference Trigger, and Pause Trigger. An analog or digital trigger can initiate these three trigger actions. Up to two C Series hardware-timed digital input modules can be used in any chassis slot(s) to supply a digital trigger. To find your module triggering options, refer to the documentation included with your C Series I/O modules. For more information about using digital modules for triggering, refer to the *Digital I/O* section.

AI Start Trigger Signal

Use the Start Trigger signal to begin a measurement acquisition. A measurement acquisition consists of one or more samples. If you do not use triggers, begin a measurement with a software command. Once the acquisition begins, configure the acquisition to stop in one of the following ways:

- When a certain number of points has been sampled (in finite mode)
- After a hardware reference trigger (in finite mode)
- With a software command (in continuous mode)

An acquisition that uses a start trigger (but not a reference trigger) is sometimes referred to as a posttriggered acquisition. That is, samples are measured only after the trigger.

When you are using an internal sample clock, you can specify a default delay from the start trigger to the first sample.

Using a Digital Source

To use the Start Trigger signal with a digital source, specify a source and an edge. Use the following signals as the source:

- Any PFI terminal
- Counter *n* Internal Output

The source also can be one of several other internal signals on your NI cDAQ-9178/9174 chassis. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments»NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help*, go to `ni.com/manuals`.

You also can specify whether the measurement acquisition begins on the rising edge or falling edge of StartTrigger.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event. When you use an analog trigger source for StartTrigger, the acquisition begins on the first rising edge of the Analog Comparison Event signal.

Routing AI Start Trigger to an Output Terminal

You can route the Start Trigger signal to any output PFI terminal. The output is an active high pulse.

AI Reference Trigger Signal

Use Reference Trigger to stop a measurement acquisition. To use a reference trigger, specify a buffer of finite size and a number of pretrigger samples (samples that occur before the reference trigger). The number of posttrigger samples (samples that occur after the reference trigger) desired is the buffer size minus the number of pretrigger samples.

Once the acquisition begins, the NI cDAQ-9178/9174 chassis writes samples to the buffer. After the NI cDAQ-9178/9174 chassis captures the specified number of pretrigger samples, the NI cDAQ-9178/9174 chassis begins to look for the reference trigger condition. If the reference trigger condition occurs before the NI cDAQ-9178/9174 captures the specified number of pretrigger samples, the NI cDAQ-9178/9174 ignores the condition.

If the buffer becomes full, the NI cDAQ-9178/9174 continuously discards the oldest samples in the buffer to make space for the next sample. This data can be accessed (with some limitations) before the NI cDAQ-9178/9174 chassis discards it. Refer to the KnowledgeBase document, *Can a Pretriggered Acquisition be Continuous?*, for more information. To access this KnowledgeBase, go to `ni.com/info` and enter the info code `rdcanq`.

When the reference trigger occurs, the NI cDAQ-9178/9174 continues to write samples to the buffer until the buffer contains the number of posttrigger samples desired. Figure 9 shows the final buffer.

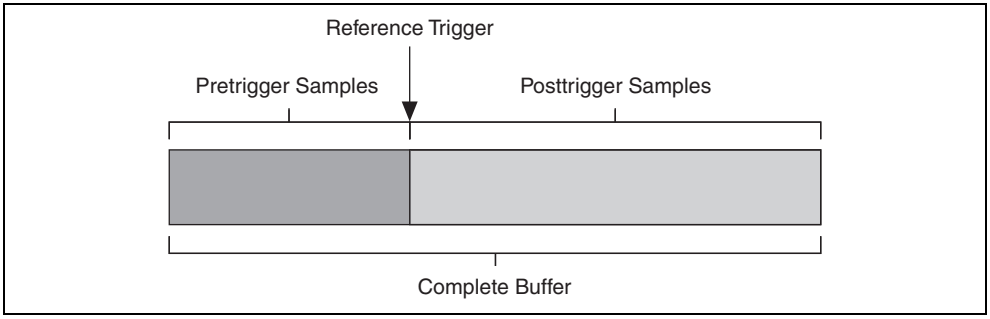


Figure 9. Reference Trigger Final Buffer

Using a Digital Source

To use Reference Trigger with a digital source, specify a source and an edge. Either PFI or one of several internal signals on the NI cDAQ-9178/9174 chassis can provide the source. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

You also can specify whether the measurement acquisition stops on the rising edge or falling edge of the Reference Trigger.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event.

When you use an analog trigger source, the acquisition stops on the first rising or falling edge of the Analog Comparison Event signal, depending on the trigger properties.

Routing the Reference Trigger Signal to an Output Terminal

You can route ReferenceTrigger to any output PFI terminal. Reference Trigger is active high by default.

AI Pause Trigger Signal

You can use the Pause Trigger to pause and resume a measurement acquisition. The internal sample clock pauses while the external trigger signal is active and resumes when the signal is inactive. You can program the active level of the pause trigger to be high or low.

Using a Digital Source

To use the Pause Trigger, specify a source and a polarity. The source can be either from PFI or one of several other internal signals on your NI cDAQ-9178/9174 chassis. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help in LabVIEW**. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event.

When you use an analog trigger source, the internal sample clock pauses when the Analog Comparison Event signal is low and resumes when the signal goes high (or vice versa).

Note Pause triggers are only sensitive to the level of the source, not the edge.

Analog Input Timing Signals

Sample Clock

A sample consists of one reading from each channel in the AI task. SampleClock signals the start of a sample of all analog input channels in the task. SampleClock can be generated from external or internal sources as shown in Figure 10.

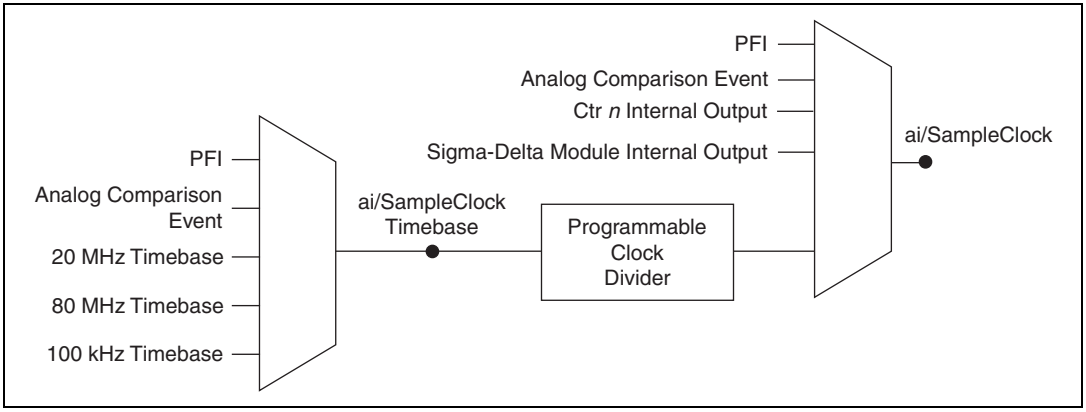


Figure 10. Sample Clock Timing Options

Routing the Sample Clock to an Output Terminal

You can route SampleClock to any output PFI terminal. SampleClock is an active high pulse by default.

AI Sample Clock Timebase

The AI Sample Clock Timebase signal is divided down to provide a source for SampleClock. SampleClockTimebase can be generated from external or internal sources. SampleClockTimebase is not available as an output from the chassis.

AI Convert Clock Behavior For Analog Input Modules

Scanned

Scanned C Series analog input modules contain a single A/D converter and a multiplexer to select between multiple input channels. When the cDAQ Module Interface receives a Sample Clock pulse, it begins generating a Convert Clock for each scanned module in the current task. Each Convert Clock

signals the acquisition of a single channel from that module. The Convert Clock rate depends on the module being used, the number of channels used on that module, and the system Sample Clock rate.

The driver chooses the fastest conversion rate possible based on the speed of the A/D converter for each module and adds 10 µs of padding between each channel to allow for adequate settling time. This scheme enables the channels to approximate simultaneous sampling. If the AI Sample Clock rate is too fast to allow for 10 µs of padding, NI-DAQmx selects a conversion rate that spaces the AI Convert Clock pulses evenly throughout the sample. NI-DAQmx uses the same amount of padding for all the modules in the task. To explicitly specify the conversion rate, use the **ActiveDevs** and **AI Convert Clock Rate** properties using the **DAQmx Timing** property node or functions.

Simultaneous Sample-and-Hold

Simultaneous sample-and-hold (SSH) C Series analog input modules contain multiple A/D converters or circuitry that allows all the input channels to be sampled at the same time. These modules sample their inputs on every Sample Clock pulse.

Sigma-Delta

Sigma-delta C Series analog input modules function much like SSH modules, but use A/D converters that require a high-frequency oversample clock to produce accurate, synchronized data. Sigma-delta modules in the cDAQ chassis automatically share a single oversample clock to synchronize data from all sigma-delta modules when they all share the same task. The NI cDAQ-9178/9174 supports a maximum of two synchronization pulse signals configured for your system. This limits the system to two tasks with sigma-delta modules.

The oversample clock is used as the AI Sample Clock Timebase. While most modules supply a common oversample clock frequency (12.8 MHz), some modules, such as the NI 9234, supply a different frequency. When sigma-delta modules with different oversample clock frequencies are used in an analog input task, the AI Sample Clock Timebase can use any of the available frequencies; by default, the fastest available is used. The sampling rate of all modules in the system is an integer divisor of the frequency of the AI Sample Clock Timebase.

When one or more sigma-delta modules are in an analog input task, the sigma-delta modules also provide the signal used as the AI Sample Clock. This signal is used to cause A/D conversion for other modules in the system, just as the AI Sample Clock does when a sigma-delta module is not being used.

When sigma-delta modules are in an AI task, the chassis automatically issues a synchronization pulse to each sigma-delta modules that resets their ADCs at the same time. Because of the filtering used in sigma-delta A/D converters, these modules usually exhibit a fixed input delay relative to non-sigma-delta modules in the system. This input delay is specified in the C Series I/O module documentation.

Slow Sample Rate Modules

Some C Series analog input modules are specifically designed for measuring signals that vary slowly, such as temperature. Because of their slow rate, it is not appropriate for these modules to constrain the AI Sample Clock to operate at or slower than their maximum rate. When using such a module in the cDAQ chassis, the maximum Sample Clock rate can run faster than the maximum rate for the module. When operating at a rate faster than these slow rate modules can support, the slow rate module returns the same point repeatedly, until a new conversion completes. In a hardware-timed task, the first point is acquired when the task is committed. The second point is acquired after the start trigger as shown in Figure 11.

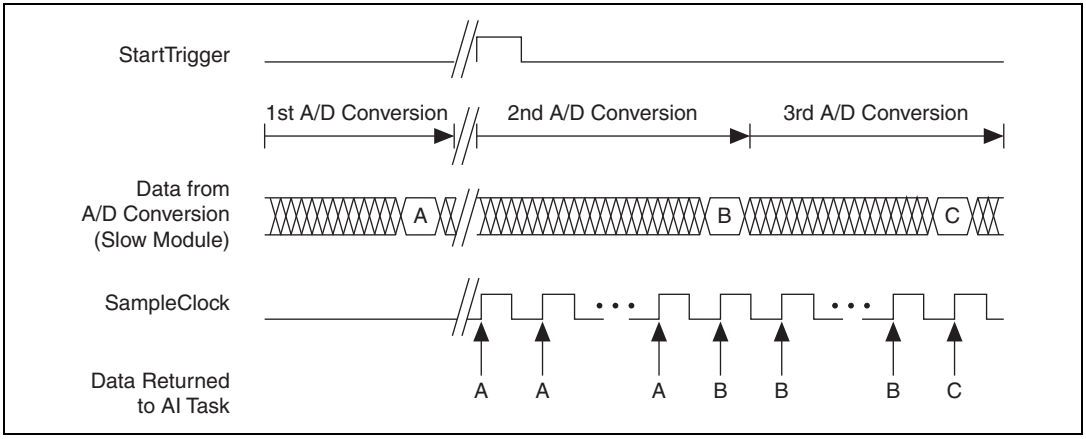


Figure 11. Sample Clock Timing Example

For example, if running an AI task at 1 kHz using a module with a maximum rate of 10 Hz, the slow module returns 100 samples of the first point, followed by 100 samples of the second point, etc. Other modules in the task will return 1,000 new data points per second, which is normal. When performing a single-point acquisition, no points are repeated. To avoid this behavior, use multiple AI timing engines, and assign slow sample rate modules to a task with a rate at or slower than their maximum rate.

Refer to the KnowledgeBase document, *C Series Modules Supported in the NI cDAQ-9178/9174 CompactDAQ*, for more information. To access this KnowledgeBase, go to `ni.com/info` and enter the info code `rdcdaq`.

Getting Started with AI Applications in Software

You can use the NI cDAQ-9178/9174 chassis in the following analog input applications:

- Single-Point
- Finite
- Continuous

For more information about programming analog input applications and triggers in software, refer to the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Analog Output

To generate analog output, insert an analog output C Series I/O module in any slot on the NI cDAQ-9178/9174 chassis. The generation specifications, such as the number of channels, channel configuration, update rate, and output range, are determined by the type of C Series I/O module used. For more information, refer to the documentation included with your C Series I/O modules.

You can run one hardware-timed (waveform) analog output task at a time on the NI cDAQ-9178/9174 chassis. At the same time, you can also run one or more software-timed (single-point or immediate) tasks.

For each analog output module, you can either:

- Assign all of the channels on the module to the hardware-timed task.
- Assign all of the channels on the module to one or more software-timed tasks.

On a single AO module, you cannot assign some channels to a hardware-timed task and other channels (on the same module) to a software-timed task.

Analog Output Data Generation Methods

When performing an analog output operation, you either can perform software-timed or hardware-timed generations. Hardware-timed generations must be buffered.

Software-Timed Generations

With a software-timed generation, software controls the rate at which data is generated. Software sends a separate command to the hardware to initiate each DAC conversion. In NI-DAQmx, software-timed generations are referred to as on-demand timing. Software-timed generations are also referred to as immediate or static operations. They are typically used for writing out a single value, such as a constant DC voltage.

The following considerations apply to software-timed generations:

- If any AO channel on a module is used in a hardware-timed (waveform) task, no channels on that module can be used in a software-timed task.
- You can configure software-timed generation to simultaneously update.
- Only one simultaneous update task can run at a time.
- Simultaneous update is not restricted to 16 channels.
- A hardware-timed AO task and a simultaneous update AO task cannot run at the same time.

Hardware-Timed Generations

With a hardware-timed generation, a digital hardware signal controls the rate of the generation. This signal can be generated internally on the chassis or provided externally.

Hardware-timed generations have several advantages over software-timed acquisitions:

- The time between samples can be much shorter.
- The timing between samples is deterministic.
- Hardware-timed acquisitions can use hardware triggering.

Hardware-timed AO operations on the NI cDAQ-9178/9174 chassis must be buffered.

Buffered Analog Output

A buffer is a temporary storage in computer memory for generated samples. In a buffered generation, data is moved from a host buffer to the NI cDAQ-9178/9174 onboard FIFO before it is written to the C Series I/O modules.

One property of buffered I/O operations is sample mode. The sample mode can be either finite or continuous:

- Finite—Finite sample mode generation refers to the generation of a specific, predetermined number of data samples. After the specified number of samples is written out, the generation stops.

- Continuous—Continuous generation refers to the generation of an unspecified number of samples. Instead of generating a set number of data samples and stopping, a continuous generation continues until you stop the operation. There are three different continuous generation modes that control how the data is written. These modes are regeneration, onboard regeneration, and non-regeneration:
  - In regeneration mode, you define a buffer in host memory. The data from the buffer is continually downloaded to the FIFO to be written out. New data can be written to the host buffer at any time without disrupting the output. There is no limitation on the number of waveform channels supported by regeneration mode.
  - With onboard regeneration, the entire buffer is downloaded to the FIFO and regenerated from there. After the data is downloaded, new data cannot be written to the FIFO. To use onboard regeneration, the entire buffer must fit within the FIFO size. The advantage of using onboard regeneration is that it does not require communication with the main host memory once the operation is started, which prevents problems that may occur due to excessive bus traffic or operating system latency. There is a limit of 16 waveform channels for onboard regeneration.
  - With non-regeneration, old data is not repeated. New data must continually be written to the buffer. If the program does not write new data to the buffer at a fast enough rate to keep up with the generation, the buffer underflows and causes an error. There is no limitation on the number of waveform channels supported by non-regeneration.

Analog Output Triggering

Analog output supports two different triggering actions:

- Start Trigger
- Pause Trigger
- AO Start Trigger
- AO Pause Trigger

An analog or digital trigger can initiate these actions. Up to two C Series hardware-timed digital input modules can be used in any chassis slot to supply a digital trigger. An analog trigger can be supplied by some C Series analog modules.

AO Start Trigger Signal

Use the AO Start Trigger (ao/StartTrigger) signal to initiate a waveform generation. If you do not use triggers, you can begin a generation with a software command. If you are using an internal sample clock, you can specify a delay from the start trigger to the first sample. For more information, refer to the *NI-DAQmx Help*. The *NI-DAQmx Help* is available after installation from **Start»All Programs» National Instruments»NI-DAQ»NI-DAQmx Help**.

Using a Digital Source

To use ao/StartTrigger, specify a source and a rising or falling edge. The source can be one of the following signals:

- A pulse initiated by host software
- Any PFI terminal
- AI Reference Trigger
- AI Start Trigger

The source also can be one of several internal signals on the NI cDAQ-9178/9174 chassis. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help*, go to `ni.com/manuals`.

You also can specify whether the waveform generation begins on the rising edge or falling edge of ao/StartTrigger.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event, depending on the trigger properties.

When you use an analog trigger source, the waveform generation begins on the first rising or falling edge of the Analog Comparison Event signal, depending on the trigger properties. The analog trigger circuit must be configured by a simultaneously running analog input task.

Routing AO Start Trigger Signal to an Output Terminal

You can route ao/StartTrigger to any output PFI terminal. The output is an active high pulse.

AO Pause Trigger Signal

Use the AO Pause Trigger signal (ao/PauseTrigger) to mask off samples in a DAQ sequence. When ao/PauseTrigger is active, no samples occur, but ao/PauseTrigger does not stop a sample that is in progress. The pause does not take effect until the beginning of the next sample.

When you generate analog output signals, the generation pauses as soon as the pause trigger is asserted. If the source of the sample clock is the onboard clock, the generation resumes as soon as the pause trigger is deasserted, as shown in Figure 12.

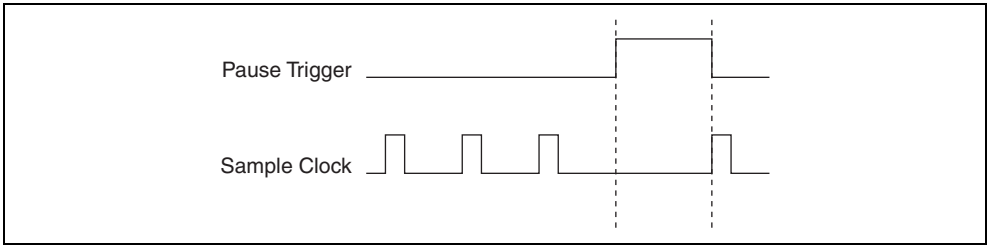


Figure 12. ao/PauseTrigger with the Onboard Clock Source

If you are using any signal other than the onboard clock as the source of the sample clock, the generation resumes as soon as the pause trigger is deasserted and another edge of the sample clock is received, as shown in Figure 13.

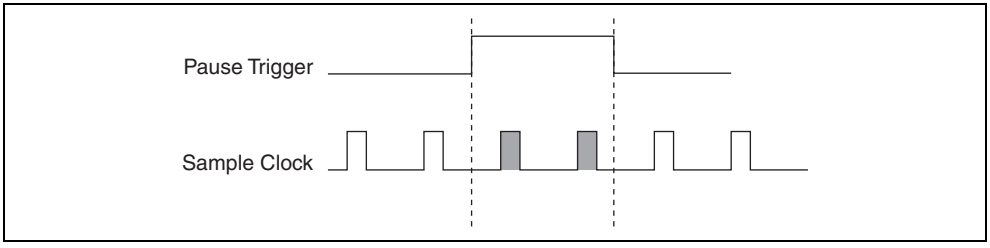


Figure 13. ao/PauseTrigger with Other Signal Source

Using a Digital Source

To use ao/PauseTrigger, specify a source and a polarity. The source can be a PFI signal or one of several other internal signals on the NI cDAQ-9178/9174 chassis.

You also can specify whether the samples are paused when ao/PauseTrigger is at a logic high or low level. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event, depending on the trigger properties.

When you use an analog trigger source, the samples are paused when the Analog Comparison Event signal is at a high or low level, depending on the trigger properties. The analog trigger circuit must be configured by a simultaneously running analog input task.

Analog Output Timing Signals

The NI cDAQ-9178/9174 chassis features the following AO (waveform generation) timing signals:

- AO Sample Clock
- AO Sample Clock Timebase

AO Sample Clock

The AO sample clock (ao/SampleClock) signals when all the analog output channels in the task update. ao/SampleClock can be generated from external or internal sources as shown in Figure 14.

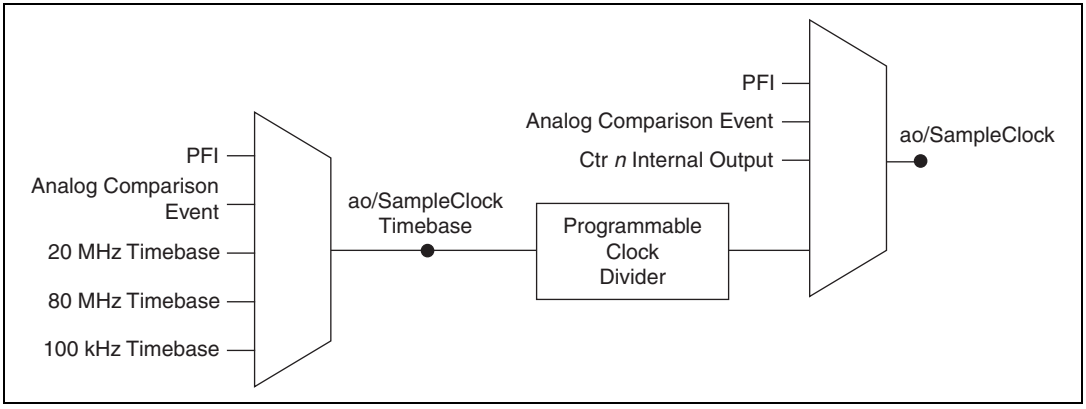


Figure 14. Analog Output Timing Options

Routing AO Sample Clock to an Output Terminal

You can route ao/SampleClock to any output PFI terminal. ao/SampleClock is active high by default.

AO Sample Clock Timebase Signal

The AO Sample Clock Timebase (ao/SampleClockTimebase) signal is divided down to provide a source for ao/SampleClock. ao/SampleClockTimebase can be generated from external or internal sources, and is not available as an output from the chassis.

Minimizing Glitches on the Output Signal

When you use a DAC to generate a waveform, you may observe glitches on the output signal. These glitches are normal; when a DAC switches from one voltage to another, it produces glitches due to released charges. The largest glitches occur when the most significant bit of the DAC code changes. You can build a lowpass deglitching filter to remove some of these glitches, depending on the frequency and nature of the output signal. Go to `ni.com/support` for more information about minimizing glitches.

Getting Started with AO Applications in Software

You can use the NI cDAQ-9178/9174 chassis in the following analog output applications:

- Single-Point (On-Demand) Generation
- Finite Generation
- Continuous Generation
- Waveform Generation

For more information about programming analog output applications and triggers in software, refer the *LabVIEW Help* or to the *NI-DAQmx Help*.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Digital I/O

To use digital I/O, insert a digital I/O C Series module into any slot on the NI cDAQ-9178/9174 chassis. The I/O specifications, such as number of lines, logic levels, update rate, and line direction, are determined by the type of C Series I/O module used. For more information, refer to the documentation included with your C Series I/O modules.

Hardware-Timed Versus Static DIO Modules

Digital I/O module capabilities are determined by the type of digital signals that the module is capable of measuring or generating. Static digital I/O modules are designed for signals that change slowly and are accessed by software-timed reads and writes. Hardware-timed digital I/O modules are for signals that change rapidly and are updated by either software-timed or hardware-timed reads and writes. Hardware-timed digital I/O modules can be used in any chassis slot and can perform the following tasks:

- Software-timed reads and writes
- Digital waveform generation and acquisition (hardware-timed input/output)
- Counter/timer (up to two modules)
- Access PFI signals (up to two modules)

To determine the capability of digital I/O modules supported by the NI cDAQ-9178/9174 chassis, refer to the KnowledgeBase document, *C Series Modules Supported in the NI cDAQ-9178/9174 CompactDAQ*. To access this KnowledgeBase document, go to `ni.com/info` and enter the info code `rdcdaq`.

Static DIO

Each of the DIO lines can be used as a static DI or DO line. You can use static DIO lines to monitor or control digital signals on some C Series I/O modules. Each DIO line can be individually configured as a digital input (DI) or digital output (DO), depending on the C Series I/O module being used.

All samples of static DI lines and updates of static DO lines are software-timed.

Digital Waveform Acquisition (Hardware-Timed Input)

You can acquire digital waveforms using hardware-timed digital modules. The DI waveform acquisition FIFO stores the digital samples. The NI cDAQ-9178/9174 chassis samples the DIO lines on each rising or falling edge of the di/SampleClock signal.

Digital Input Triggering

A trigger is a signal that causes an action, such as starting or stopping the acquisition of data. When you configure a trigger, you must decide how you want to produce the trigger and the action you want the trigger to cause. The NI cDAQ-9178/9174 chassis supports internal software, external digital triggering, and digital triggering.

Three triggers are available: Start Trigger, Reference Trigger, and Pause Trigger. An analog or digital trigger can initiate these three trigger actions. Up to two C Series hardware-timed digital input modules can be used in any chassis slot(s) to supply a digital trigger. To find your module triggering options, refer to the documentation included with your C Series I/O modules. For more information about using analog modules for triggering, refer to the *Analog Input* and *Analog Output* sections.

DI Start Trigger Signal

Use the DI Start Trigger (di/StartTrigger) signal to begin a measurement acquisition. A measurement acquisition consists of one or more samples. If you do not use triggers, begin a measurement with a software command. Once the acquisition begins, configure the acquisition to stop in one of the following ways:

- When a certain number of points has been sampled (in finite mode)
- After a hardware reference trigger (in finite mode)
- With a software command (in continuous mode)

An acquisition that uses a start trigger (but not a reference trigger) is sometimes referred to as a posttriggered acquisition. That is, samples are measured only after the trigger.

When you are using an internal sample clock, you can specify a delay from the start trigger to the first sample.

Using a Digital Source

To use di/StartTrigger with a digital source, specify a source and an edge. Use the following signals as the source:

- Any PFI terminal
- Counter *n* Internal Output

The source also can be one of several other internal signals on your NI cDAQ-9178/9174 chassis. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments»NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help*, go to `ni.com/manuals`.

You also can specify whether the measurement acquisition begins on the rising edge or falling edge of di/StartTrigger.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event. When you use an analog trigger source for di/StartTrigger, the acquisition begins on the first rising edge of the Analog Comparison Event signal.

Routing DI Start Trigger to an Output Terminal

You can route di/StartTrigger to any output PFI terminal. The output is an active high pulse.

DI Reference Trigger Signal

Use a reference trigger (di/ReferenceTrigger) signal to stop a measurement acquisition. To use a reference trigger, specify a buffer of finite size and a number of pretrigger samples (samples that occur before the reference trigger). The number of posttrigger samples (samples that occur after the reference trigger) desired is the buffer size minus the number of pretrigger samples.

Once the acquisition begins, the NI cDAQ-9178/9174 chassis writes samples to the buffer. After the NI cDAQ-9178/9174 chassis captures the specified number of pretrigger samples, the NI cDAQ-9178/9174 chassis begins to look for the reference trigger condition. If the reference trigger condition occurs before the NI cDAQ-9178/9174 captures the specified number of pretrigger samples, the NI cDAQ-9178/9174 ignores the condition.

If the buffer becomes full, the NI cDAQ-9178/9174 continuously discards the oldest samples in the buffer to make space for the next sample. This data can be accessed (with some limitations) before the NI cDAQ-9178/9174 chassis discards it. Refer to the KnowledgeBase document, *Can a Pretriggered Acquisition be Continuous?*, for more information. To access this KnowledgeBase, go to `ni.com/info` and enter the info code `rdcanq`.

When the reference trigger occurs, the NI cDAQ-9178/9174 continues to write samples to the buffer until the buffer contains the number of posttrigger samples desired. Figure 15 shows the final buffer.

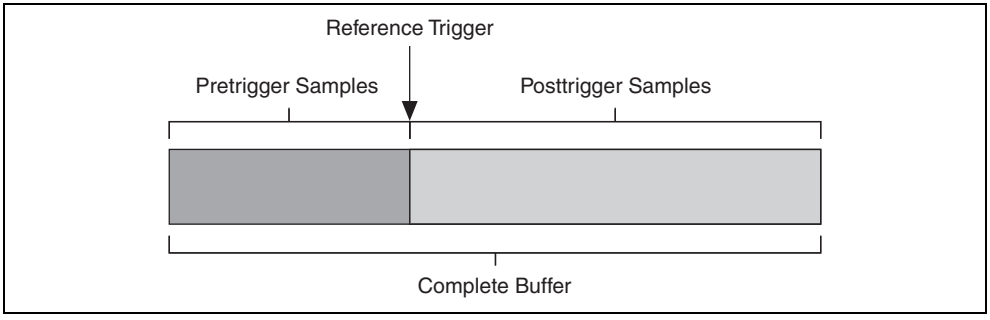


Figure 15. Reference Trigger Final Buffer

Using a Digital Source

To use di/ReferenceTrigger with a digital source, specify a source and an edge. Either PFI or one of several internal signals on the NI cDAQ-9178/9174 chassis can provide the source. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

You also can specify whether the measurement acquisition stops on the rising edge or falling edge of di/ReferenceTrigger.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event.

When you use an analog trigger source, the acquisition stops on the first rising or falling edge of the Analog Comparison Event signal, depending on the trigger properties.

Routing DI Reference Trigger Signal to an Output Terminal

You can route di/ReferenceTrigger to any output PFI terminal. Reference Trigger is active high by default.

DI Pause Trigger Signal

You can use the DI PauseTrigger (di/PauseTrigger) signal to pause and resume a measurement acquisition. The internal sample clock pauses while the external trigger signal is active and resumes when the signal is inactive. You can program the active level of the pause trigger to be high or low.

Using a Digital Source

To use di/PauseTrigger, specify a source and a polarity. The source can be either from PFI or one of several other internal signals on your NI cDAQ-9178/9174 chassis. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help in LabVIEW**. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event.

When you use an analog trigger source, the internal sample clock pauses when the Analog Comparison Event signal is low and resumes when the signal goes high (or vice versa).

Note Pause triggers are only sensitive to the level of the source, not the edge.

Digital Input Timing Signals

DI Sample Clock Signal

Use the DI Sample Clock (di/SampleClock) signal to sample digital I/O on any slot using hardware-timed digital modules, and store the result in the DI waveform acquisition FIFO. If the NI cDAQ-9178/9174 chassis receives a di/SampleClock signal when the FIFO is full, it reports an overflow error to the host software.

DI Sample Clock

A sample consists of one reading from each channel in the DI task. di/SampleClock signals the start of a sample of all digital input channels in the task. di/SampleClock can be generated from external or internal sources as shown in Figure 16.

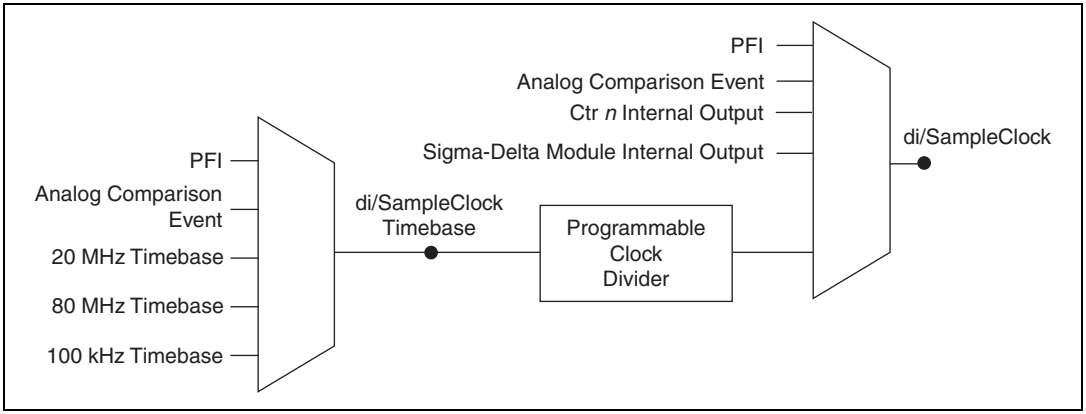


Figure 16. Sample Clock Timing Options

Routing DI Sample Clock to an Output Terminal

You can route di/SampleClock to any output PFI terminal.

DI Sample Clock Timebase

The DI Sample Clock Timebase (di/SampleClockTimebase) signal is divided down to provide a source for di/SampleClock. di/SampleClock Timebase can be generated from external or internal sources. di/SampleClockTimebase is not available as an output from the chassis.

Using an Internal Source

To use di/SampleClock with an internal source, specify the signal source and the polarity of the signal. Use the following signals as the source:

- AI Sample Clock
- AO Sample Clock
- Counter *n* Internal Output
- Frequency Output
- DI Change Detection Output

Several other internal signals can be routed to di/SampleClock. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Using an External Source

You can route the following signals as di/SampleClock:

- Any PFI terminal
- Analog Comparison Event (an analog trigger)

You can sample data on the rising or falling edge of di/SampleClock.

Routing DI Sample Clock to an Output Terminal

You can route di/SampleClock to any output PFI terminal. The PFI circuitry inverts the polarity of di/SampleClock before driving the PFI terminal.

Getting Started with DI Applications in Software

You can use the NI cDAQ-9178/9174 chassis in the following digital input applications:

- Single-Point
- Finite
- Continuous

For more information about programming digital input applications and triggers in software, refer to the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Change Detection Event

The Change Detection Event is the signal generated when a change on the rising or falling edge lines is detected by the change detection task.

Routing Change Detection Event to an Output Terminal

You can route ChangeDetectionEvent to any output PFI terminal.

Change Detection Acquisition

You can configure lines on hardware-timed digital modules to detect rising or falling edges. When one or more of these lines sees the edge specified for that line, the NI cDAQ-9178/9174 samples all the lines in the task. The rising and falling edge lines do not necessarily have to be in the task.

Change detection acquisitions can be buffered or nonbuffered:

- **Nonbuffered Change Detection Acquisition**—In a nonbuffered acquisition, data is transferred from the NI cDAQ-9178/9174 directly to a PC buffer.
- **Buffered Change Detection Acquisition**—A buffer is a temporary storage in computer memory for acquired samples. In a buffered acquisition, data is stored in the NI cDAQ-9178/9174 onboard FIFO then transferred to a PC buffer. Buffered acquisitions typically allow for much faster transfer rates than nonbuffered acquisitions because data accumulates and is transferred in blocks, rather than one sample at a time.

Digital Output

To generate digital output, insert a digital output C Series I/O module in any slot on the NI cDAQ-9178/9174 chassis. The generation specifications, such as the number of channels, channel configuration, update rate, and output range, are determined by the type of C Series I/O module used. For more information, refer to the documentation included with your C Series I/O modules.

You can run one hardware-timed digital output task at a time on the NI cDAQ-9178/9174 chassis. At the same time, you can also run one or more software-timed (single-point) tasks.

For each digital output module, you can either:

- Assign all of the channels on the module to the hardware-timed task.
- Assign all of the channels on the module to one or more software-timed tasks.

Digital Output Data Generation Methods

When performing a digital output operation, you either can perform software-timed or hardware-timed generations. Hardware-timed generations must be buffered.

Software-Timed Generations

With a software-timed generation, software controls the rate at which data is generated. Software sends a separate command to the hardware to initiate each digital generation. In NI-DAQmx, software-timed generations are referred to as on-demand timing. Software-timed generations are also referred to as immediate or static operations. They are typically used for writing out a single value.

For software-timed generations, if any DO channel on a module is used in a hardware-timed task, no channels on that module can be used in a software-timed task.

Hardware-Timed Generations

With a hardware-timed generation, a digital hardware signal controls the rate of the generation. This signal can be generated internally on the chassis or provided externally.

Hardware-timed generations have several advantages over software-timed acquisitions:

- The time between samples can be much shorter.
- The timing between samples is deterministic.
- Hardware-timed acquisitions can use hardware triggering.

Hardware-timed DO operations on the NI cDAQ-9178/9174 chassis must be buffered.

Buffered Digital Output

A buffer is a temporary storage in computer memory for generated samples. In a buffered generation, data is moved from a host buffer to the NI cDAQ-9178/9174 onboard FIFO before it is written to the C Series I/O modules.

One property of buffered I/O operations is sample mode. The sample mode can be either finite or continuous:

- Finite—Finite sample mode generation refers to the generation of a specific, predetermined number of data samples. After the specified number of samples is written out, the generation stops.
- Continuous—Continuous generation refers to the generation of an unspecified number of samples. Instead of generating a set number of data samples and stopping, a continuous generation continues until you stop the operation. There are three different continuous generation modes that control how the data is written. These modes are regeneration, onboard regeneration, and non-regeneration:
  - In regeneration mode, you define a buffer in host memory. The data from the buffer is continually downloaded to the FIFO to be written out. New data can be written to the host buffer at any time without disrupting the output.
  - With onboard regeneration, the entire buffer is downloaded to the FIFO and regenerated from there. After the data is downloaded, new data cannot be written to the FIFO. To use onboard regeneration, the entire buffer must fit within the FIFO size. The advantage of using onboard regeneration is that it does not require communication with the main host memory once the operation is started, which prevents problems that may occur due to excessive bus traffic or operating system latency.

Note For the NI cDAQ-9178, install modules in slots 1 through 4 to maximize accessible FIFO size. Using a module in slots 5 through 8 will reduce the accessible FIFO size.

  - With non-regeneration, old data is not repeated. New data must continually be written to the buffer. If the program does not write new data to the buffer at a fast enough rate to keep up with the generation, the buffer underflows and causes an error.

Digital Output Triggering

Digital output supports two different triggering actions:

- Start Trigger
- Pause Trigger

A digital or digital trigger can initiate these actions. Any PFI terminal can supply a digital trigger, and some C Series analog modules can supply an analog trigger. For more information, refer to the *Minimizing Glitches on the Output Signal* section of this document or to the documentation included with your C Series I/O module(s).

Digital Output Timing Signals

The NI cDAQ-9178/9174 chassis features the following DO timing signals:

- DO Sample Clock Timebase
- DO Start Trigger
- DO Pause Trigger

DO Sample Clock Signal

The DO Sample Clock (do/SampleClock) signals when all the digital output channels in the task update. do/SampleClock can be generated from external or internal sources as shown in Figure 17.

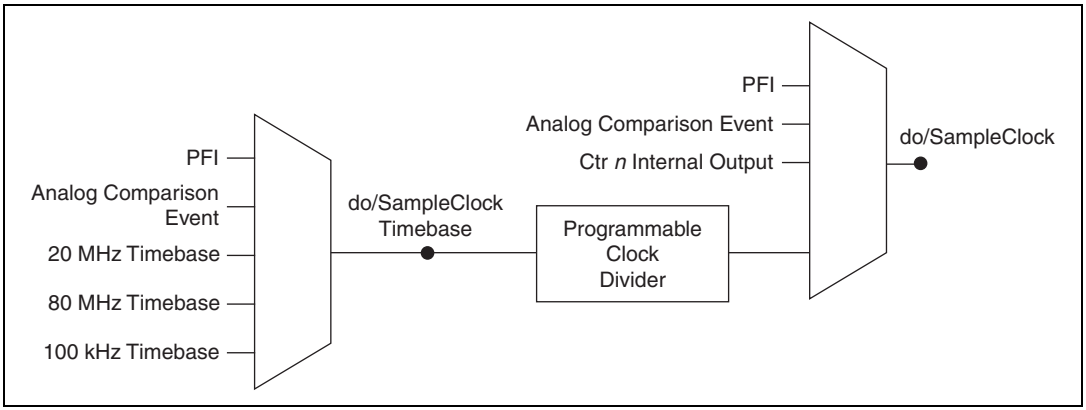


Figure 17. Digital Output Timing Options

Routing DO Sample Clock to an Output Terminal

You can route do/SampleClock to any output PFI terminal. do/SampleClock is active low by default.

DO Sample Clock Timebase Signal

The DO Sample Clock Timebase (do/SampleClockTimebase) signal is divided down to provide a source for do/SampleClock. do/SampleClockTimebase can be generated from external or internal sources, and is not available as an output from the chassis.

DO Start Trigger Signal

Use the DO Start Trigger (do/StartTrigger) signal to initiate a waveform generation. If you do not use triggers, you can begin a generation with a software command. If you are using an internal sample clock, you can specify a delay from the start trigger to the first sample. For more information, refer to the *NI-DAQmx Help*. The *NI-DAQmx Help* is available after installation from **Start»All Programs» National Instruments»NI-DAQ»NI-DAQmx Help**.

Using a Digital Source

To use do/StartTrigger, specify a source and a rising or falling edge. The source can be one of the following signals:

- A pulse initiated by host software
- Any PFI terminal
- AI Reference Trigger
- AI Start Trigger

The source also can be one of several internal signals on the NI cDAQ-9178/9174 chassis. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments»NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help*, go to `ni.com/manuals`.

You also can specify whether the waveform generation begins on the rising edge or falling edge of do/StartTrigger.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event, depending on the trigger properties.

When you use an analog trigger source, the waveform generation begins on the first rising or falling edge of the Analog Comparison Event signal, depending on the trigger properties. The analog trigger circuit must be configured by a simultaneously running analog input task.

Routing DO Start Trigger Signal to an Output Terminal

You can route do/StartTrigger to any output PFI terminal. The output is an active high pulse.

DO Pause Trigger Signal

Use the DO Pause Trigger signal (do/PauseTrigger) to mask off samples in a DAQ sequence. When do/PauseTrigger is active, no samples occur, but do/PauseTrigger does not stop a sample that is in progress. The pause does not take effect until the beginning of the next sample.

When you generate digital output signals, the generation pauses as soon as the pause trigger is asserted. If the source of the sample clock is the onboard clock, the generation resumes as soon as the pause trigger is deasserted, as shown in Figure 18.

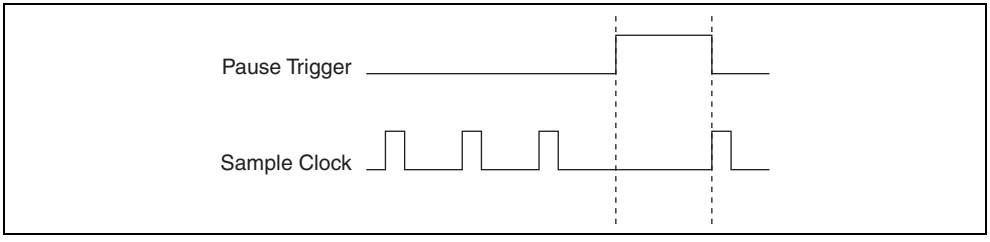


Figure 18. do/PauseTrigger with the Onboard Clock Source

If you are using any signal other than the onboard clock as the source of the sample clock, the generation resumes as soon as the pause trigger is deasserted and another edge of the sample clock is received, as shown in Figure 19.

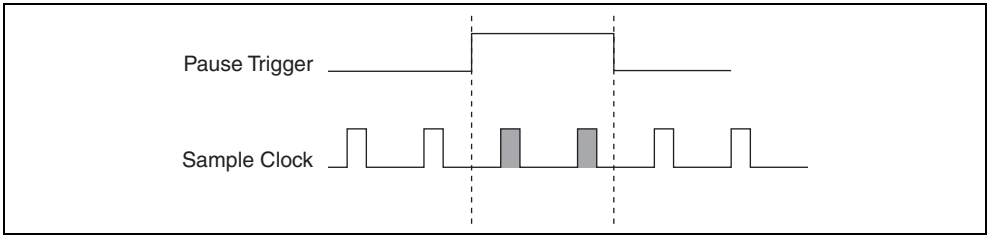


Figure 19. do/PauseTrigger with Other Signal Source

Using a Digital Source

To use do/PauseTrigger, specify a source and a polarity. The source can be a PFI signal or one of several other internal signals on the NI cDAQ-9178/9174 chassis.

You also can specify whether the samples are paused when do/PauseTrigger is at a logic high or low level. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Using an Analog Source

Some C Series I/O modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event, depending on the trigger properties.

When you use an analog trigger source, the samples are paused when the Analog Comparison Event signal is at a high or low level, depending on the trigger properties. The analog trigger circuit must be configured by a simultaneously running analog input task.

Getting Started with DO Applications in Software

You can use the NI cDAQ-9178/9174 chassis in the following digital output applications:

- Single-Point (On-Demand) Generation
- Finite Generation
- Continuous Generation

For more information about programming digital output applications and triggers in software, refer the *LabVIEW Help* or to the *NI-DAQmx Help*.

The *NI-DAQmx Help* is available after installation from **Start»All Programs»National Instruments» NI-DAQ»NI-DAQmx Help**. To view the *LabVIEW Help,* select **Help»Search the LabVIEW Help** in LabVIEW. Alternately, to download the *LabVIEW Help,* go to `ni.com/manuals`.

Digital Input/Output Configuration for NI 9401

When you change the configuration of lines on a NI 9401 digital I/O module between input and output, NI-DAQmx temporarily reserves all of the lines on the module for communication to send the module a line configuration command. If another task or route is actively using the module, to avoid interfering with the other task, NI-DAQmx generates an error instead of sending the line configuration command. During the line configuration command, the output lines are maintained without glitching.

PFI

You can configure channels of a hardware-timed digital module as Programmable Function Interface (PFI) terminals. Hardware-timed digital modules can be installed in up to two chassis slots. The NI cDAQ-9178 chassis also provides two terminals for PFI.

You can configure each PFI individually as the following:

- Timing input signal for AI, AO, DI, DO, or counter/timer functions
- Timing output signal from AI, AO, DI, DO, or counter/timer functions

Counters

The NI cDAQ-9178/9174 has four general-purpose 32-bit counter/timers and one frequency generator. The general-purpose counter/timers can be used for many measurement and pulse generation applications. Figure 20 shows the NI cDAQ-9178/9174 Counter 0 and the frequency generator. All four counters on the NI cDAQ-9178/9174 are identical.

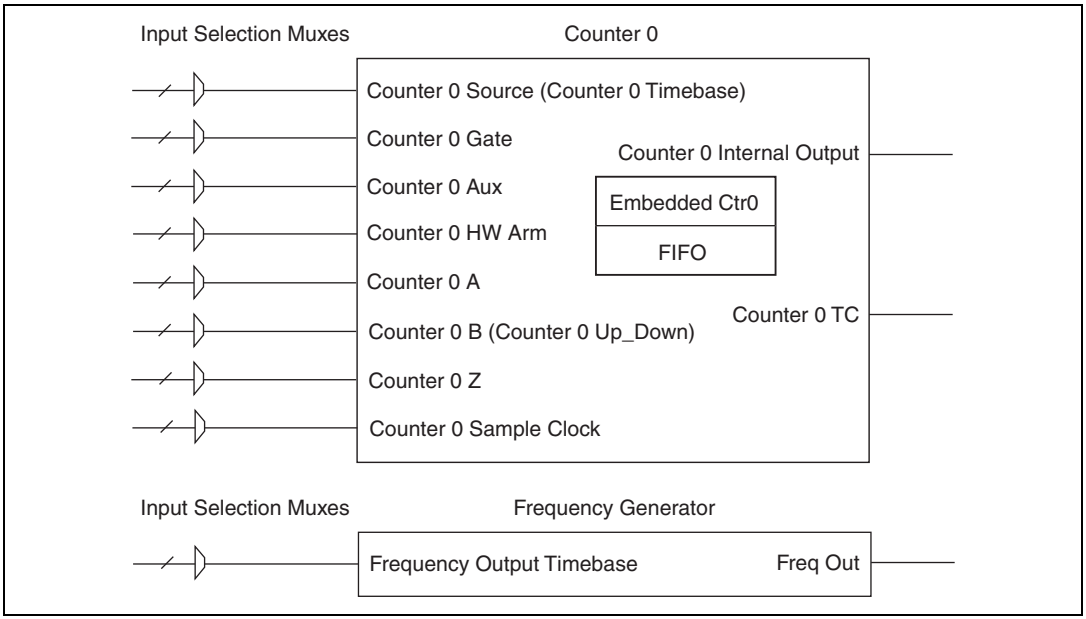


Figure 20. NI cDAQ-9178/9174 Counter 0 and Frequency Generator

Counters have eight input signals, although in most applications only a few inputs are used.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Each counter has a FIFO that can be used for buffered acquisition and generation. Each counter also contains an embedded counter (Embedded Ctr*n*) for use in what are traditionally two-counter measurements and generations. The embedded counters cannot be programmed independent of the main counter; signals from the embedded counters are not routable.

Counter Timing Engine

Unlike analog input, analog output, digital input, and digital output, NI cDAQ-9178/9174 counters do not have the ability to divide down a timebase to produce an internal counter sample clock. For sample

clocked operations, an external signal must be provided to supply a clock source. The source can be any of the following signals:

- AI Sample Clock
- AI Start Trigger
- AI Reference Trigger
- AO Sample Clock
- DI Sample Clock
- DI Start Trigger
- DO Sample Clock
- CTR *n* Internal Output
- Freq Out
- PFI
- Change Detection Event
- Analog Comparison Event

Not all timed counter operations require a sample clock. For example, a simple buffered pulse width measurement latches in data on each edge of a pulse. For this measurement, the measured signal determines when data is latched in. These operations are referred to as implicit timed operations. However, many of the same measurements can be clocked at an interval with a sample clock. These are referred to as sample clocked operations. Table 4 shows the different options for the different measurements.

Table 4. Counter Timing Measurements

Measurement	Implicit Timing Support	Sample Clocked Timing Support
Buffered Edge Count	No	Yes
Buffered Pulse Width	Yes	Yes
Buffered Pulse	Yes	Yes
Buffered Semi-Period	Yes	No
Buffered Frequency	Yes	Yes
Buffered Period	Yes	Yes
Buffered Position	No	Yes
Buffered Two-Signal Edge Separation	Yes	Yes

Counter Input Applications

The following sections list the various counter input applications available on the NI cDAQ-9178/9174:

- *Counting Edges*
- *Pulse-Width Measurement*
- *Pulse Measurement*
- *Semi-Period Measurement*

- *Frequency Measurement*
- *Period Measurement*
- *Position Measurement*

Counting Edges

In edge counting applications, the counter counts edges on its Source after the counter is armed. You can configure the counter to count rising or falling edges on its Source input. You also can control the direction of counting (up or down), as described in the *Controlling the Direction of Counting* section. The counter values can be read on demand or with a sample clock.

Refer to the following sections for more information about edge counting options:

- *Single Point (On-Demand) Edge Counting*
- *Buffered (Sample Clock) Edge Counting*

Single Point (On-Demand) Edge Counting

With single point (on-demand) edge counting, the counter counts the number of edges on the Source input after the counter is armed. On-demand refers to the fact that software can read the counter contents at any time without disturbing the counting process. Figure 21 shows an example of single point edge counting.

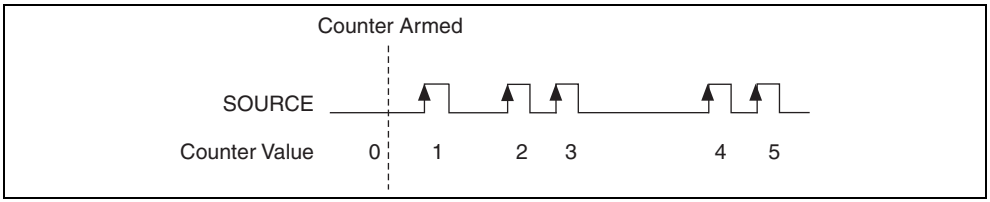


Figure 21. Single Point (On-Demand) Edge Counting

You also can use a pause trigger to pause (or gate) the counter. When the pause trigger is active, the counter ignores edges on its Source input. When the pause trigger is inactive, the counter counts edges normally.

You can route the pause trigger to the Gate input of the counter. You can configure the counter to pause counting when the pause trigger is high or when it is low. Figure 22 shows an example of on-demand edge counting with a pause trigger.

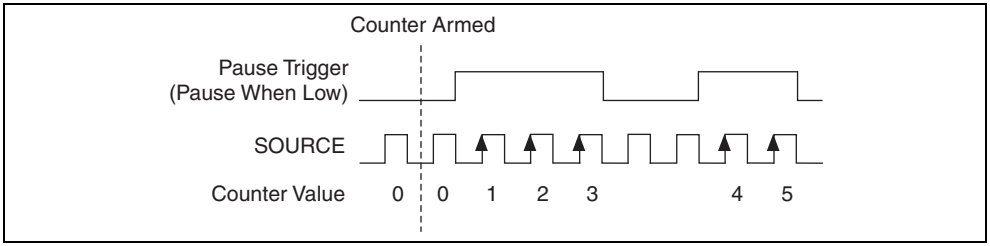


Figure 22. Single Point (On-Demand) Edge Counting with Pause Trigger

Buffered (Sample Clock) Edge Counting

With buffered edge counting (edge counting using a sample clock), the counter counts the number of edges on the Source input after the counter is armed. The value of the counter is sampled on each active edge of a sample clock and stored in the FIFO. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream.

The count values returned are the cumulative counts since the counter armed event. That is, the sample clock does not reset the counter. You can configure the counter to sample on the rising or falling edge of the sample clock.

Figure 23 shows an example of buffered edge counting. Notice that counting begins when the counter is armed, which occurs before the first active edge on Sample Clock.

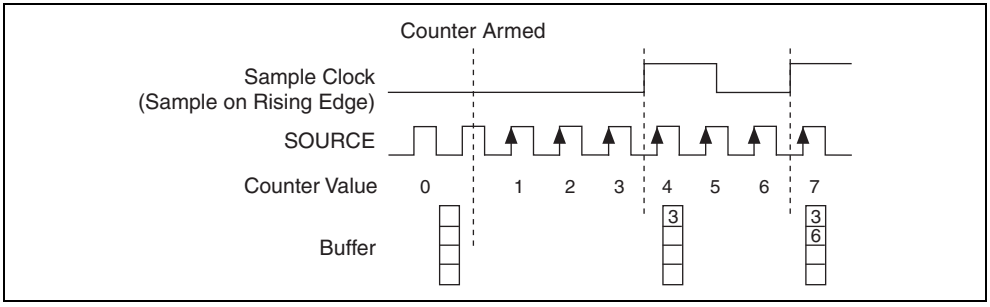


Figure 23. Buffered (Sample Clock) Edge Counting

Controlling the Direction of Counting

In edge counting applications, the counter can count up or down. You can configure the counter to do the following:

- Always count up
- Always count down
- Count up when the Counter 0 B input is high; count down when it is low

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Pulse-Width Measurement

In pulse-width measurements, the counter measures the width of a pulse on its Gate input signal. You can configure the counter to measure the width of high pulses or low pulses on the Gate signal.

You can route an internal or external periodic clock signal (with a known period) to the Source input of the counter. The counter counts the number of rising (or falling) edges on the Source signal while the pulse on the Gate signal is active.

You can calculate the pulse width by multiplying the period of the Source signal by the number of edges returned by the counter.

A pulse-width measurement will be accurate even if the counter is armed while a pulse train is in progress. If a counter is armed while the pulse is in the active state, it will wait for the next transition to the active state to begin the measurement.

Refer to the following sections for more information about NI cDAQ-9178/9174 pulse-width measurement options:

- *Single Pulse-Width Measurement*
- *Implicit Buffered Pulse-Width Measurement*
- *Sample Clocked Buffered Pulse-Width Measurement*

Single Pulse-Width Measurement

With single pulse-width measurement, the counter counts the number of edges on the Source input while the Gate input remains active. When the Gate input goes inactive, the counter stores the count in the FIFO and ignores other edges on the Gate and Source inputs. Software then reads the stored count.

Figure 24 shows an example of a single pulse-width measurement.

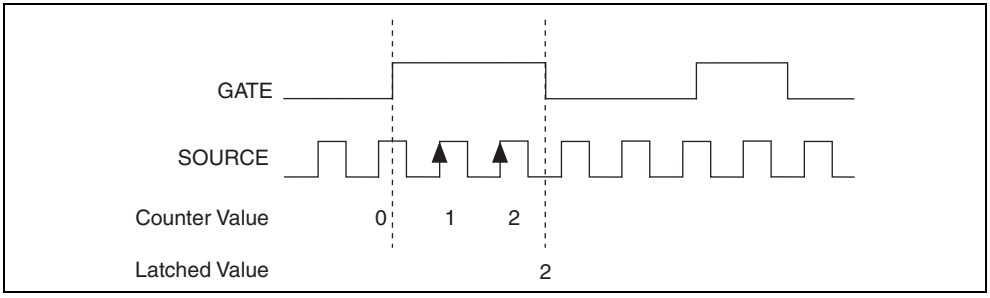


Figure 24. Single Pulse-Width Measurement

Implicit Buffered Pulse-Width Measurement

An implicit buffered pulse-width measurement is similar to single pulse-width measurement, but buffered pulse-width measurement takes measurements over multiple pulses.

The counter counts the number of edges on the Source input while the Gate input remains active. On each trailing edge of the Gate signal, the counter stores the count in the counter FIFO. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 25 shows an example of an implicit buffered pulse-width measurement.

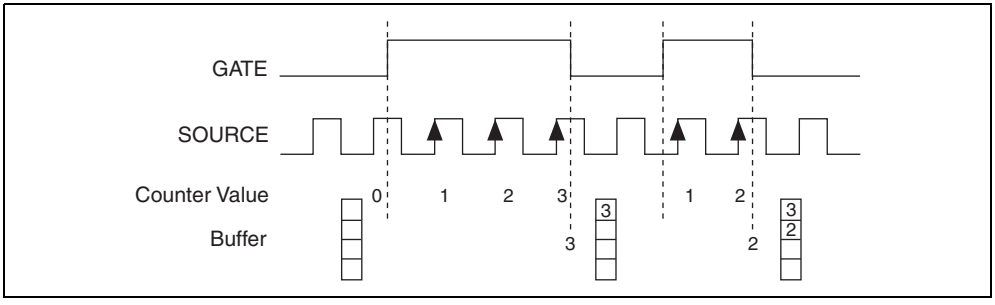


Figure 25. Implicit Buffered Pulse-Width Measurement

Sample Clocked Buffered Pulse-Width Measurement

A sample clocked buffered pulse-width measurement is similar to single pulse-width measurement, but buffered pulse-width measurement takes measurements over multiple pulses correlated to a sample clock.

The counter counts the number of edges on the Source input while the Gate input remains active. On each sample clock edge, the counter stores the count in the FIFO of the last pulse width to complete. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 26 shows an example of a sample clocked buffered pulse-width measurement.

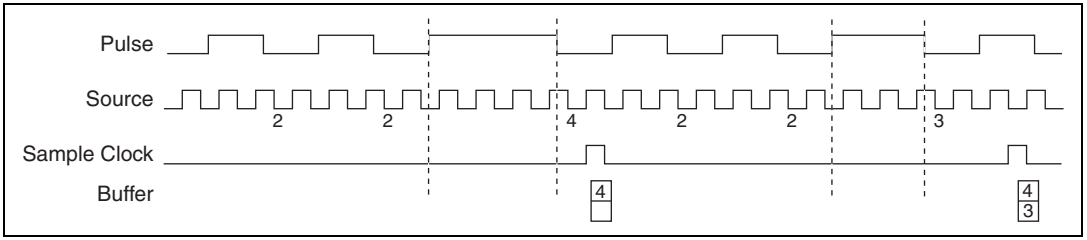


Figure 26. Sample Clocked Buffered Pulse-Width Measurement

Note If a pulse does not occur between sample clocks, an overrun error occurs.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Pulse Measurement

In pulse measurements, the counter measures the high and low time of a pulse on its Gate input signal after the counter is armed. A pulse is defined in terms of its high and low time, high and low ticks or frequency and duty cycle. This is similar to the pulse-width measurement, except that the inactive pulse is measured as well.

You can route an internal or external periodic clock signal (with a known period) to the Source input of the counter. The counter counts the number of rising (or falling) edges occurring on the Source input between two edges of the Gate signal.

You can calculate the high and low time of the Gate input by multiplying the period of the Source signal by the number of edges returned by the counter.

Refer to the following sections for more information about NI cDAQ-9178/9174 pulse measurement options:

- *Single Pulse Measurement*
- *Implicit Buffered Pulse Measurement*
- *Sample Clocked Buffered Pulse Measurement*

Single Pulse Measurement

Single (on-demand) pulse measurement is equivalent to two single pulse-width measurements on the high (H) and low (L) ticks of a pulse, as shown in Figure 27.

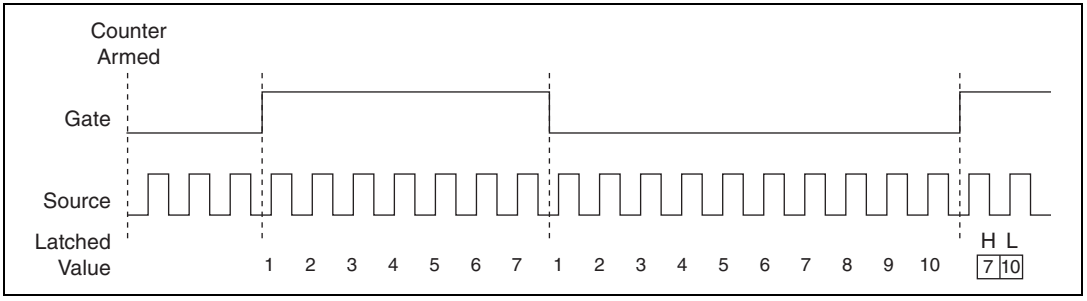


Figure 27. Single (On-Demand) Pulse Measurement

Implicit Buffered Pulse Measurement

In an implicit buffered pulse measurement, on each edge of the Gate signal, the counter stores the count in the FIFO. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream.

The counter begins counting when it is armed. The arm usually occurs between edges on the Gate input but the counting does not start until the desired edge. You can select whether to read the high pulse or low pulse first using the **StartingEdge** property in NI-DAQmx.

Figure 28 shows an example of an implicit buffered pulse measurement.

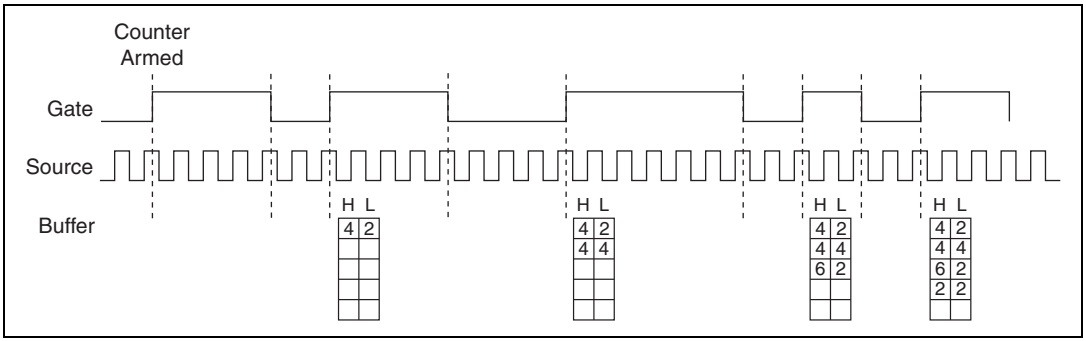


Figure 28. Implicit Buffered Pulse Measurement

Sample Clocked Buffered Pulse Measurement

A sample clocked buffered pulse measurement is similar to single pulse measurement, but a buffered pulse measurement takes measurements over multiple pulses correlated to a sample clock.

The counter performs a pulse measurement on the Gate. On each sample clock edge, the counter stores the high and low ticks in the FIFO of the last pulse to complete. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 29 shows an example of a sample clocked buffered pulse measurement.

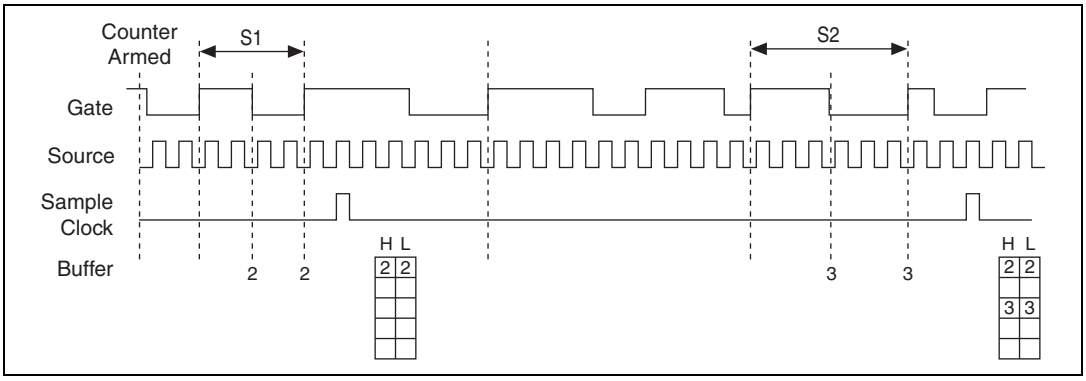


Figure 29. Sample Clocked Buffered Pulse Measurement

Note If a pulse does not occur between sample clocks, an overrun error occurs.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Pulse versus Semi-Period Measurements

In hardware, pulse measurement and semi-period are the same measurement. Both measure the high and low times of a pulse. The functional difference between the two measurements is how the data is returned. In a semi-period measurement, each high or low time is considered one point of data and returned in units of seconds or ticks. In a pulse measurement, each pair of high and low times is considered one point of data and returned as a paired sample in units of frequency and duty cycle, high and low time or high and low ticks. When reading data, 10 points in a semi-period measurement will get an array of five high times and five low times. When you read 10 points in a pulse measurement, you get an array of 10 pairs of high and low times.

Also, pulse measurements support sample clock timing while semi-period measurements do not.

Semi-Period Measurement

In semi-period measurements, the counter measures a semi-period on its Gate input signal after the counter is armed. A semi-period is the time between any two consecutive edges on the Gate input.

You can route an internal or external periodic clock signal (with a known period) to the Source input of the counter. The counter counts the number of rising (or falling) edges occurring on the Source input between two edges of the Gate signal.

You can calculate the semi-period of the Gate input by multiplying the period of the Source signal by the number of edges returned by the counter.

Refer to the following sections for more information about semi-period measurement options:

- *Single Semi-Period Measurement*
- *Implicit Buffered Semi-Period Measurement*

Refer to the *Pulse versus Semi-Period Measurements* section for information about the differences between semi-period measurement and pulse measurement.

Single Semi-Period Measurement

Single semi-period measurement is equivalent to single pulse-width measurement.

Implicit Buffered Semi-Period Measurement

In implicit buffered semi-period measurement, on each edge of the Gate signal, the counter stores the count in the FIFO. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream.

The counter begins counting when it is armed. The arm usually occurs between edges on the Gate input. You can select whether to read the first active low or active high semi period using the **CI.SemiPeriod.StartingEdge** property in NI-DAQmx.

Figure 30 shows an example of an implicit buffered semi-period measurement.

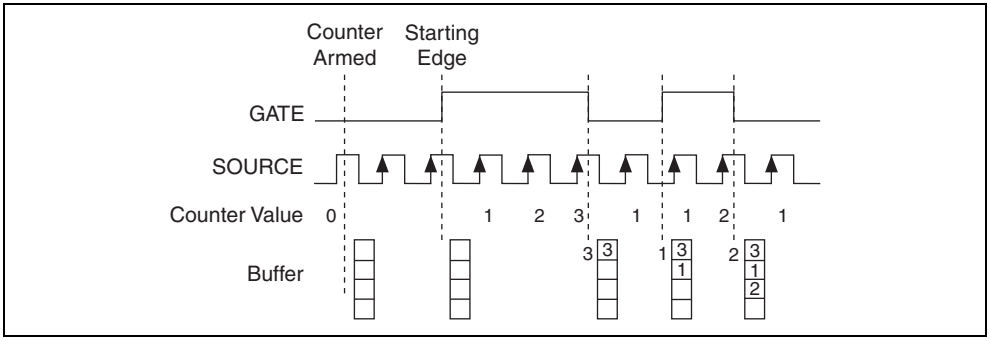


Figure 30. Implicit Buffered Semi-Period Measurement

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Frequency Measurement

You can use the counters to measure frequency in several different ways. Refer to the following sections for information about NI cDAQ-9178/9174 frequency measurement options:

- *Low Frequency with One Counter*
- *High Frequency with Two Counters*
- *Large Range of Frequencies with Two Counters*
- *Sample Clocked Buffered Frequency Measurement*

Low Frequency with One Counter

For low frequency measurements with one counter, you measure one period of your signal using a known timebase.

You can route the signal to measure (*fx*) to the Gate of a counter. You can route a known timebase (*fk*) to the Source of the counter. The known timebase can be an onboard timebase, such as 80 MHz Timebase, 20 MHz Timebase, or 100 kHz Timebase, or any other signal with a known rate.

You can configure the counter to measure one period of the gate signal. The frequency of *fx* is the inverse of the period. Figure 31 illustrates this method.

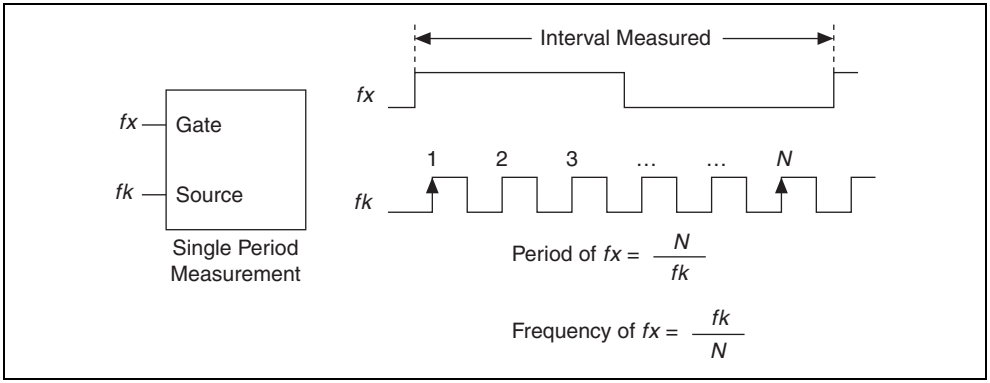


Figure 31. Low Frequency with One Counter

High Frequency with Two Counters

For high frequency measurements with two counters, you measure one pulse of a known width using your signal and derive the frequency of your signal from the result.

Note Counter 0 is always paired with Counter 1. Counter 2 is always paired with Counter 3.

In this method, you route a pulse of known duration (*T*) to the Gate of a counter. You can generate the pulse using a second counter. You also can generate the pulse externally and connect it to a PFI terminal. You only need to use one counter if you generate the pulse externally.

Route the signal to measure (*fx*) to the Source of the counter. Configure the counter for a single pulse-width measurement. If you measure the width of pulse *T* to be *N* periods of *fx*, the frequency of *fx* is *N*/*T*.

Figure 32 illustrates this method. Another option is to measure the width of a known period instead of a known pulse.

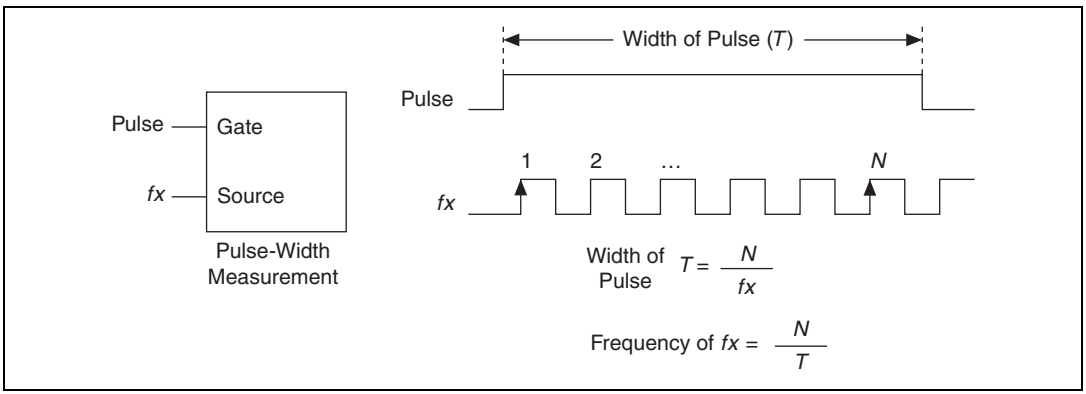


Figure 32. High Frequency with Two Counters

Large Range of Frequencies with Two Counters

By using two counters, you can accurately measure a signal that might be high or low frequency. This technique is called reciprocal frequency measurement. When measuring a large range of frequencies with two counters, you generate a long pulse using the signal to measure. You then measure the long pulse with a known timebase. The NI cDAQ-9178/9174 can measure this long pulse more accurately than the faster input signal.

Note Counter 0 is always paired with Counter 1. Counter 2 is always paired with Counter 3.

You can route the signal to measure to the Source input of Counter 0, as shown in Figure 33. Assume this signal to measure has frequency *fx*. NI-DAQmx automatically configures Counter 0 to generate a single pulse that is the width of *N* periods of the source input signal.

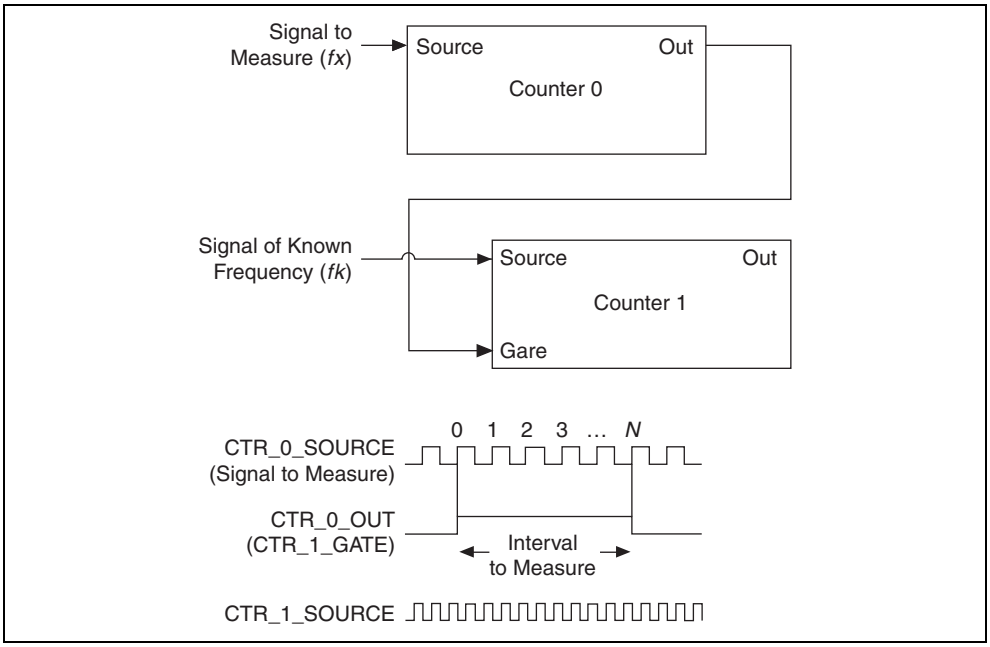


Figure 33. Large Range of Frequencies with Two Counters

Next, route the Counter 0 Internal Output signal to the Gate input of Counter 1. You can route a signal of known frequency (*fk*) to the Counter 1 Source input. Configure Counter 1 to perform a single pulse-width measurement. Suppose the result is that the pulse width is *J* periods of the *fk* clock.

From Counter 0, the length of the pulse is *N*/*fx*. From Counter 1, the length of the same pulse is *J*/*fk*. Therefore, the frequency of *fx* is given by *fx* = *fk* * (*N*/*J*).

Sample Clocked Buffered Frequency Measurement

Sample clocked buffered point frequency measurements can either be a single frequency measurement or an average between sample clocks. Use **CI.Freq.EnableAveraging** to set the behavior. For buffered frequency, the default is True.

A sample clocked buffered frequency measurement with **CI.Freq.EnableAveraging** set to True uses the embedded counter and a sample clock to perform a frequency measurement. For each sample clock period, the embedded counter counts the signal to measure (*fx*) and the primary counter counts the internal time-base of a known frequency (*fk*). Suppose T1 is the number of ticks of the unknown signal counted between sample clocks and T2 is the number of ticks counted of the known timebase as shown in Figure 34. The frequency measured is:

$$fx = fk * (T1/T2)$$

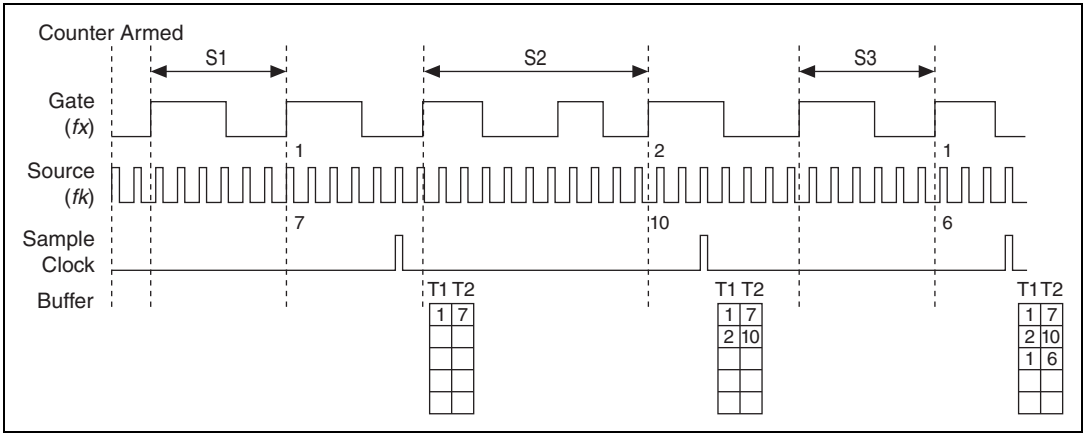


Figure 34. Sample Clocked Buffered Frequency Measurement (Averaging)

When **CI.Freq.EnableAveraging** is set to false, the frequency measurement returns the frequency of the pulse just before the sample clock. This single measurement is a single frequency measurement and is not an average between clocks as shown in Figure 35.

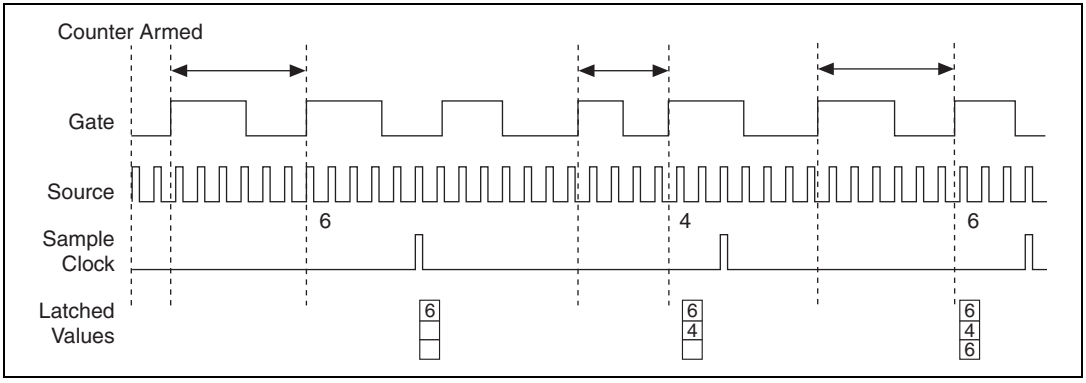


Figure 35. Sample Clocked Buffered Frequency Measurement (Non-Averaging)

With sample clocked frequency measurements, ensure that the frequency to measure is twice as fast as the sample clock to prevent a measurement overflow.

Choosing a Method for Measuring Frequency

The best method to measure frequency depends on several factors including the expected frequency of the signal to measure, the desired accuracy, how many counters are available, and how long the measurement can take. For all frequency measurement methods, assume the following:

fx	is the frequency to be measured if no error
fk	is the known source or gate frequency
measurement time (*T*)	is the time it takes to measure a single sample

Divide down (*N*)	is the integer to divide down measured frequency, only used in large range two counters
fs	is the sample clock rate, only used in sample clocked frequency measurements

Here is how these variables apply to each method, summarized in Table 5.

- **One counter**—With one counter measurements, a known timebase is used for the source frequency (*fk*). The measurement time is the period of the frequency to be measured, or 1/*fx*.
- **Two counter high frequency**—With the two counter high frequency method, the second counter provides a known measurement time. The gate frequency equals 1/*measurement time*.
- **Two counter large range**—The two counter larger range measurement is the same as a one counter measurement, but now the user has an integer divide down of the signal. An internal timebase is still used for the source frequency (*fk*), but the divide down means that the measurement time is the period of the divided down signal, or *N*/*fx* where *N* is the divide down.
- **Sample clocked**—For sample clocked frequency measurements, a known timebase is counted for the source frequency (*fk*). The measurement time is the period of the sample clock (*fs*).

Table 5. Frequency Measurement Methods

Variable	Sample Clocked	One Counter	Two Counter	
			High Frequency	Large Range
fk	Known timebase	Known timebase	$\frac{1}{\text{ating perio}}$	Known timebase
Measurement time	$\frac{1}{fs}$	$\frac{1}{fx}$	gating period	$\frac{N}{fx}$
Max. frequency error	$fx \times \frac{fx}{fk \times \left\lfloor \frac{fx}{fs} - 1 \right\rfloor}$	$fx \times \frac{fx}{fk - fx}$	*fk*	$fx \times \frac{fx}{N \times fk - fx}$
Max. error %	$\frac{fx}{fk \times \left\lfloor \frac{fx}{fs} - 1 \right\rfloor}$	$\frac{fx}{fk - fx}$	$\frac{fk}{fx}$	$\frac{fx}{N \times fk - fx}$

Note: Accuracy equations do not take clock stability into account. Refer to your device specifications for information about clock stability.

Which Method Is Best?

This depends on the frequency to be measured, the rate at which you want to monitor the frequency and the accuracy you desire. Take for example, measuring a 50 kHz signal. Assuming that the measurement times for the sample clocked (with averaging) and two counter frequency measurements are configured the same, Table 6 summarizes the results.

Table 6. 50 kHz Frequency Measurement Methods

Variable	Sample Clocked	One Counter	Two Counter	
			High Frequency	Large Range
fx	50,000	50,000	50,000	50,000
fk	80 M	80 M	1,000	80 M
Measurement time (*mS*)	1	.02	1	1
N	—	—	—	50
Max. frequency error (Hz)	.638	31.27	1,000	.625
Max. error %	.00128	.0625	2	.00125

From this, you can see that while the measurement time for one counter is shorter, the accuracy is best in the sample clocked and two counter large range measurements. For another example, Table 7 shows the results for 5 MHz.

Table 7. 5 MHz Frequency Measurement Methods

Variable	Sample Clocked	One Counter	Two Counter	
			High Frequency	Large Range
fx	5 M	5 M	5 M	5 M
fk	80 M	80 M	1,000	80 M
Measurement time (*mS*)	1	.0002	1	1
N	—	—	—	5,000
Max. Frequency error (Hz)	62.51	333 k	1,000	62.50
Max. Error %	.00125	6.67	.02	.00125

Again the measurement time for the one counter measurement is lowest but the accuracy is lower. Note that the accuracy and measurement time of the sample clocked and two counter large range are almost the same. The advantage of the sample clocked method is that even when the frequency to measure changes, the measurement time does not and error percentage varies little. For example, if you configured a large range two counter measurement to use a divide down of 50 for a 50 k signal, then you would get the accuracy measurement time and accuracy listed in Table 6. But if your signal ramped up to 5 M, then with a divide down of 50, your measurement time is 0.01 ms, but your error is now 0.125%. The error with a sample clocked frequency measurement is not as dependent on the measured frequency so at 50 k and 5 M with a measurement time of 1 ms the error percentage is still close to 0.00125%. One of the disadvantages of a sample clocked frequency measurement is that the frequency to be

measured must be at least twice the sample clock rate to ensure that a full period of the frequency to be measured occurs between sample clocks.

- Low frequency measurements with one counter is a good method for many applications. However, the accuracy of the measurement decreases as the frequency increases.
- High frequency measurements with two counters is accurate for high frequency signals. However, the accuracy decreases as the frequency of the signal to measure decreases. At very low frequencies, this method may be too inaccurate for your application. Another disadvantage of this method is that it requires two counters (if you cannot provide an external signal of known width). An advantage of high frequency measurements with two counters is that the measurement completes in a known amount of time.
- Measuring a large range of frequencies with two counters measures high and low frequency signals accurately. However, it requires two counters, and it has a variable sample time and variable error % dependent on the input signal.

Table 8 summarizes some of the differences in methods of measuring frequency.

Table 8. Frequency Measurement Method Comparison

Method	Number of Counters Used	Number of Measurements Returned	Measures High Frequency Signals Accurately	Measures Low Frequency Signals Accurately
Low frequency with one counter	1	1	Poor	Good
High frequency with two counters	1 or 2	1	Good	Poor
Large range of frequencies with two counters	2	1	Good	Good
Sample clocked (averaged)	1	1	Good	Good

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Period Measurement

In period measurements, the counter measures a period on its Gate input signal after the counter is armed. You can configure the counter to measure the period between two rising edges or two falling edges of the Gate input signal.

You can route an internal or external periodic clock signal (with a known period) to the Source input of the counter. The counter counts the number of rising (or falling) edges occurring on the Source input between the two active edges of the Gate signal.

You can calculate the period of the Gate input by multiplying the period of the Source signal by the number of edges returned by the counter.

Period measurements return the inverse results of frequency measurements. Refer to the *Frequency Measurement* section for more information.

Position Measurement

You can use the counters to perform position measurements with quadrature encoders or two-pulse encoders. You can measure angular position with X1, X2, and X4 angular encoders. Linear position can be measured with two-pulse encoders. You can choose to do either a single point (on-demand) position measurement or a buffered (sample clock) position measurement. You must arm a counter to begin position measurements.

Refer to the following sections for more information about the NI cDAQ-9178/9174 position measurement options:

- *Measurements Using Quadrature Encoders*
- *Measurements Using Two Pulse Encoders*
- *Buffered (Sample Clock) Position Measurement*

Measurements Using Quadrature Encoders

The counters can perform measurements of quadrature encoders that use X1, X2, or X4 encoding. A quadrature encoder can have up to three channels—channels A, B, and Z.

- **X1 Encoding**—When channel A leads channel B in a quadrature cycle, the counter increments. When channel B leads channel A in a quadrature cycle, the counter decrements. The amount of increments and decrements per cycle depends on the type of encoding—X1, X2, or X4.

  Figure 36 shows a quadrature cycle and the resulting increments and decrements for X1 encoding. When channel A leads channel B, the increment occurs on the rising edge of channel A. When channel B leads channel A, the decrement occurs on the falling edge of channel A.

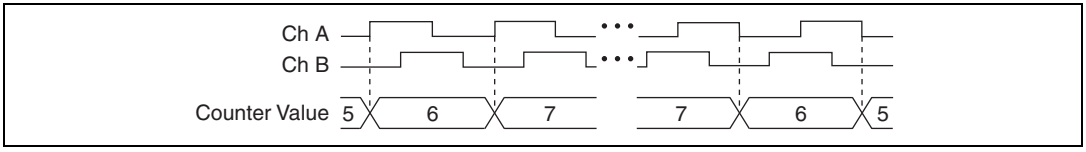


Figure 36. X1 Encoding

- **X2 Encoding**—The same behavior holds for X2 encoding except the counter increments or decrements on each edge of channel A, depending on which channel leads the other. Each cycle results in two increments or decrements, as shown in Figure 37.

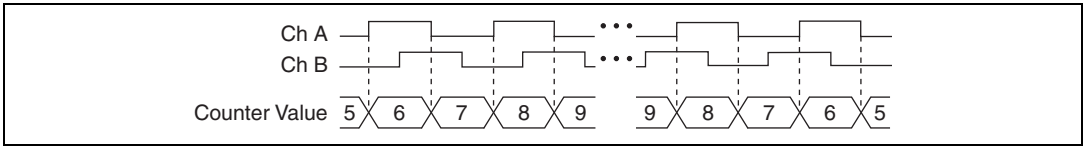


Figure 37. X2 Encoding

- **X4 Encoding**—Similarly, the counter increments or decrements on each edge of channels A and B for X4 encoding. Whether the counter increments or decrements depends on which channel leads the other. Each cycle results in four increments or decrements, as shown in Figure 38.

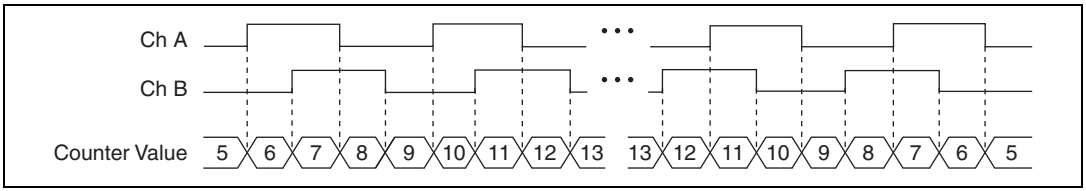


Figure 38. X4 Encoding

Channel Z Behavior

Some quadrature encoders have a third channel, channel Z, which is also referred to as the index channel. A high level on channel Z causes the counter to be reloaded with a specified value in a specified phase of the quadrature cycle. You can program this reload to occur in any one of the four phases in a quadrature cycle.

Channel Z behavior—when it goes high and how long it stays high—differs with quadrature encoder designs. You must refer to the documentation for your quadrature encoder to obtain timing of channel Z with respect to channels A and B. You must then ensure that channel Z is high during at least a portion of the phase you specify for reload. For instance, in Figure 39, channel Z is never high when channel A is high and channel B is low. Thus, the reload must occur in some other phase.

In Figure 39, the reload phase is when both channel A and channel B are low. The reload occurs when this phase is true and channel Z is high. Incrementing and decrementing takes priority over reloading. Thus, when the channel B goes low to enter the reload phase, the increment occurs first. The reload occurs within one maximum timebase period after the reload phase becomes true. After the reload occurs, the counter continues to count as before. The figure illustrates channel Z reload with X4 decoding.

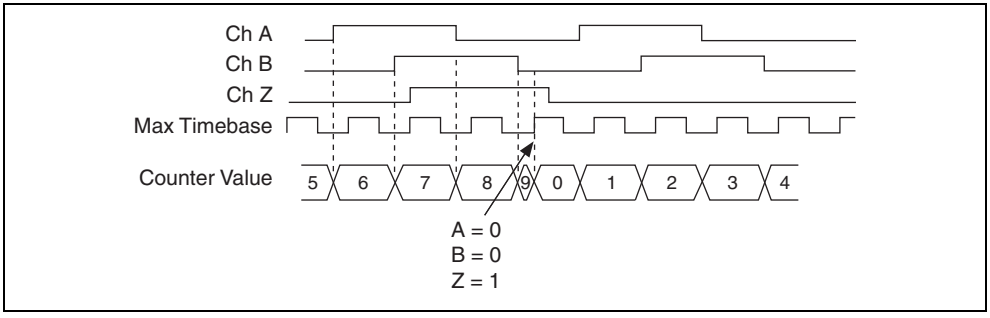


Figure 39. Channel Z Reload with X4 Decoding

Measurements Using Two Pulse Encoders

The counter supports two pulse encoders that have two channels—channels A and B.

The counter increments on each rising edge of channel A. The counter decrements on each rising edge of channel B, as shown in Figure 40.

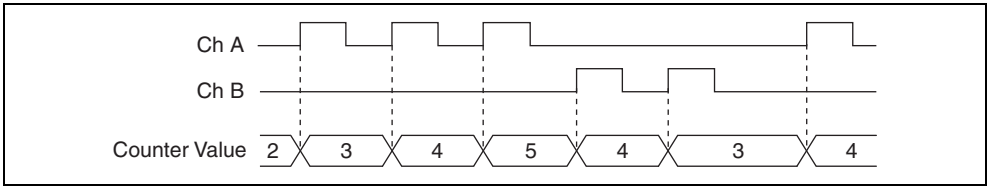


Figure 40. Measurements Using Two Pulse Encoders

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Buffered (Sample Clock) Position Measurement

With buffered position measurement (position measurement using a sample clock), the counter increments based on the encoding used after the counter is armed. The value of the counter is sampled on each active edge of a sample clock. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream. The count values returned are the cumulative counts since the counter armed event; that is, the sample clock does not reset the counter. You can route the counter sample clock to the Gate input of the counter. You can configure the counter to sample on the rising or falling edge of the sample clock.

Figure 41 shows an example of a buffered X1 position measurement.

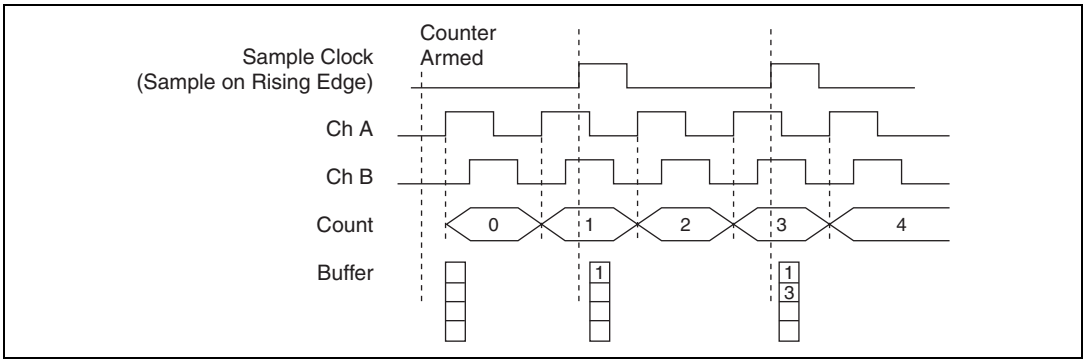


Figure 41. Buffered Position Measurement

Two-Signal Edge-Separation Measurement

Two-signal edge-separation measurement is similar to pulse-width measurement, except that there are two measurement signals—Aux and Gate. An active edge on the Aux input starts the counting and an active edge on the Gate input stops the counting. You must arm a counter to begin a two edge separation measurement.

After the counter has been armed and an active edge occurs on the Aux input, the counter counts the number of rising (or falling) edges on the Source. The counter ignores additional edges on the Aux input.

The counter stops counting upon receiving an active edge on the Gate input. The counter stores the count in the FIFO.

You can configure the rising or falling edge of the Aux input to be the active edge. You can configure the rising or falling edge of the Gate input to be the active edge.

Use this type of measurement to count events or measure the time that occurs between edges on two signals. This type of measurement is sometimes referred to as start/stop trigger measurement, second gate measurement, or A-to-B measurement.

Refer to the following sections for more information about the NI cDAQ-9178/9174 edge-separation measurement options:

- *Single Two-Signal Edge-Separation Measurement*
- *Implicit Buffered Two-Signal Edge-Separation Measurement*
- *Sample Clocked Buffered Two-Signal Separation Measurement*

Single Two-Signal Edge-Separation Measurement

With single two-signal edge-separation measurement, the counter counts the number of rising (or falling) edges on the Source input occurring between an active edge of the Gate signal and an active edge of the Aux signal. The counter then stores the count in the FIFO and ignores other edges on its inputs. Software then reads the stored count.

Figure 42 shows an example of a single two-signal edge-separation measurement.

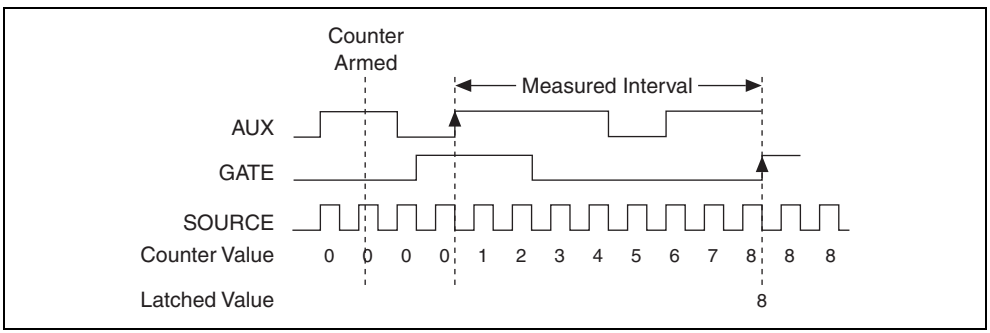


Figure 42. Single Two-Signal Edge-Separation Measurement

Implicit Buffered Two-Signal Edge-Separation Measurement

Implicit buffered and single two-signal edge-separation measurements are similar, but implicit buffered measurement measures multiple intervals.

The counter counts the number of rising (or falling) edges on the Source input occurring between an active edge of the Gate signal and an active edge of the Aux signal. The counter then stores the count in the FIFO. On the next active edge of the Gate signal, the counter begins another measurement. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 43 shows an example of an implicit buffered two-signal edge-separation measurement.

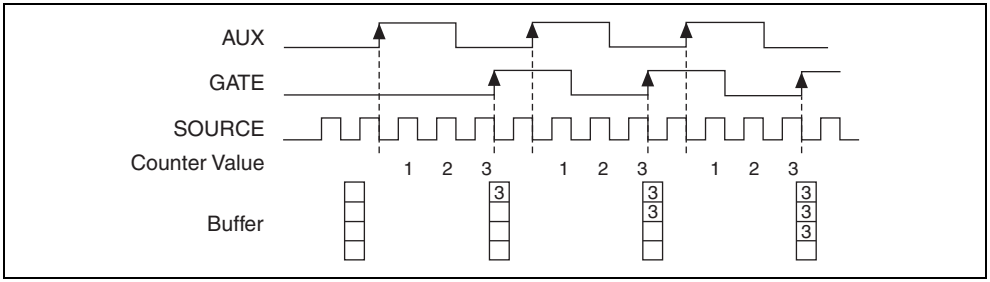


Figure 43. Implicit Buffered Two-Signal Edge-Separation Measurement

Sample Clocked Buffered Two-Signal Separation Measurement

A sample clocked buffered two-signal separation measurement is similar to single two-signal separation measurement, but buffered two-signal separation measurement takes measurements over multiple intervals correlated to a sample clock. The counter counts the number of rising (or falling) edges on the Source input occurring between an active edge of the Gate signal and an active edge of the Aux signal. The counter then stores the count in the FIFO on a sample clock edge. On the next active edge of the Gate signal, the counter begins another measurement. The USB-STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 44 shows an example of a sample clocked buffered two-signal separation measurement.

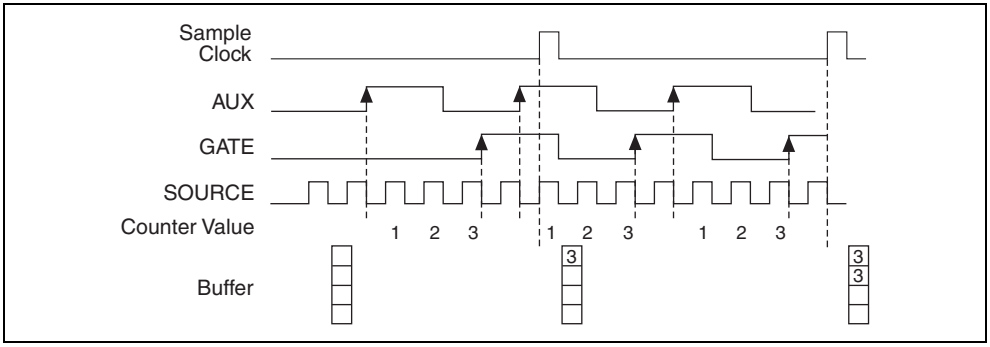


Figure 44. Sample Clocked Buffered Two-Signal Separation Measurement

Note If an active edge on the Gate and an active edge on the Aux does not occur between sample clocks, an overrun error occurs.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Counter Output Applications

The following sections list the various counter output applications available on the NI cDAQ-9178/9174:

- *Simple Pulse Generation*
- *Pulse Train Generation*
- *Frequency Generation*
- *Frequency Division*
- *Pulse Generation for ETS*

Simple Pulse Generation

Refer to the following sections for more information about the NI cDAQ-9178/9174 simple pulse generation options:

- *Single Pulse Generation*
- *Single Pulse Generation with Start Trigger*

Single Pulse Generation

The counter can output a single pulse. The pulse appears on the Counter *n* Internal Output signal of the counter.

You can specify a delay from when the counter is armed to the beginning of the pulse. The delay is measured in terms of a number of active edges of the Source input.

You can specify a pulse width. The pulse width is also measured in terms of a number of active edges of the Source input. You also can specify the active edge of the Source input (rising or falling).

Figure 45 shows a generation of a pulse with a pulse delay of four and a pulse width of three (using the rising edge of Source).

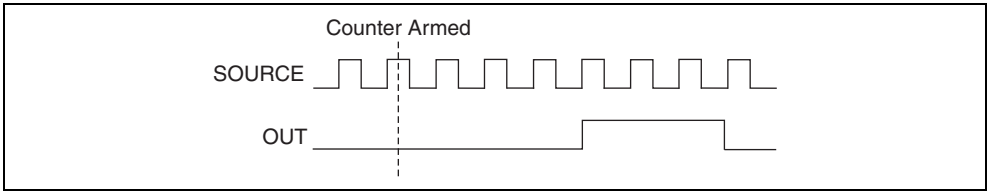


Figure 45. Single Pulse Generation

Single Pulse Generation with Start Trigger

The counter can output a single pulse in response to one pulse on a hardware Start Trigger signal. The pulse appears on the Counter *n* Internal Output signal of the counter.

You can specify a delay from the Start Trigger to the beginning of the pulse. You also can specify the pulse width. The delay is measured in terms of a number of active edges of the Source input.

You can specify a pulse width. The pulse width is also measured in terms of a number of active edges of the Source input. You can also specify the active edge of the Source input (rising and falling).

Figure 46 shows a generation of a pulse with a pulse delay of four and a pulse width of three (using the rising edge of Source).

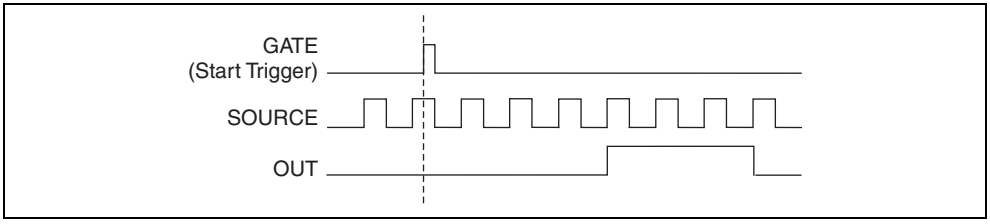


Figure 46. Single Pulse Generation with Start Trigger

Pulse Train Generation

Refer to the following sections for more information about the NI cDAQ-9178/9174 pulse train generation options:

- *Finite Pulse Train Generation*
- *Retriggerable Pulse or Pulse Train Generation*
- *Continuous Pulse Train Generation*
- *Buffered Pulse Train Generation*
- *Finite Implicit Buffered Pulse Train Generation*
- *Continuous Buffered Implicit Pulse Train Generation*
- *Finite Buffered Sample Clocked Pulse Train Generation*
- *Continuous Buffered Sample Clocked Pulse Train Generation*

Finite Pulse Train Generation

This function generates a train of pulses with programmable frequency and duty cycle for a predetermined number of pulses. With NI cDAQ-9178/9174 counters, the primary counter generates the specified pulse train and the embedded counter counts the pulses generated by the primary counter. When the embedded counter reaches the specified tick count, it generates a trigger that stops the primary counter generation.

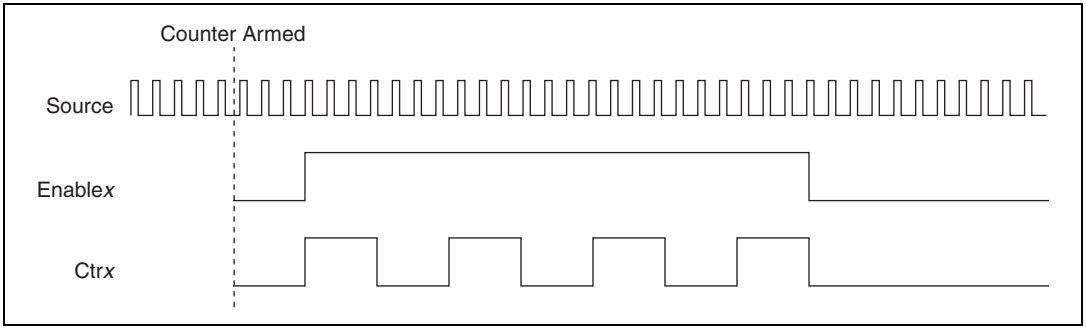


Figure 47. Finite Pulse Train Generation: Four Ticks Initial Delay, Four Pulses

Retriggerable Pulse or Pulse Train Generation

The counter can output a single pulse or multiple pulses in response to each pulse on a hardware Start Trigger signal. The generated pulses appear on the Counter *n* Internal Output signal of the counter.

You can route the Start Trigger signal to the Gate input of the counter. You can specify a delay from the Start Trigger to the beginning of each pulse. You also can specify the pulse width. The delay and pulse width are measured in terms of a number of active edges of the Source input. The initial delay can be applied to only the first trigger or to all triggers using the **CO.EnableInitalDelayOnRetrigger** property. The default for a single pulse is True, while the default for finite pulse trains is False.

The counter ignores the Gate input while a pulse generation is in progress. After the pulse generation is finished, the counter waits for another Start Trigger signal to begin another pulse generation. For retriggered pulse generation, pause triggers are not allowed since the pause trigger also uses the gate input.

Figure 48 shows a generation of two pulses with a pulse delay of five and a pulse width of three (using the rising edge of Source) with **CO.EnableInitalDelayOnRetrigge**r set to the default True.

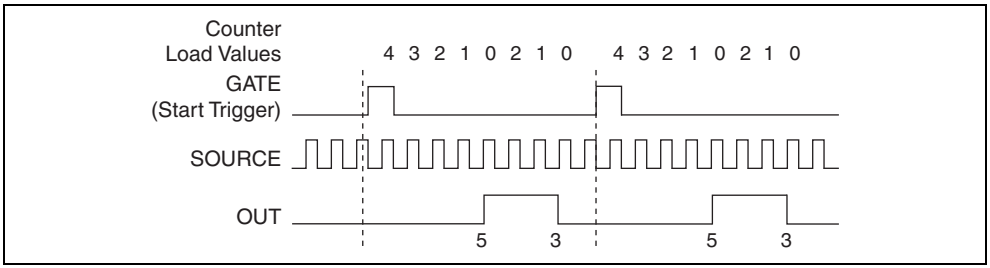


Figure 48. Retriggerable Single Pulse Generation with Initial Delay on Retrigger

Figure 49 shows the same pulse train with **CO.EnableInitalDelayOnRetrigger** set to the default False.

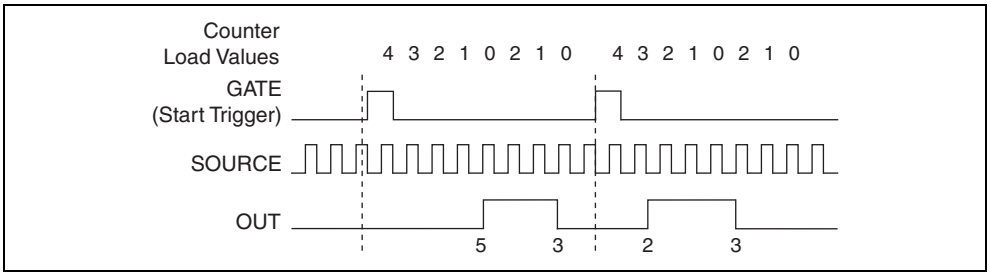


Figure 49. Retriggerable Single Pulse Generation False

Note The minimum time between the trigger and the first active edge is two ticks of the source.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Continuous Pulse Train Generation

This function generates a train of pulses with programmable frequency and duty cycle. The pulses appear on the Counter *n* Internal Output signal of the counter.

You can specify a delay from when the counter is armed to the beginning of the pulse train. The delay is measured in terms of a number of active edges of the Source input.

You specify the high and low pulse widths of the output signal. The pulse widths are also measured in terms of a number of active edges of the Source input. You also can specify the active edge of the Source input (rising or falling).

The counter can begin the pulse train generation as soon as the counter is armed, or in response to a hardware Start Trigger. You can route the Start Trigger to the Gate input of the counter.

You also can use the Gate input of the counter as a Pause Trigger (if it is not used as a Start Trigger). The counter pauses pulse generation when the Pause Trigger is active.

Figure 50 shows a continuous pulse train generation (using the rising edge of Source).

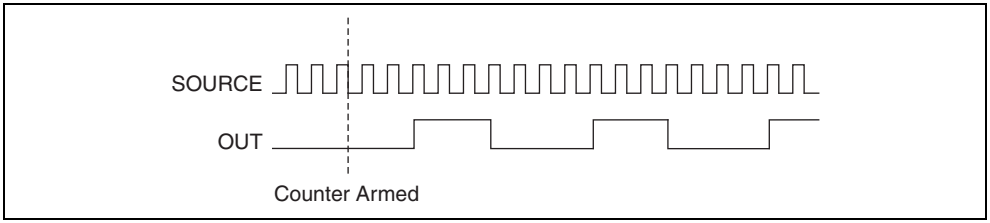


Figure 50. Continuous Pulse Train Generation

Continuous pulse train generation is sometimes called frequency division. If the high and low pulse widths of the output signal are *M* and *N* periods, then the frequency of the Counter *n* Internal Output signal is equal to the frequency of the Source input divided by *M* + *N*.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Buffered Pulse Train Generation

NI cDAQ-9178/9174 counters can use the FIFO to perform a buffered pulse train generation. This pulse train can use implicit timing or sample clock timing. When using implicit timing, the pulse idle time and active time changes with each sample you write. With sample clocked timing, each sample you write updates the idle time and active time of your generation on each sample clock edge. Idle time and active time can also be defined in terms of frequency and duty cycle or idle ticks and active ticks.

Note On buffered implicit pulse trains the pulse specifications in the DAQmx Create Counter Output Channel are ignored so that you generate the number of pulses defined in the multipoint write. On buffered sample clock pulse trains the pulse specifications in the DAQmx Create Counter Output Channel are generated after the counters starts and before the first sample clock so that you generate the number of updates defined in the multipoint write.

Finite Implicit Buffered Pulse Train Generation

This function generates a predetermined number of pulses with variable idle and active times. Each point you write generates a single pulse. The number of pairs of idle and active times (pulse specifications) you write determines the number of pulses generated. All points are generated back to back to create a user defined pulse train.

Table 9 and Figure 51 detail a finite implicit generation of three samples.

Table 9. Finite Implicit Buffered Pulse Train Generation

Sample	Idle Ticks	Active Ticks
1	2	2
2	3	4
3	2	2

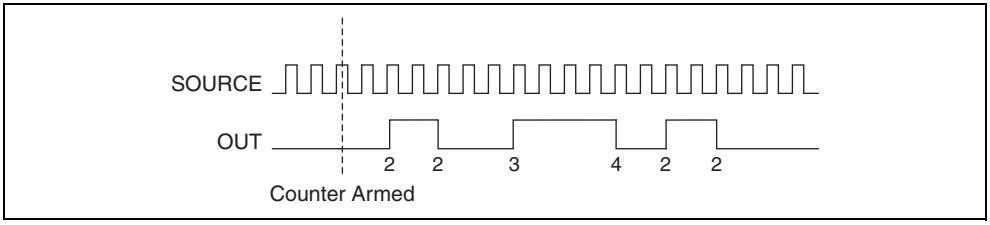


Figure 51. Finite Implicit Buffered Pulse Train Generation

Continuous Buffered Implicit Pulse Train Generation

This function generates a continuous train of pulses with variable idle and active times. Instead of generating a set number of data samples and stopping, a continuous generation continues until you stop the operation. Each point you write generates a single pulse. All points are generated back to back to create a user defined pulse train.

Finite Buffered Sample Clocked Pulse Train Generation

This function generates a predetermined number of pulse train updates. Each point you write defines pulse specifications that are updated with each sample clock. When a sample clock occurs, the current pulse (idle followed by active) finishes generation and the next pulse updates with the next sample specifications.

Note When the last sample is generated, the pulse train continues to generate with these specifications until the task is stopped.

Table 10 and Figure 52 detail a finite sample clocked generation of three samples where the pulse specifications from the create channel are two ticks idle, two ticks active, and three ticks initial delay.

Table 10. Finite Buffered Sample Clocked Pulse Train Generation

Sample	Idle Ticks	Active Ticks
1	3	3
2	2	2
3	3	3

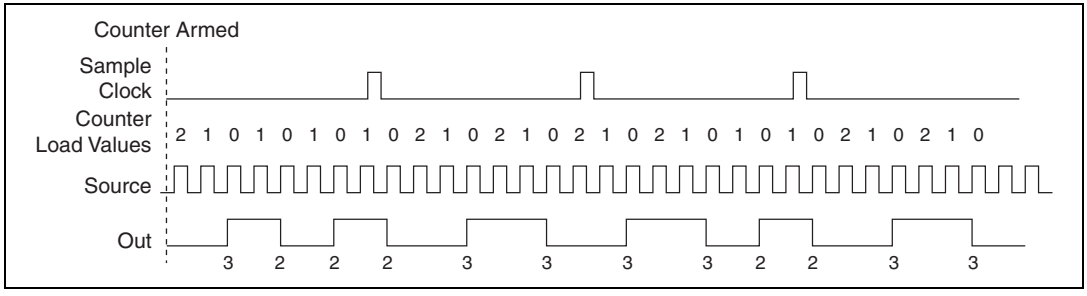


Figure 52. Finite Buffered Sample Clocked Pulse Train Generation

There are several different methods of continuous generation that control what data is written. These methods are regeneration, FIFO regeneration, and non-regeneration modes.

Regeneration is the repetition of the data that is already in the buffer.

Standard regeneration is when data from the PC buffer is continually downloaded to the FIFO to be written out. New data can be written to the PC buffer at any time without disrupting the output. With FIFO regeneration, the entire buffer is downloaded to the FIFO and regenerated from there. Once the data is downloaded, new data cannot be written to the FIFO. To use FIFO regeneration, the entire buffer must fit within the FIFO size. The advantage of using FIFO regeneration is that it does not require communication with the main host memory once the operation is started, thereby preventing any problems that may occur due to excessive bus traffic.

With non-regeneration, old data is not repeated. New data must be continually written to the buffer. If the program does not write new data to the buffer at a fast enough rate to keep up with the generation, the buffer underflows and causes an error.

Continuous Buffered Sample Clocked Pulse Train Generation

This function generates a continuous train of pulses with variable idle and active times. Instead of generating a set number of data samples and stopping, a continuous generation continues until you stop the operation. Each point you write specifies pulse specifications that are updated with each sample clock. When a sample clock occurs, the current pulse finishes generation and the next pulse uses the next sample specifications.

Frequency Generation

You can generate a frequency by using a counter in pulse train generation mode or by using the frequency generator circuit, as described in the *Using the Frequency Generator* section.

Using the Frequency Generator

The frequency generator can output a square wave at many different frequencies. The frequency generator is independent of the four general-purpose 32-bit counter/timer modules on the NI cDAQ-9178/9174.

Figure 53 shows a block diagram of the frequency generator.

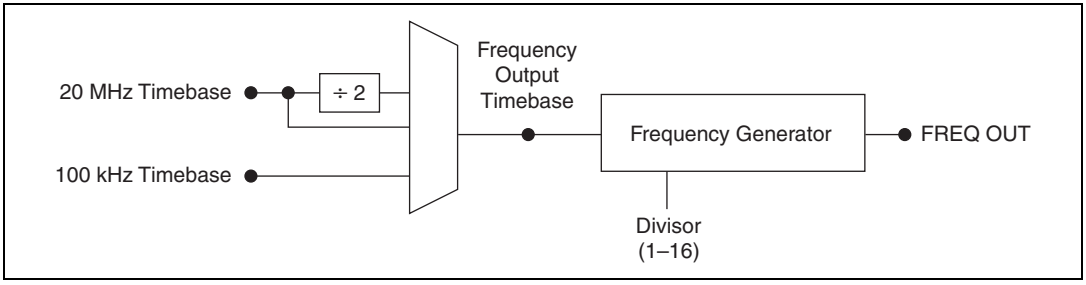


Figure 53. Frequency Generator Block Diagram

The frequency generator generates the Frequency Output signal. The Frequency Output signal is the Frequency Output Timebase divided by a number you select from 1 to 16. The Frequency Output Timebase can be either the 20 MHz Timebase, the 20 MHz Timebase divided by 2, or the 100 kHz Timebase.

The duty cycle of Frequency Output is 50% if the divider is either 1 or an even number. For an odd divider, suppose the divider is set to D. In this case, Frequency Output is low for (D + 1)/2 cycles and high for (D – 1)/2 cycles of the Frequency Output Timebase.

Figure 54 shows the output waveform of the frequency generator when the divider is set to 5.

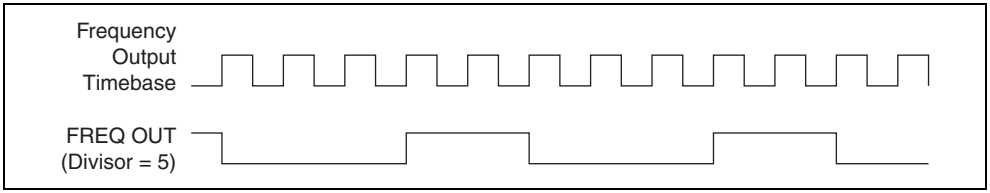


Figure 54. Frequency Generator Output Waveform

Frequency Output can be routed out to any PFI terminal. All PFI terminals are set to high-impedance at startup. The FREQ OUT signal also can be routed to many internal timing signals.

In software, program the frequency generator as you would program one of the counters for pulse train generation.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Frequency Division

The counters can generate a signal with a frequency that is a fraction of an input signal. This function is equivalent to continuous pulse train generation. Refer to the *Continuous Pulse Train Generation* section for detailed information.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Pulse Generation for ETS

In the equivalent time sampling (ETS) application, the counter produces a pulse on the output a specified delay after an active edge on Gate. After each active edge on Gate, the counter cumulatively increments the delay between the Gate and the pulse on the output by a specified amount. Thus, the delay between the Gate and the pulse produced successively increases.

The increase in the delay value can be between 0 and 255. For instance, if you specify the increment to be 10, the delay between the active Gate edge and the pulse on the output increases by 10 every time a new pulse is generated.

Suppose you program your counter to generate pulses with a delay of 100 and pulse width of 200 each time it receives a trigger. Furthermore, suppose you specify the delay increment to be 10. On the first trigger, your pulse delay will be 100, on the second it will be 110, on the third it will be 120; the process will repeat in this manner until the counter is disarmed. The counter ignores any Gate edge that is received while the pulse triggered by the previous Gate edge is in progress.

The waveform thus produced at the counter's output can be used to provide timing for undersampling applications where a digitizing system can sample repetitive waveforms that are higher in frequency than the Nyquist frequency of the system. Figure 55 shows an example of pulse generation for ETS; the delay from the trigger to the pulse increases after each subsequent Gate active edge.

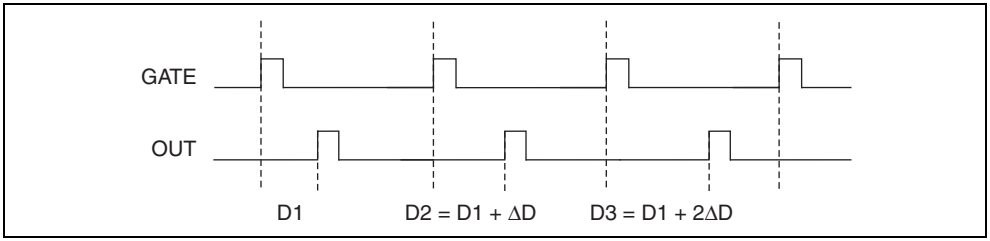


Figure 55. Pulse Generation for ETS

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Counter Timing Signals

The NI cDAQ-9178/9174 feature the following counter timing signals:

- *Counter n Source Signal*
- *Counter n Gate Signal*
- *Counter n Aux Signal*
- *Counter n A Signal*
- *Counter n B Signal*
- *Counter n Z Signal*
- *Counter n Up_Down Signal*
- *Counter n HW Arm Signal*
- *Counter n Sample Clock Signal*
- *Counter n Internal Output Signal*
- *Counter n TC Signal*
- *Frequency Output Signal*

Note All counter timing signals can be filtered.

In this section, *n* refers to the NI cDAQ-9178/9174 Counter 0, 1, 2, or 3. For example, Counter *n* Source refers to four signals—Counter 0 Source (the source input to Counter 0), Counter 1 Source (the source input to Counter 1), Counter 2 Source (the source input to Counter 2), or Counter 3 Source (the source input to Counter 3).

Each of these signals supports digital filtering.

Counter *n* Source Signal

The selected edge of the Counter *n* Source signal increments and decrements the counter value depending on the application the counter is performing. Table 11 lists how this terminal is used in various applications.

Table 11. Counter Applications and Counter *n* Source

Application	Purpose of Source Terminal
Pulse Generation	Counter Timebase
One Counter Time Measurements	Counter Timebase
Two Counter Time Measurements	Input Terminal
Non-Buffered Edge Counting	Input Terminal
Buffered Edge Counting	Input Terminal
Two-Edge Separation	Counter Timebase

Routing a Signal to Counter *n* Source

Each counter has independent input selectors for the Counter *n* Source signal. Any of the following signals can be routed to the Counter *n* Source input:

- 80 MHz Timebase
- 20 MHz Timebase
- 100 kHz Timebase
- Any PFI terminal
- Analog Comparison Event
- Change Detection Event

In addition, TC or Gate from a counter can be routed to a different counter source.

Some of these options may not be available in some driver software.

Routing Counter *n* Source to an Output Terminal

You can route Counter *n* Source out to any PFI terminal.

Counter *n* Gate Signal

The Counter *n* Gate signal can perform many different operations depending on the application including starting and stopping the counter, and saving the counter contents.

Routing a Signal to Counter *n* Gate

Each counter has independent input selectors for the Counter *n* Gate signal. Any of the following signals can be routed to the Counter *n* Gate input:

- Any PFI terminal
- AI Reference Trigger (ai/ReferenceTrigger)
- AI Start Trigger (ai/StartTrigger)
- AO Sample Clock (ao/SampleClock)
- DI Sample Clock (di/SampleClock)
- DI Reference Trigger (di/ReferenceTrigger)

- DO Sample Clock (do/SampleClock)
- Change Detection Event
- Analog Comparison Event

In addition, a counter's Internal Output or Source can be routed to a different counter's gate.

Some of these options may not be available in some driver software.

Routing Counter *n* Gate to an Output Terminal

You can route Counter *n* Gate out to any PFI terminal.

Counter *n* Aux Signal

The Counter *n* Aux signal indicates the first edge in a two-signal edge-separation measurement.

Routing a Signal to Counter *n* Aux

Each counter has independent input selectors for the Counter *n* Aux signal. Any of the following signals can be routed to the Counter *n* Aux input:

- Any PFI terminal
- AI Reference Trigger (ai/ReferenceTrigger)
- AI Start Trigger (ai/StartTrigger)
- Analog Comparison Event
- Change Detection Event

In addition, a counter's Internal Output, Gate or Source can be routed to a different counter's Aux. A counter's own gate can also be routed to its Aux input.

Some of these options may not be available in some driver software.

Counter *n* A, Counter *n* B, and Counter *n* Z Signals

Counter *n* B can control the direction of counting in edge counting applications. Use the A, B, and Z inputs to each counter when measuring quadrature encoders or measuring two pulse encoders.

Routing Signals to A, B, and Z Counter Inputs

Each counter has independent input selectors for each of the A, B, and Z inputs. Any of the following signals can be routed to each input:

- Any PFI terminal
- Analog Comparison Event

Routing Counter *n* Z Signal to an Output Terminal

You can route Counter *n* Z out to any PFI terminal.

Counter *n* Up_Down Signal

Counter *n* Up_Down is another name for the Counter *n* B signal.

Counter *n* HW Arm Signal

The Counter *n* HW Arm signal enables a counter to begin an input or output function.

To begin any counter input or output function, you must first enable, or arm, the counter. In some applications, such as a buffered edge count, the counter begins counting when it is armed. In other applications, such as single pulse-width measurement, the counter begins waiting for the Gate signal when it is armed. Counter output operations can use the arm signal in addition to a start trigger.

Software can arm a counter or configure counters to be armed on a hardware signal. Software calls this hardware signal the Arm Start Trigger. Internally, software routes the Arm Start Trigger to the Counter *n* HW Arm input of the counter.

Routing Signals to Counter *n* HW Arm Input

Any of the following signals can be routed to the Counter *n* HW Arm input:

- Any PFI terminal
- AI Reference Trigger (ai/ReferenceTrigger)
- AI Start Trigger (ai/StartTrigger)
- Analog Comparison Event
- Change Detection Event

A counter's Internal Output can be routed to a different counter's HW Arm.

Some of these options may not be available in some driver software.

Counter *n* Sample Clock Signal

Use the Counter *n* Sample Clock (Ctr*n*SampleClock) signal to perform sample clocked acquisitions and generations.

You can specify an internal or external source for Counter *n* Sample Clock. You also can specify whether the measurement sample begins on the rising edge or falling edge of Counter *n* Sample Clock.

If the NI cDAQ-9178/9174 receives a Counter *n* Sample Clock when the FIFO is full, it reports an overflow error to the host software.

Using an Internal Source

To use Counter *n* Sample Clock with an internal source, specify the signal source and the polarity of the signal. The source can be any of the following signals:

- DI Sample Clock (di/SampleClock)
- DO Sample Clock (do/SampleClock)
- AI Sample Clock (ai/SampleClock, te0/SampleClock, te1/SampleClock)
- AI Convert Clock (ai/ConvertClock)
- AO Sample Clock (ao/SampleClock)
- DI Change Detection output

Several other internal signals can be routed to Counter *n* Sample Clock through internal routes. Refer to *Device Routing in MAX* in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an External Source

You can route any of the following signals as Counter *n* Sample Clock:

- Any PFI terminal
- Analog Comparison Event

You can sample data on the rising or falling edge of Counter *n* Sample Clock.

Routing Counter *n* Sample Clock to an Output Terminal

You can route Counter *n* Sample Clock out to any PFI terminal. The PFI circuitry inverts the polarity of Counter n Sample Clock before driving the PFI terminal.

Counter *n* Internal Output and Counter *n* TC Signals

The Counter *n* Internal Output signal changes in response to Counter *n* TC.

The two software-selectable output options are pulse output on TC and toggle output on TC. The output polarity is software-selectable for both options.

With pulse or pulse train generation tasks, the counter drives the pulse(s) on the Counter *n* Internal Output signal. The Counter *n* Internal Output signal can be internally routed to be a counter/timer input or an "external" source for AI, AO, DI, or DO timing signals.

Routing Counter *n* Internal Output to an Output Terminal

You can route Counter *n* Internal Output to any PFI terminal.

Frequency Output Signal

The Frequency Output (FREQ OUT) signal is the output of the frequency output generator.

Routing Frequency Output to a Terminal

You can route Frequency Output to any PFI terminal.

Default Counter/Timer Routing

Counter/timer signals are available to correlated digital I/O C Series modules. To determine the signal routing options for modules installed in your system, refer to the **Device Routes** tab in MAX.

You can use these defaults or select other sources and destinations for the counter/timer signals in NI-DAQmx. Refer to *Connecting Counter Signals* in the *NI-DAQmx Help* or the *LabVIEW Help* for more information about how to connect your signals for common counter measurements and generations. NI cDAQ-9178/9174 default PFI lines for counter functions are listed in *Physical Channels* in the *NI-DAQmx Help* or the *LabVIEW Help*.

Counter Triggering

Counters support three different triggering actions:

- **Arm Start Trigger**—To begin any counter input or output function, you must first enable, or arm, the counter. Software can arm a counter or configure counters to be armed on a hardware signal. Software calls this hardware signal the Arm Start Trigger. Internally, software routes the Arm Start Trigger to the Counter *n* HW Arm input of the counter.

  For counter output operations, you can use it in addition to the start and pause triggers. For counter input operations, you can use the arm start trigger to have start trigger-like behavior. The arm start trigger can be used for synchronizing multiple counter input and output tasks.

  When using an arm start trigger, the arm start trigger source is routed to the Counter *n* HW Arm signal.
- **Start Trigger**—For counter output operations, a start trigger can be configured to begin a finite or continuous pulse generation. Once a continuous generation has triggered, the pulses continue to generate until you stop the operation in software. For finite generations, the specified number of pulses is generated and the generation stops unless you use the retriggerable attribute. When you use this attribute, subsequent start triggers cause the generation to restart.

  When using a start trigger, the start trigger source is routed to the Counter *n* Gate signal input of the counter. Counter input operations can use the arm start trigger to have start trigger-like behavior.
- **Pause Trigger**—You can use pause triggers in edge counting and continuous pulse generation applications. For edge counting acquisitions, the counter stops counting edges while the external trigger signal is low and resumes when the signal goes high or vice versa. For continuous pulse

generations, the counter stops generating pulses while the external trigger signal is low and resumes when the signal goes high or vice versa.

When using a pause trigger, the pause trigger source is routed to the Counter *n* Gate signal input of the counter.

Other Counter Features

The following sections list the other counter features available on the NI cDAQ-9178/9174.

Cascading Counters

You can internally route the Counter n Internal Output and Counter n TC signals of each counter to the Gate inputs of the other counter. By cascading two counters together, you can effectively create a 64-bit counter. By cascading counters, you also can enable other applications. For example, to improve the accuracy of frequency measurements, use reciprocal frequency measurement, as described in the *Large Range of Frequencies with Two Counters* section.

Prescaling

Prescaling allows the counter to count a signal that is faster than the maximum timebase of the counter. The NI cDAQ-9178/9174 offers 8X and 2X prescaling on each counter (prescaling can be disabled). Each prescaler consists of a small, simple counter that counts to eight (or two) and rolls over. This counter can run faster than the larger counters, which simply count the rollovers of this smaller counter. Thus, the prescaler acts as a frequency divider on the Source and puts out a frequency that is one-eighth (or one-half) of what it is accepting as shown in Figure 56.

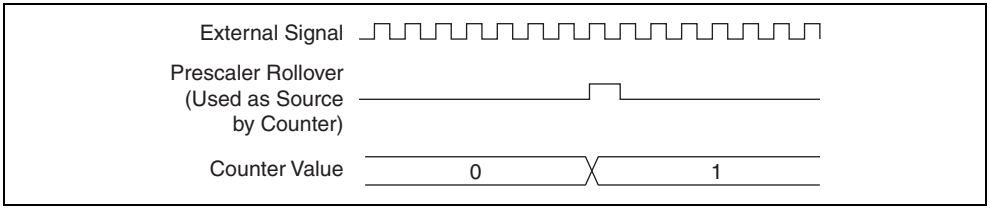


Figure 56. Prescaling

Prescaling is intended to be used for frequency measurement where the measurement is made on a continuous, repetitive signal. The prescaling counter cannot be read; therefore, you cannot determine how many edges have occurred since the previous rollover. Prescaling can be used for event counting provided it is acceptable to have an error of up to seven (or one) ticks. Prescaling can be used when the counter Source is an external signal. Prescaling is not available if the counter Source is one of the internal timebases (80MHzTimebase, 20MHzTimebase, or 100kHzTimebase).

Synchronization Modes

The 32-bit counter counts up or down synchronously with the Source signal. The Gate signal and other counter inputs are asynchronous to the Source signal, so the NI cDAQ-9178/9174 synchronizes these signals before presenting them to the internal counter.

Depending on how you configure your device, the NI cDAQ-9178/9174 uses one of two synchronization methods:

- *80 MHz Source Mode*
- *External or Internal Source Less than 20 MHz*

80 MHz Source Mode

In 80 MHz source mode, the device synchronizes signals on the rising edge of the source, and counts on the third rising edge of the source. Edges are pipelined so no counts are lost, as shown in Figure 57.

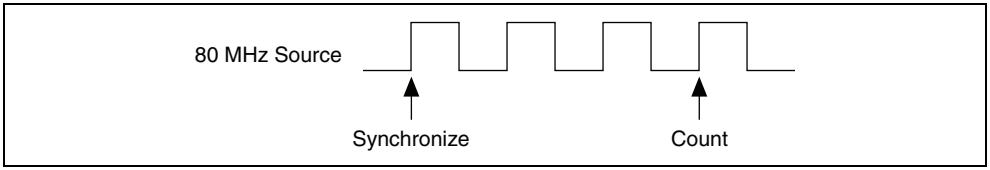


Figure 57. 80 MHz Source Mode

External or Internal Source Less than 20 MHz

With an external or internal source less than 20 MHz, the device generates a delayed Source signal by delaying the Source signal by several nanoseconds. The device synchronizes signals on the rising edge of the delayed Source signal, and counts on the following rising edge of the source, as shown in Figure 58.

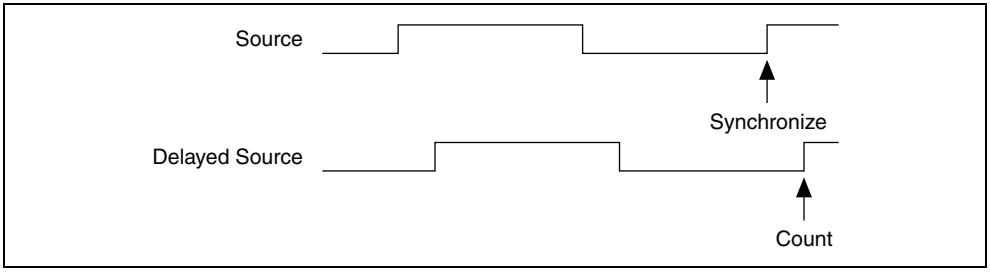


Figure 58. External or Internal Source Less than 20 MHz

Digital Routing and Clock Generation

The digital routing circuitry has the following functions:

- Manages the flow of data between the bus interface and the acquisition/generation sub-systems (analog input, analog output, digital I/O, and the counters). The digital routing circuitry uses FIFOs (if present) in each sub-system to ensure efficient data movement.
- Routes timing and control signals. The acquisition/generation sub-systems use these signals to manage acquisitions and generations. These signals can come from the following sources:
  - Your C Series I/O modules
  - User input through the PFI terminals using hardware-timed digital C Series I/O modules or the NI cDAQ-9178 chassis PFI terminals.
- Routes and generates the main clock signals for the NI cDAQ-9178/9174 chassis. To determine the signal routing options for C Series I/O modules installed in the NI cDAQ-9178/9174 chassis, refer to the **Device Routes** tab in MAX.

Clock Routing

Figure 59 shows the clock routing circuitry of the NI cDAQ-9178/9174 chassis.

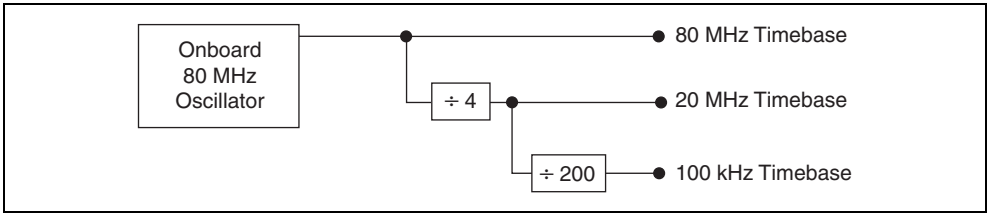


Figure 59. NI cDAQ-9178/9174 Clock Routing Circuitry

80 MHz Timebase

You can use the 80 MHz Timebase as the Source input to the 32-bit general-purpose counter/timers.

The 80 MHz Timebase can be generated from the onboard oscillator.

20 MHz Timebase

The 20 MHz Timebase normally generates many of the AI and AO timing signals. It can function as the Source input to the 32-bit general-purpose counter/timers.

The 20 MHz Timebase is generated by dividing down the 80 MHz Timebase, as shown in Figure 59.

100 kHz Timebase

You can use the 100 kHz Timebase to generate many of the AI and AO timing signals. It can also function as the Source input to the 32-bit general-purpose counter/timers.

The 100 kHz Timebase is generated by dividing down the 20 MHz Timebase by 200, as shown in Figure 59.

Specifications

These specifications are for the NI cDAQ-9178/9174 chassis only. These specifications are typical at 25 °C unless otherwise noted. For the C Series I/O module specifications, refer to the documentation for the C Series I/O modules you are using.

Analog Input

Input FIFO size ..... 127 samples per slot

Sample rate[1]

- Maximum ..... 6.4 MS/s, system dependent (multi-channel, aggregate)
- Minimum ..... 0 S/s

Timing accuracy[2] ..... 50 ppm of sample rate

Timing resolution[2] ..... 12.5 ns

Number of channels supported ..... Determined by the C Series I/O modules

Analog Output

Numbers of channels supported

- In hardware-timed task using onboard regeneration 16
- In hardware-timed task not using onboard regeneration ..... Determined by the C Series I/O modules
- In non-hardware-timed task ..... Determined by the C Series I/O modules

Maximum update rate

- Regeneration ..... 1.6 MS/s (multi-channel, aggregate)
- Non-regeneration ..... Determined by the C Series I/O modules

Timing accuracy ..... 50 ppm of sample rate

Timing resolution ..... 12.5 ns

Output FIFO size

- Onboard regeneration ..... 8,191 samples shared among channels used
- Non-regeneration ..... 127 samples per slot

AO waveform modes ..... Non-periodic waveform, periodic waveform regeneration mode from onboard memory, periodic waveform regeneration from host buffer including dynamic update

Digital Waveform Characteristics

Waveform acquisition (DI) FIFO ..... 127 samples per slot

Waveform generation (DO) FIFO

- NI cDAQ-9178/9174
  - Slots 1–4 ..... 2,047 samples
- NI cDAQ-9178 only
  - Slots 5–8 ..... 1,023 samples

[1] Performance dependent on type of installed C Series I/O modules and number of channels in the task.

[2] Does not include group delay. Refer to C Series I/O module documentation for more information.

Digital input sample clock frequency

Streaming to application memory ..... System-dependent

Finite ..... 0 to 10 MHz

Digital output sample clock frequency

Streaming from application memory ..... System-dependent

Regeneration from FIFO ..... 0 to 10 MHz

Finite ..... 0 to 10 MHz

Digital output or digital input sample clock source ..... Any PFI, analog sample or convert clock, analog output sample clock, Ctr *n* Internal Output, and many other sources

General-Purpose Counter/Timers

Number of counter/timers 4

Resolution ..... 32 bits

Counter measurements ..... Edge counting, pulse, semi-period, period, two-edge separation, pulse width

Position measurements ..... X1, X2, X4 quadrature encoding with Channel Z reloading; two-pulse encoding

Output applications ..... Pulse, pulse train with dynamic updates, frequency division, equivalent time sampling

Internal base clocks ..... 80 MHz, 20 MHz, 100 kHz

External base clock frequency ..... 0 to 20 MHz

Base clock accuracy ..... 50 ppm

Output frequency ..... 0 to 20 MHz

Inputs ..... Gate, Source, HW_Arm, Aux, A, B, Z, Up_Down

Routing options for inputs ..... Any PFI, analog trigger, many internal signals

FIFO ..... Dedicated 127-sample FIFO

Frequency Generator

Number of channels 1

Base clocks ..... 10 MHz, 20 MHz, 100 kHz

Divisors ..... 1 to 16 (integers)

Base clock accuracy ..... 50 ppm

Output is available on any PFI terminal.

Module PFI Characteristics

Functionality ..... Static digital input, static digital output, timing input, and timing output

Timing output sources ..... Many analog input, analog output, counter, digital input, and digital output timing signals

Timing input frequency ..... 0 to 20 MHz

Timing output frequency ..... 0 to 20 MHz

Chassis PFI Characteristics (NI cDAQ-9178 Only)

Max input or output frequency ..... 1 MHz

	Minimum	Maximum
Positive Going Threshold Voltage	1.43	2.28
Negative Going Threshold Voltage	0.86	1.53
Hysteresis	0.48	0.87

Maximum voltage ..... 25 V

Minimum voltage ..... –20 V

Cable length ..... 3 m (10 ft.)

Cable impedance ..... 50 Ω

Connector ..... BNC

Output voltage

- High ..... 5.25 V max
  - Sourcing 100 µA ..... 4.65 V min
  - Sourcing 2 mA ..... 3.60 V min
- Low
  - Sinking 100 µA ..... 0.10 V max
  - Sinking 2 mA ..... 0.64 V max

Power-on state ..... High impedance

External Digital Triggers

Source ..... Any PFI terminal or chassis PFI terminal (NI cDAQ-9178 only)

Polarity ..... Software-selectable for most signals

Analog input function ..... Start Trigger, Reference Trigger, Pause Trigger, Sample Clock, Sample Clock Timebase

Analog output function ..... Start Trigger, Pause Trigger, Sample Clock, Sample Clock Timebase

Counter/timer functions ..... Gate, Source, HW_Arm, Aux, A, B, Z, Up_Down

Module I/O States

At power-on .....Module-dependent. Refer to the documentation included with the C Series I/O module(s).

Note The chassis may revert the input/output of the modules to its power-on state when the USB cable is removed.

Power Requirements

You must use a National Electric Code (NEC) Class 2 power source with the NI cDAQ-9178/9174 chassis.

Note Some C Series I/O modules have additional power requirements. For more information about C Series I/O module(s) power requirements, refer to documentation included with the C Series I/O module(s).

Note Sleep mode for C Series I/O modules is not supported in the NI cDAQ-9178/9174.

Input voltage range .....9 V to 30 V

Maximum required input power .....15 W

Power input connector .....2 positions 3.5mm pitch pluggable screw terminal with screw locks similar to Sauro CTM020F8

Power input mating connector .....Sauro CTF020VT or equivalent

Bus Interface

USB specification .....USB 2.0 Hi-Speed

Power from USB

4.10 to 5.25 V .....500 µA maximum

High-performance data streams 7

Types available.....Analog input, analog output, digital input, digital output, counter/timer input, counter/timer output

Physical Characteristics

NI cDAQ-9178 chassis

Weight (unloaded).....Approx. 878 g (31.0 oz)

Dimensions (unloaded) .....25.4 cm × 8.81 cm × 5.89 cm (10.0 in. × 3.5 in. × 2.3 in.)

NI cDAQ-9174 chassis

Weight (unloaded).....Approx. 574 g (20.2 oz)

Dimensions (unloaded) .....15.9 cm × 8.81 cm × 5.89 cm (6.28 in. × 3.5 in. × 2.3 in.)

Safety

If you need to clean the chassis, wipe it with a dry towel.

This product is designed to meet the requirements of the following standards of safety for electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN-61010-1
- UL 61010-1, CSA 61010-1

Note For UL and other safety certifications, refer to the product label or visit `ni.com/certification`, search by model number or product line, and click the appropriate link in the Certification column.

Environmental

The NI cDAQ-9178/9174 chassis is intended for indoor use only. For outdoor use, mount the system in a suitably rated enclosure.

Operating temperature[1]
(IEC-60068-2-1 and IEC-60068-2-2) .....–20 to 55 °C

Storage temperature
(IEC-60068-2-1 and IEC-60068-2-2) .....–40 to 85 °C

Ingress protection.....IP 30

Operating humidity (IEC-60068-2-56).....10 to 90% RH, noncondensing

Storage humidity (IEC-60068-2-56).....5 to 95% RH, noncondensing

Maximum altitude.....2,000 m

Pollution Degree (IEC 60664) 2

Shock and Vibration

To meet these specifications, you must panel mount the NI cDAQ-9178/9174 system and affix ferrules to the ends of the terminal lines.

Operational shock .....30 g peak, half-sine, 11 ms pulse
(Tested in accordance with IEC-60068-2-27. Test profile developed in accordance with MIL-PRF-28800F.)

Random vibration

Operating .....5 to 500 Hz, 0.3 g_{rms}

Non-operating .....5 to 500 Hz, 2.4 g_{rms}
(Tested in accordance with IEC-60068-2-64. Non-operating test profile exceeds the requirements of MIL-PRF-28800F, Class 3.)

[1] When operated in temperatures below 0 °C, you must use the PS-5 power supply, or another power supply rated for below 0 °C.

Safety Standards

This product meets the requirements of the following standards of safety for electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN 61010-1
- UL 61010-1, CSA 61010-1

Note For UL and other safety certifications, refer to the product label or the *Online Product Certification* section.

Electromagnetic Compatibility

This product meets the requirements of the following EMC standards for electrical equipment for measurement, control, and laboratory use:

- EN 61326 (IEC 61326): Class A emissions; Basic immunity
- EN 55011 (CISPR 11): Group 1, Class A emissions
- AS/NZS CISPR 11: Group 1, Class A emissions
- FCC 47 CFR Part 15B: Class A emissions
- ICES-001: Class A emissions

Note For the standards applied to assess the EMC performance of this product, refer to the *Online Product Certification* section.

Note For EMC compliance, operate this device with shielded cables and the included power supply.

CE Compliance CE

This product meets the essential requirements of applicable European Directives as follows:

- 2006/95/EC; Low-Voltage Directive (safety)
- 2004/108/EC; Electromagnetic Compatibility Directive (EMC)

Online Product Certification

Note Refer to the product Declaration of Conformity (DoC) for additional regulatory compliance information. To obtain product certifications and the DoC for this product, visit `ni.com/certification`, search by model number or product line, and click the appropriate link in the Certification column.

Environmental Management

National Instruments is committed to designing and manufacturing products in an environmentally responsible manner. NI recognizes that eliminating certain hazardous substances from our products is beneficial not only to the environment but also to NI customers.

For additional environmental information, refer to the *NI and the Environment* Web page at `ni.com/environment`. This page contains the environmental regulations and directives with which NI complies, as well as other environmental information not included in this document.

Waste Electrical and Electronic Equipment (WEEE)

EU Customers At the end of their life cycle, all products *must* be sent to a WEEE recycling center. For more information about WEEE recycling centers and National Instruments WEEE initiatives, visit ni.com/environment/weee.htm.

电子信息产品污染控制管理办法 （中国 RoHS）

中国客户 National Instruments 符合中国电子信息产品中限制使用某些有害物质指令 (RoHS)。关于 National Instruments 中国 RoHS 合规性信息，请登录 ni.com/environment/rohs_china。(For information about China RoHS compliance, go to ni.com/environment/rohs_china.)

Where to Go for Support

The National Instruments Web site is your complete resource for technical support. At ni.com/support you have access to everything from troubleshooting and application development self-help resources to email and phone assistance from NI Application Engineers.

National Instruments corporate headquarters is located at 11500 North Mopac Expressway, Austin, Texas, 78759-3504. National Instruments also has offices located around the world to help address your support needs. For telephone support in the United States, create your service request at ni.com/support and follow the calling instructions or dial 512 795 8248. For telephone support outside the United States, contact your local branch office:

Australia 1800 300 800, Austria 43 662 457990-0, Belgium 32 (0) 2 757 0020,
Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 5050 9800,
Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 358 (0) 9 725 72511,
France 01 57 66 24 24, Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737,
Italy 39 02 41309277, Japan 0120-527196, Korea 82 02 3451 3400, Lebanon 961 (0) 1 33 28 28,
Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466,
New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60, Poland 48 22 328 90 10,
Portugal 351 210 311 210, Russia 7 495 783 6851, Singapore 1800 226 5886,
Slovenia 386 3 425 42 00, South Africa 27 0 11 805 8197, Spain 34 91 640 0085,
Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151, Taiwan 886 02 2377 2222,
Thailand 662 278 6777, Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

372838A-01 Oct09