NI-Sync

2025-03-20

Contents

Synchronization Methods 3
Signal-Based vs. Time-Based Synchronization 3
Signal-Based Synchronization in NI-Sync 3
Time-Based Synchronization in NI-Sync. 5
Clocks and Triggers. 9
Triggers in NI-Sync 9
Defining NI-Sync Trigger Terminals 9
Trigger Terminal Connections 11
Clocks in NI-Sync 14
Clock Terminal Connections 14
Defining NI-Sync Clock Terminals 15
Time References and Protocols 19
GPS 19
IEEE 1588-2008 20
IRIG-B 23
PPS 24

Signal-Based vs. Time-Based Synchronization

Synchronizing multiple devices requires accurate timing for event synchronization and data correlation. NI-Sync achieves timing accuracy between devices using two synchronization methods: signal-based and time-based.

In a signal-based application, clocks and trigger signals are shared directly between nodes that require synchronization. In a time-based application, nodes independently synchronize their individual clocks based on a common time source, usually an external timing reference like GPS.

Note A combination of signal-based and time-based programming is often the best choice for complex system configurations with many different types of nodes in different locations.

Refer to the following table to weigh the pros and cons of using signal-based or time-based synchronization in an application:

Signal-Based	Time-Based
• Nodes are less than 100 meters apart. • System has no access to an external time reference. • Nodes are physically connected using cables or switches • System is in a closed environment without network access. • System requires higher precision synchronization. • System requires synchronization of a small number of nodes.	• Nodes are more than 100 meters apart. • System has access to an external time reference, such as an IRIG generator or a GPS antenna. • Nodes are connected through a network (including through Ethernet). • System requires synchronization of a large number of nodes. • Nodes frequently change location.

Signal-Based Synchronization in NI-Sync

Signal-based synchronization is characterized by physically sharing a combination of clocks and triggers to synchronize data acquisition and instrument control. Using a signal-based synchronization method, you can route clock and trigger signals throughout the system with Connect Clock Terminals and Connect Trigger Terminals. Once you connect clock and trigger terminals, the signals are used to synchronize devices and modules using the terminal routes you established.

Using the signal-based synchronization method, you can perform the following actions:

- Generate a clock that runs at a user-defined frequency.
- Use triggers to synchronize data acquisition.
- Divide a clock to create new frequencies that are still synchronized with the main clock.
- Route clock and trigger terminals between devices using Connect Clock Terminals and Connect Trigger Terminals.
- Lock the backplane clock to an external device's clock using the PLL circuit.
- Discipline the backplane clock to a module's oscillator.
- Measure the frequency on a specific terminal using Measure Frequency.

The clock signal(s) used to synchronize devices in the signal-based synchronization method can originate from an external clock connected to ClkIn, a DDS signal created with a signal-based timing and synchronization module, the backplane clock of a PXI or PXIe chassis, or the oscillator of an NI-Sync device.

Signal-Based Synchronization Phases

The signal-based synchronization method can be broken down into the following phases:

- Initialize: Create an NI-Sync session to establish communication with an NI signal-based timing and synchronization device. Initialize creates a unique session handle that identifies the device to subsequent NI-Sync nodes. You can also use Initialize to reset the device to a known state and verify that the NI-Sync instrument driver is valid for the device.
- Configure Hardware: Set up your device for synchronization, typically using niSync Properties. You can choose the type of signal to generate or synchronize with, the frequency of the signal (if you are using DDS), and the synchronization clock the

front and rear zones of your chassis should use.
- Connect Terminals: Connect clock and trigger terminals between modules and chassis. Use this step to route triggers to different parts of the chassis and synchronize modules to the same clock signal. During the Connect Terminals phase, you can connect source and destination terminals, set a synchronization clock for triggers, and discipline the backplane clock to a clock signal. You can use Connect Clock Terminals, Connect Trigger Terminals, and Connect Software Trigger in the Connect Terminals phase.
- Configure and Perform Measurement: Begin data acquisition using the parameters you set in the Configure Hardware phase and the terminal routes you created in the Connect Terminals phases. Taking a measurement is an application-specific operation that usually involves the use of a separate API, such as NI-DAQmx. You can also return synchronization information using niSync Properties or send a software trigger using Send Software Trigger in this phase.
- Disconnect Terminals: Disconnect the terminal routes you set up in the Connect Terminals phase. Once you disconnect terminals, they are free for use in other applications. You can use Disconnect Clock Terminals, Disconnect Trigger Terminals, and Disconnect Software Trigger in this phase.
- Close: Close the NI-Sync session and end communication between the driver and the device using Close. This phase is necessary for deallocating memory and freeing other operating system resources. You must close every NI-Sync session you initialize, even if an error occurs when you execute the program.

Signal-Based Advanced and Utility Functions

Functions that do not fall into the above signal-based synchronization phases are considered Advanced or Utility functions.

Time-Based Synchronization in NI-Sync

Time-based synchronization, also known as distributed clock synchronization, is characterized by the use of an external timing source such as GPS, 1588, or an external IRIG generator. The system timing module uses the external time reference to determine the present time and create a clock that is locked to the external source. The individual clocks of each module and device in the system are synchronized to the same external source, ensuring synchronization between nodes no matter how far

apart they are. Devices act on timing signals originating from a local clock that is synchronized to the other clocks in the system, so instead of sharing timing signals directly, the devices periodically adjust their local timing sources to match the chosen external time reference.

Using the time-based synchronization method, you can perform the following actions:

- Create future time events that execute at a specific board time to control clock and trigger signals.
- Write and read timestamps to measure clock skew, record the start time of data acquisition, and troubleshoot timing issues.
- Create timed loops that run at a specific time of the day.
- Discipline the backplane clock to an external time reference.
- Return the current data and time, or the date and time when a measurement was taken.
- Generate a sample clock that starts and stops at a specific board time.

Synchronizing distributed clocks requires constant adjustment. A clock is essentially a two-part device that consists of a frequency source and an accumulator. In theory, if you set two clocks identically and their frequency sources run at the exact same rate, they are synchronized indefinitely. In practice, however, clocks are set with limited precision, frequency sources run at slightly different rates, and the rate of a frequency source changes over time and temperature. Most time-based NI timing and synchronization devices use an over-controlled crystal oscillator (OCXO) or a temperature-controlled crystal oscillator (TCXO) as a frequency source, but even these highly accurate frequency sources vary due to initial manufacturing tolerance, temperature and pressure changes, and aging.

Because of these instabilities, distributed clocks must be synchronized continually to match each other in frequency and phase. While a time-based system is generally not as accurate as a signal-based system, you can use the time-based synchronization method to synchronize complex systems with many different nodes spread out over a large area--even nodes that are moving--with no loss of accuracy. If you are using the GPS timing protocol, you can even measure the location, speed, and altitude of a node using time-based synchronization.

You can configure an NI-Sync device to use four different external time references: IEEE 1588-2008, GPS, IRIG-B, and PPS.

Time-Based Synchronization Phases

The time-based synchronization method can be broken down into the following phases:

- Initialize: Create an NI-Sync session to establish communication with a time-based module. Initialize creates a unique session handle that identifies the device to subsequent NI-Sync nodes. You can also use niSync Initialize to reset the device to a known state and verify that the NI-Sync instrument driver is valid for the device.
- Configure Hardware: Set up your device for synchronization, typically by selecting an external time reference using Set Time Reference. If you are using IEE 1588-2008 as an external time reference, you can also start participation in 1588 in this phase. You can use niSync Properties, Set Time Reference, and Start 1588 in the Configure Hardware phase.
- Get Time: Use Get Time to return the current board time of your device, which is synchronized with the external time reference you set in the Configure Hardware phase. You can pass this board time on to subsequent nodes to schedule future time events, start clocks at a specific time, or create timed loops in the later phases.

In the following phases, you configure future time events, enable timestamps, and generate clocks. These phases are independent of each other; you can use all of them, none of them, or a combination of the phases with time-based synchronization.

- Create Future Time Event: Use Create Future Time Event to schedule a future time event at a later in point in the application. A future time event changes the signal at a specific terminal when the board time reaches a specified point. You can create multiple future time events that change the signal levels on different terminals, change the signal multiple times at the same terminal to create waveforms, or use the future time event as a trigger to start data acquisition. The terminal you create a future time event on cannot be used for other purposes until you clear the future time event.
- Enable Timestamp Trigger: Use Enable Time Stamp Trigger to record a timestamp every time the signal at a specified terminal changes. This is useful for tracking future time events, recording the timing of data acquisition, and comparing clocks between devices to check for jitter and offset.
- Create Clock: Use Create Clock to create a custom clock that is synchronized with the external timing reference. You can set the clock to run for a specific number of

ticks per cycle, a set length of time, or at a certain frequency. You can use the board time you returned in the Get Time phase to determine when the created clock should start and stop.

In the following phases, you read timestamps you created and close the connections you established earlier in the application to free the resources for other uses.

- Read Trigger Timestamp: Use the Read Trigger Time Stamp function to read the timestamps you recorded in the Enable Timestamp Trigger phase. You can read a single timestamp or multiple timestamps in this phase. You can only read timestamps if you enabled the timestamp trigger with Enable Time Stamp Trigger.
- Clear Future Time Events/Disable Timestamp Trigger/Clear Clock: Disconnect the future time events, timestamp triggers, and clocks you created earlier in the program. Every time you create a future time event, enable a timestamp trigger, or create a clock, you must free the terminal at the end of the program using the appropriate function. Use Clear Future Time Events to remove future time events, Disable Time Stamp Trigger to stop recording timestamps, and Clear Clock to remove a generated clock from a terminal. If you are using IEEE 1588-2008 as the external time reference, you can also stop 1588 participation in this phase.
- Close: Close the NI-Sync session and end communication between the driver and your device using Close. This phase is necessary for deallocating memory and freeing other operating system resources. You must close every NI-Sync session you initialize, even if an error occurs when you execute the program.

Defining NI-Sync Trigger Terminals

Descriptions of trigger terminals you can connect with NI-Sync functions.

Using niSync Connect Trigger Terminals, you can route triggers between modules, synchronize triggers to different synchronization clocks, and export clock signals along some trigger lines. The function uses trigger terminals to pass trigger signals from place to place. Use the following table to determine what hardware lines each source and destination terminal refers to.

Note The following trigger terminals are not available on all PXI or PXIe modules. Refer to the device's hardware manual to determine if a trigger terminal is available on your device.

Trigger Terminal	Description
PXI_Trig<n>	The basic trigger lines of your PXI or PXIe chassis. PXI triggers go to and from all slots in the chassis—though the signals do not reach all slots at the same time—and all modules receive the same PXI triggers, so PXI_Trig0 is the same for Slot 3 as it is for Slot 4, and so on.
PXI_Star<n>	The star trigger lines of your PXI or PXIe chassis. Each trigger line is a dedicated connection between the system timing slot and one other slot. Star triggers are all of equal length, so signals routed via star trigger should reach their destinations at the same time if they were sent at the same time.  **Note** Each PXI_Star trigger is mapped to a single slot. This mapping is vendor specific. Refer to your hardware documentation to determine the orientation of PXI_Star lines in the chassis.

Trigger Terminal	Description
PFI<n>	The PFI connectors on the front panel of the module. You can use PFI connectors to route triggers between multiple chassis or devices.
PFI_LVDS<n>	The PFI low voltage differential signaling (LVDS) input/output connectors on the front panel of the device. PFI LVDS lines consist of paired PFI lines and can be used to route timing and triggering signals between multiple PXIe chassis at high speeds. You can achieve faster speeds when using an LVDS line compared to a single-ended PFI line. Signals on PFI LVDS lines use the standard PFI synchronization clock.
Ground (Source Only)	The Ground source continuously outputs a logic low signal, unless you invert it with niSync Connect Trigger Terminals.
Full Speed Clock (Source Only)	The full speed synchronization clock signal of the destination terminal zone. Use this source to send a full-speed clock signal along a trigger line (for example, to route a PXI_Clk10 clock signal to a PFI line). **Caution** Routing a clock signal through a PXI_Trig line is not recommended, due to poor clock signal integrity caused by the topology of the PXI_Trig lines. Use PXI_Star, PXIe_DStar, or PFI lines instead.
Divided Clock 1 (Source Only)	The first divided clock signal of the destination terminal zone. This source divides the synchronization clock of the destination terminal by the value you specify in the Clock Divisor 1 property and uses the result as the trigger source. Use this source to send a divided clock signal along a trigger line (for example, to route a divided DDS signal to a PXI_Star line).
Divided Clock 2(Source Only)	The second divided clock signal of the destination terminal zone. This source divides

Trigger Terminal	Description
	the synchronization clock of the destination terminal by the value you specify in the Clock Divisor 2 property and uses the result as the trigger source. Use this source to send a divided clock signal along a trigger line (for example, to route a divided PXI_Clk10 signal to a PXI_Star line).
ClkIn (Source Only)	The ClkIn connector on the front panel of your device. Use this terminal source to route triggers from an external device.
PXIe_DStarB<n>	The differential star trigger lines of your PXIe chassis. Use PXIe_DStarB lines to send trigger signals from the system timing slot to a peripheral slot of the chassis. **Note** Each PXIe_DStarB trigger is mapped to a single slot. This mapping is vendor specific. Refer to the chassis documentation to determine the orientation of differential star trigger lines.
PXIe_DStarC<n>	The differential star trigger lines of your PXIe chassis. Use PXIe_DStarC lines to send trigger and clock signals from a peripheral slot to the system timing slot of the chassis. **Note** Each PXIe_DStarC trigger is mapped to a single slot. This mapping is vendor specific.

Trigger Terminal Connections

Use the following table to determine which trigger source terminals are compatible with which trigger destination terminals. You can connect trigger terminals with niSync Connect Trigger Terminals.

Note

- Only devices in a system timing slot can use PXI_Star<n> terminals as a trigger source.
- Only devices in a system timing slot can use PXIe_DStarB<n> terminals as a trigger source.
- Only devices in a peripheral slot can use PXIe_DStarC<n> terminals as a trigger source.

Trigger Source	Trigger Destination
PXI_Trig<n>	• PXI_Trig<n> (except for source PXI_Trig) • PXI_Star<n> • PFI<n> • PFI_LVDS<n> • PXIe_DStarB<n> • PXIe_DStarC
PXI_Star<n>	• PXI_Trig<n> • PXI_Star<n> (except for source PXI_Star) • PFI<n> • PFI_LVDS<n> • PXIe_DStarB<n> • PXIe_DStarC
PXI_Star	Cannot be used as a trigger source terminal.
PFI<n>	• PXI_Trig<n> • PXI_Star<n> • PFI<n> (except for source PFI terminal) • PFI_LVDS<n> (except for source PFI terminal) • PXIe_DStarB<n> • PXIe_DStarC

Trigger Source	Trigger Destination
PFI_LVDS<n>	• PXI_Trig<n> • PXI_Star<n> • PFI<n> (except for source PFI_LVDS terminal) • PFI_LVDS<n> (except for source PFI_LVDS terminal) • PXIe_DStarB<n> • PXIe_DStarC
Ground	• PXI_Trig<n> • PXI_Star<n> • PFI<n> • PFI_LVDS<n> • PXIe_DStarB<n> • PXIe_DStarC
Synchronization Clock • Full Speed • Divided 1 • Divided 2	• PXI_Trig<n> • PXI_Star<n> • PFI<n> • PFI_LVDS<n> • PXIe_DStarB<n> • PXIe_DStarC
ClkIn	• PFI_LVDS<n>
PXIe_DStarC<n>	• PXI_Trig<n> • PFI<n> • PFI_LVDS<n> • PXIe_DStarB<n> • PXIe_DStarC
PXIe_DStarB	• PXI_Trig0 • PXI_Star<n>

Trigger Source	Trigger Destination
	• PFI<n> • PXIe_DStarB<n> • PXIe_DStarC

Clock Terminal Connections

Use the following table to determine which clock source terminals are compatible with which clock destination terminals. You can connect clock terminals with niSync Connect Clock Terminals.

Note

- Only devices in the system timing slot can use PXI_Clk10_In as a clock destination terminal.
- Only devices in the system timing slot can use PXI_Clk10 or PXIe_DStarA<n> terminals as a clock source terminal.
- Only devices in a peripheral slot can use the PXIe_DStarC<n> terminals as a clock source terminal.
- Refer to the hardware manual to determine if a clock source or destination terminal is available on your module.

Clock Source	Clock Destination
PXI_Clk10	• ClkOut • BoardClk
ClkIn	• PXI_Clk10_In • ClkOut • PFI_LVDS<n> • PXIe_DStarA<n>

Clock Source	Clock Destination
Oscillator	• PXI_Clk10_In • ClkOut • BoardClk
DDS Clock	• ClkOut • PFI_LVDS<n> • PXIe_DStarA<n>
PFI_LVDS <n>	• ClkOut • PFI_LVDS<n> • PXIe_DStarA<n>
PXIe_DStarC<n>	• ClkOut • PFI_LVDS<n> • PXIe_DStarA<n>
PXI_DStarA	• ClkOut • PFI_LVDS<n>

Defining NI-Sync Clock Terminals

Using niSync Connect Clock Terminals, you can route clocks between chassis, discipline the PXI backplane clock to the oscillator of a system timing module, or route an external time reference throughout a chassis or to multiple chassis. The function uses clock terminals to route clock signals between devices. The following terminals are available with NI-Sync:

Note Not all of the following terminals are available on every PXI or PXIe chassis. Refer to the hardware documentation to determine if your device supports a particular source or destination terminal.

Clock Source Terminals

Clock Source	Description
ClkIn	The ClkIn input connector on the front panel of the device. Using ClkIn, you can connect a 10 MHz reference clock directly to the PXI_Clk10_In pin. You also can phase lock a clock connected to ClkIn using a PLL circuit, or you can use a clock connected to ClkIn as the synchronization clock for the front and rear zones of your chassis.
PXI_Clk10	The 10 MHz backplane clock of the PXI or PXIe chassis.
Oscillator	The oscillator of the device specified in the instrument handle terminal.
DDS Clock	The DDS signal generated by the device specified in the instrument handle terminal.
PFI_LVDS<n>	The PFI low voltage differential signaling (LVDS) input/output connectors on the front panel of your device. PFI LVDS lines consist of paired PFI lines. You can use PFI_LVDS to route clock and trigger signals between multiple PXI chassis at high speeds. You can achieve faster speeds when using an LVDS line compared to a single-ended PFI line. Signals on PFI LVDS lines use the standard PFI synchronization clock.
PXIe_DStarC<n>	The differential star trigger line of your PXIe chassis. Use DStarC lines to route clock and/or trigger signals from a peripheral slot to a system timing slot. **Note** Each PXIe_DStarC trigger is mapped to a single slot. This mapping is vendor specific. Refer to the chassis documentation for more information on the mapping of differential star trigger lines.
PXIe_DStarA	The differential star trigger line of your PXIe chassis. Use DStarA lines to route clock signals

Clock Source	Description
	from a system timing slot to a peripheral slot. **Note** Each PXIe_DStarA trigger is mapped to a single slot. This mapping is vendor specific. Refer to the chassis documentation for more information on the mapping of differential star trigger lines.

Clock Destination Terminals

Clock Destination	Description
PXI_Clk10_In	The connector pin that you use to provide the backplane with a reference 10 MHz signal from the system timing slot. When you connect a signal to this pin, PXI_Clk10 and PXIe_Clk100 are phase aligned to this reference.
ClkOut	The ClkOut connector on the front panel of the device. Use this terminal to export clocks to an external device or to another chassis. You can export only clock signals through the ClkOut connector.
BoardClk	The timekeeper used to schedule future time events and timestamping on PXI-6682, PXI-6682H, PXI-6683, and PXI-6683H modules. BoardClk accepts a 10 MHz reference clock and multiplies it by 10 to create a 100 MHz clock for use as a timekeeper. The only two valid sources for BoardClk are the on-board oscillator and PXI_Clk10. You can specify the source of BoardClk using niSync Connect Clock Terminals. **Note** BoardClk is a valid terminal only on PXI-668x devices.
PFI_LVDS<n>	The PFI_LVDS output connector on the front panel of your device.

Clock Destination	Description
PXIe_DStarA<n>	The differential star trigger line of the PXIe chassis. Use DStarA lines to route clock signals from the system timing slot to a peripheral slot of the chassis.

GPS

Describes the GPS timing protocol.

GPS is a system of 24 triangulation satellites funded and controlled by the U.S. Department of Defense. The global positioning satellites (GPS) continually transmit their coordinates in space along with a time message on a 1.5 GHz carrier frequency. A time-based timing and synchronization module that has a GPS receiver can use this global timebase to precisely correlate, trigger, and timestamp measurement data. Each global positioning satellite contains multiple atomic clocks, which are controlled and referenced by the Master Clock (MC) at the United States Naval Observatory (USNO), called UTC (USNO).

Synchronizing to GPS Time

NI time-based synchronization modules can use GPS technology as a time reference. If you set GPS as the time reference for a module, the module uses the time updates received by the onboard GPS receiver every second, derives from it the current TAI time, and sets this time as the current board time.

When you initially connect a GPS antenna to a time-based synchronization module or set GPS as the time reference, the onboard GPS receiver searches for visible satellites. After detecting at least four satellites, the GPS receiver performs a self survey, which is the process of performing measurements of the visible satellites once per second and averaging those measurements to determine the receiver's current position as accurately as possible. During a self survey, you can use GPS as a time reference, but it is less accurate than after the self survey is finished. Once the GPS unit completes the self survey, you can precisely apply the time data received from GPS satellites.

If you configure the time-based synchronization module for mobile mode, a self-survey does not apply. If the antenna is moving and mobile mode is not enabled, you may get unexpected and invalid timing results. However, using mobile mode degrades the accuracy of the onboard GPS receiver, so you should not use it unless the antenna is moving.

Factors Affecting GPS Synchronization Accuracy

You can obtain the best GPS timing results by having an ideally located, long-term, stable GPS antenna installation. Ideally, you should mount the GPS antenna in a location where it has an unobstructed, clear view of the entire sky. In this orientation, the GPS receiver can detect additional satellites and perform additional averaging while discarding the worst signals and alleviating the effects of multipath, which is when the GPS antenna receives a signal reflected off an object or surface instead of a signal directly from the satellite(s).

Completing a self survey will also improve accuracy by performing long-term averaging of location. It is best to ensure the antenna is in a fixed location throughout the self survey process and throughout use, because any small movement of the antenna reduces accuracy. For this reason, you should also attempt to minimize the movement of a GPS antenna caused by wind or vibration.

Antenna cable latency also adds constant error. For maximum accuracy, you can calculate the latency of the GPS antenna cable in use and apply a correction. You can use the Time Reference Correction property to remove the source of this error. For example, if the antenna cable in use has a published latency of 5 ns/m, and the antenna installation uses 30 m of cable, the total delay that the antenna installation causes is 150 ns, so you can set the Time Reference Correction property to 150 to correct this. While not necessary in all cases, you can improve the accuracy of GPS by accounting for all sources of delay in the GPS installation, including cabling lightning arresters, and amplifiers.

You can also use the niSync Property Node to query the number of visible satellites using the Satellites Available property and determine if any fatal GPS errors are present using the Status parameter. A minimum of four satellites should be visible for stable GPS operations, and GPS clock accuracy and stability increase as the number of visible satellites increase.

IEEE 1588-2008

Details about the IEEE 1588-2008 timing protocol.

Using the IEEE 1588-2008 precision timing protocol (PTP), you can synchronize several

clocks connected via a multicast capable network, such as Ethernet. The protocol requires little network bandwidth overhead, processing power, and administrative setup. Depending on the hardware you use, IEEE 1588-2008 can provide submicrosecond synchronization over long distances with standard cabling. Refer to the device's hardware manual for typical performance results with NI-Sync.

NI-Sync is a hardware implementation of the IEEE 1588-2008 protocol; NI-Sync devices operate at the Layer 3 level and use the default profile.

Tip Visit `ni.com/info` and enter the info code `swsync` for more detailed information on the IEEE 1588 protocol and distributed clock technology.

Understanding Bidirectional Multicast Communication

IEEE 1588-2008 mainly uses bidirectional multicast communication to synchronize slave clocks to the 1588 grandmaster. The grandmaster clock periodically issues a sync packet, which is timestamped when it leaves the grandmaster clock. Optionally, the grandmaster may issue a follow up packet that contains the timestamp for the sync packet. Using a follow up packet, the grandmaster can accurately timestamp the sync packet on networks where you cannot know the departure time of a packet beforehand.

A slave clock receives the grandmaster's sync packet and timestamps the packet's arrival time using its own clock. The difference in the sync packet's departure timestamp and its arrival timestamp is the combination of the slave clock's offset from the grandmaster and the propagation delay of the network. When the slave adjusts its clock by the offset it measures when it receives the sync packet, it reduces the offset between the master and slave to the network propagation delay only.

The slave clock then compensates for the network propagation delay using a delay request packet. The delay request packet is timestamped when it departs from the slave, timestamped again when the master clock receives it, and then sent back to the slave clock with the new arrival timestamp. The difference between the arrival and the departure timestamps is the network propagation delay. Slave clocks can accurately measure the offset between their local clock and the master's clock using these synchronization packets, allowing them to adjust their clocks to match the time of the master. The IEEE 1588-2008 specification does not include any standard implementation for adjusting a clock, it merely provides a standard protocol for

exchanging the synchronization messages, allowing devices from different manufacturers and with different implementations to work together.

Actual performance on an IEEE 1588 network is highly application-specific. For example, the protocol does not specify the clock frequency in the master and slaves, so lower-frequency clocks have poorer time resolution, resulting in less-accurate timestamps in the PTP synchronization messages. 1588's performance is also dependent on clock stability: clocks based on TXCOs and OCXOs have a higher stability than the clocks using uncontrolled crystal oscillators. Clocks with lower stabilities drift apart faster, resulting in more frequent phase or frequency corrections and larger skews. Network topology also impacts IEEE 1588 performance.

Synchronizing Devices with IEEE 1588-2008

IEEE 1588-2008 uses two steps to synchronize devices: (1) the best master clock algorithm (BMC) determines which device serves as the master clock, and (2) the NI-Sync driver measures and corrects time skew caused by clock offsets and network delays.

The BMC defines a standard set of clock characteristics—such as the origin of a clock's time source and the stability of the clock's frequency—and then assigns a value to each clock in order to determine the highest ranking, most accurate clock on the network, which then becomes the grandmaster clock. The BMC continues to assess the quality of each clock, so if the grandmaster clock leaves the network or is no longer the most accurate clock, the algorithm automatically selects a new grandmaster. The BMC also takes into account user-defined priority values, which you can set using the Priority 1 and Priority 2 properties of niSync Properties.

You can use IEEE 1588 boundary clocks and transparent switches to measure and correct time skew across large networks. A boundary clock is a network switch with an accurate IEEE 1588 clock. A switch acting as a boundary clock runs the PTP protocol and is synchronized to an attached master clock. The boundary clock then acts as a master clock to all attached slaves. Boundary clocks do not pass Sync, Follow_Up, Delay_Req, or Delay_Resp messages. Within a subnet, a port of a boundary clock acts just like an ordinary clock with respect to synchronization and the BMC algorithm. Boundary clocks define a parent-child hierarchy of master-slave clocks: the boundary clock internally selects a port that sees the best clock as the single slave port, which then becomes a slave in the selected subnet. All other ports of the boundary clock

internally synchronize to this slave port. Using a boundary clock, all internal latencies and jitter in the switch can be compensated for and do not affect synchronization accuracy.

Transparent switches solve the same problem as boundary clocks, but in a slightly different manner. A transparent switch is so called because it does not operate as a PTP node in an IEEE 1588-2008 system. Instead, it modifies the timing contents of PTP packets to account for the delay caused by the switch. Typically, a transparent switch calculates how much time a sync packet spends inside the switch, and then modifies the timestamp of the associated follow up packet to account for this delay. The use of transparent switches allows PTP nodes to operate as if they were all part of one LAN segment connected by hubs.

IRIG-B

Describes the IRIG-B timing protocol.

IRIG-B is one of six serial protocols for distributing time codes currently defined by the Inter Range Instrumentation Group (IRIG). The six IRIG protocols mainly differ in the frame size or the amount of time before new information is sent.

IRIG-B is the most common IRIG standard and the one NI-Sync devices use. IRIG-B is an encoded TTL signal that carries the absolute time. It is similar to a pulse per second (PPS) signal, but instead of outputting a single uniform pulse every second, IRIG-B sends coded bits that make up a one-second-long data frame, and it repeats or re-synchronizes every second. IRIG-B sends coded bits that make up a one-second-long data frame, and it repeats or re-synchronizes every second. IRIG-B specifies that a 100-bit time frame is transmitted once per second, with each bit being represented as a 10 ms period.

The 0 bit represents 2 ms of a high logic state, the 1 bit represents a 5 ms high, and the P bit represents an 8 ms high. The P bit also separates seconds form minutes, minutes from hours, and so on, within the one-second frame. Data in the time frame includes Binary Coded Decimal (BCD) time of year, year, and straight binary seconds (SBS). The data can be DC biased (DC) or amplitude modulated (AM) with a 1 kHz sine wave.

The reception of the first bit of an IRIG-B data frame generates a timestamp for the

event. You cannot read or use the timestamp as a time reference until the entire IRIG-B frame has been received and decoded. After the frame is successfully decoded, the frame drives the time reference engine if you have set IRIG-B as the external time reference. Both the timestamp generated by receiving the first frame bit and the time/date encoded in the IRIG-B frame can be read using Read Last GPS or IRIG Timestamp.

PPS

Describes the pulse per second (PPS) time reference.

Pulse per second (PPS) is the simplest form of synchronization. PPS is a signal that outputs a high logic level once a second. It does not contain information about the specific time of day or year. The pulse width is generally 100 ms. You can input a PPS signal on a PFI, PXI_Trig, or PXI_Star line.

Setting PPS as the time reference for an NI-Sync device configures the device to interpret a rising edge on the configured input as representing a second's boundary. Since the PPS signal cannot indicate an absolute time, you can configure the device to use either a manual start time or its current board time, and use the PPS signal only to correct frequency.

If you configure the device to use a manual start time, the device will use the first pulse received on the PPS input terminal as the configured start time. The device interprets every subsequent pulse as occurring one second after the previous pulse. Using this configuration, you can synchronize multiple systems equipped with NI-Sync timing devices if absolute time is not a concern. You can use PPS as the time reference for all the systems you want to synchronize and then configure them with the same manual start time. You can then connect the PPS signal to the systems and start the PPS output. Since the systems are connected to the same signal with the same start time, they are closely synchronized.

If you configure the device to use current time instead of the manual start time, the device interprets the first pulse received as the time equal to the device's current time. The device applies no correction when the first pulse is received, so it interprets every subsequent pulse as occurring one second after the previous pulse. Using this configuration, you can synchronize multiple systems equipped with NI-Sync timing devices if absolute time is not a concern. You can use PPS as the time reference for all

the systems you want to synchronize and then configure them with the same manual start time. You can then connect the PPS signal to the systems and start the PPS output. Since the systems are connected to the same signal with the same start time, they are closely synchronized.

If you configure the device to use current time instead of the manual start time, the device interprets the first pulse received as the time equal to the device's current time. The device applies no correction when the first pulse is received, so it interprets every subsequent pulse as occurring one second after the previous pulse. You can use this configuration to distribute frequency corrections to multiple systems without concern for actual time values.

For best results when using PPS as a time reference, ensure that the device supplying the PPS signal provides a stable consistent 1 Hz signal. You can achieve optimal results when an OCXO, TCXO, or better oscillator drives the source signal. You can introduce error into the system if the reference signal contains significant jitter or the reference frequency strays from 1 Hz.