Identifying the Hardware Platform for Measurement Needs

This section outlines the various factors to consider when evaluating the measurement needs of your test system.

Identifying the Scope of Your Test System

The first step in identifying your test measurement needs is to determine the system's scope. Is the system testing a single product, an entire product line, or a series of product lines? Take a look at a simple example of how determining the scope can significantly change test system requirements.

Scenario 1: Testing a Single Product

Assume that you are a test engineer working for a semiconductor company. Your immediate goal is to design a system that can test the rise time, nonlinearity (integral nonlinearity or INL and differential nonlinearity or DNL), and current leakage specifications of the digital-to-analog converter (DAC) shown in Figure 1.

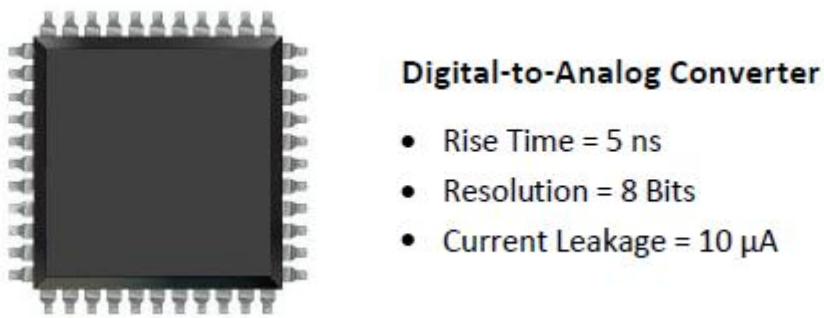

Figure 1. DAC under Test

To ensure that you test the device rather than the test system, you need a set of instruments with better specifications than the DAC under test. Thus, your test system must have a high-speed measurement instrument with a rise time that is faster than 5 ns. In order to accurately capture the rise time appropriately, you need to measure at 10x the rate of nyquist, which would require instruments with a sample rate greater than 2 GHz. In addition, you must fit the system with an instrument that has a current sensitivity greater than 10 µA. Finally, the system must have an instrument with resolution greater than 8 bits to appropriately measure the DAC code widths and perform the nonlinearity tests.

Scenario 2: Testing an Entire Product Family

Now consider building a system that can test the rise time, INL and DNL, and current leakage specifications of the entire family of analog-to-digital converters (ADCs) shown in Figure 2.

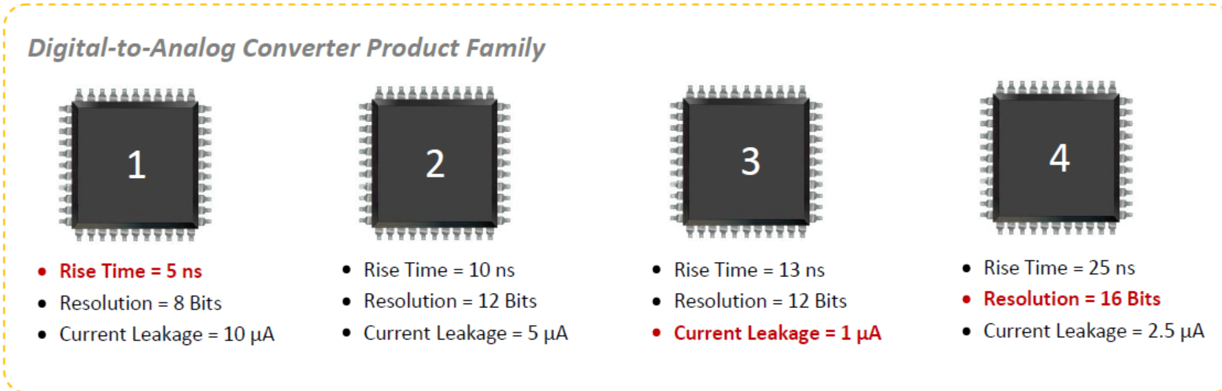

Figure 2. DAC Product Family under Test

To test the DAC product family shown in Figure 2, your system must incorporate instruments that have specifications superior to that of the entire product family. Thus, the instruments in the test system must have the following:

a) A rise time that is faster than 5 ns or a sample rate greater than 2 GHz
b) Current sensitivity greater than 1 µA
c) Resolution higher than 16 bits

Scenario 3: Testing Multiple Product Families

It is tempting to widen the test system's scope as much as possible to have a common platform for several programs; however, the following pitfalls can occur:

- To accommodate different product lines, the complexity of the core test system increases, thus increasing nonrecurring, recurring, and material costs
- Maintaining configuration control is difficult among a larger group
- Obsolescence issues increase
- Costs increase on high-production-rate product lines requiring multiple test systems even though a DUT may use only a small portion of the test system capabilities
- Designing test systems for a new product line becomes difficult because of the constraints to use only the existing capabilities of the system
- Keeping up with state-of-the-art technology grows more difficult as test capabilities start to stagnate

Future Plans and Other Considerations

In addition to understanding your current tester needs, you need to evaluate its future requirements. Are you going to use the tester to test additional product families going forward? Will you add new products to the current product family? If you answered "yes," then you must also consider the measurement needs of these future additions. If you are certain that your test system needs will expand but are unsure of the measurement requirements of your future products, you must design your system using a modular platform that is easily scalable. For example, you should make interfaces easily available to your test system such as USB, LAN, and GPIB so that you can quickly add new measurement capability to the system that is not available in the rack such as a USB-based modular instrument.

Other Considerations

- Budget and timeline
- Expected life span of the test system

- Additional test requirements such as fault diagnostic capability
- Skill level of operators
- Product volume

Avoiding Scope Creep

Ensure you understand the project vision and spend time documenting and determining the project objectives. Produce a project plan document that describes the test system deliverables. It is a good idea to document what is in scope as well as out of scope for absolute clarity. Verify the content of this document with the key stakeholder, spending time to walk them through it, and ask them to sign off on it. You should plan for some degree of scope creep in most projects; therefore, it is important to design a process to manage these changes. You can then implement a simple process of document, consider, approve, and assign resources. Use a change control form and change log from the start of the project and communicate the process for using these forms to the customer and project team. Attach a cost and time to each change so the customer is clear about its impact. Implementing a formal process helps ensure there's a clear business value for the requested change.

Choosing a Core Hardware Platform

After determining the measurement needs of your test system, you can begin architecting your hardware framework. Many test engineers jump straight into matching their measurement needs to instruments available on the market. A better approach is to first pinpoint a suitable test platform that can serve as the core or nucleus of your test system. You can choose from many platforms, most of which are based on one of the four most commonly used instrument backplanes/buses –PXI, GPIB, USB, and LAN. Because each of these buses has at least some advantages and limitations, you often have to build hybrid test systems based on multiple platforms. Even so, it is often a best practice to pick a prominent or core platform for your test architecture. This section outlines some of the factors you must consider when choosing a core platform for your test system.

Processing Power and Data Throughput

Assess the worst-case computational power and throughput rates when selecting a controller.

Scalability

Another factor is the ease with which you can scale or modify your system. This is especially important if your test system has the potential to change during the course of its lifetime. One example of this is if you are building a system to test a product family that is continually expanding. In such a case, you may need to add new functionality to the system without making significant changes that could force you to redesign your test rack.

Measurements Diversity

The platform that serves as the core of your test system must be able to address a significant portion of your test system needs. Thus, if your system requires the ability to make low-level DC measurements along with high-speed rise time measurements, you must select a platform that is capable of accommodating mixed-signal instrumentation. In general, you should choose a core platform that accommodates at least 80 percent of your test system's measurement needs.

Communication with Other Buses and Instruments

As mentioned previously, each instrument bus and platform has distinct advantages and disadvantages. By building hybrid systems based on multiple instrument buses, you can take advantage of the strengths of several different test platforms. A hybrid architecture also increases the flexibility of your test system by allowing you to choose from a larger pool of instruments on the market. Such flexibility is especially important if you are building a complex and dynamic test system that will change over time. The first step toward building a hybrid architecture is choosing a core platform that can communicate with instruments that are based on a variety of instrument buses.

Timing and Synchronization

When designing a test system composed of multiple buses and platforms, you must ensure that your core platform can synchronize those instruments by sending triggers and sharing clocks.

Lifetime

Another factor to consider is the lifetime of your test system. If you expect to use your system for several years, you should choose a platform that can stand the test of time. Sometimes products and platforms go end of life (EOL). It is often difficult to service and maintain products like these in a test system. For this reason, you must choose a proven platform for which products and replacements will be available for several years. For long military programs, consider vendor support agreements or lifetime buys of equipment that may be required.

Choosing the Right Data Bus

GPIB, USB, PCI/PCI Express, PXI/PXI Express, and Ethernet/LAN are some of the most popular communication buses available for automated test systems. The challenge for today’s test engineer is not to choose a single bus or platform on which to standardize every single application, but to choose a bus or platform appropriate for a specific application or even a specific part of an application. This chapter presents a head-to-head comparison of the most popular instrumentation buses, so that you can make informed decisions when choosing the bus and platform technologies to meet your application-specific needs. Specific bus technologies discussed below include GPIB, USB, PCI, PCI Express, and Ethernet/LAN/LXI.

Understanding Bus Performance

First, it is important to outline the relevant performance criteria for instrument control buses, in order to set a baseline for evaluation and comparison.

Bandwidth

When considering the technical merits of alternative buses, bandwidth and latency are two of the most important bus characteristics. Bandwidth measures the rate at which data is sent across the bus, typically in MB/s (106 bytes per second). A bus with high bandwidth is able to transmit more data in a given period than a bus with low bandwidth. Most users recognize the importance of bandwidth because it affects whether their data can be sent across the bus to or from a shared host processor as fast as it is acquired or generated and how much onboard memory their instruments will need. Bandwidth is important in applications such as complex waveform generation and acquisition as well as RF and communications applications. High-speed data transfer is particularly important for virtual and synthetic instrumentation architectures. The functionality and personality of a virtual or synthetic instrument is defined by software; in most cases, this means data must be moved to a host PC for processing and analysis. Figure 3 charts the bandwidth (and latency) of all the instrumentation buses examined in this paper.

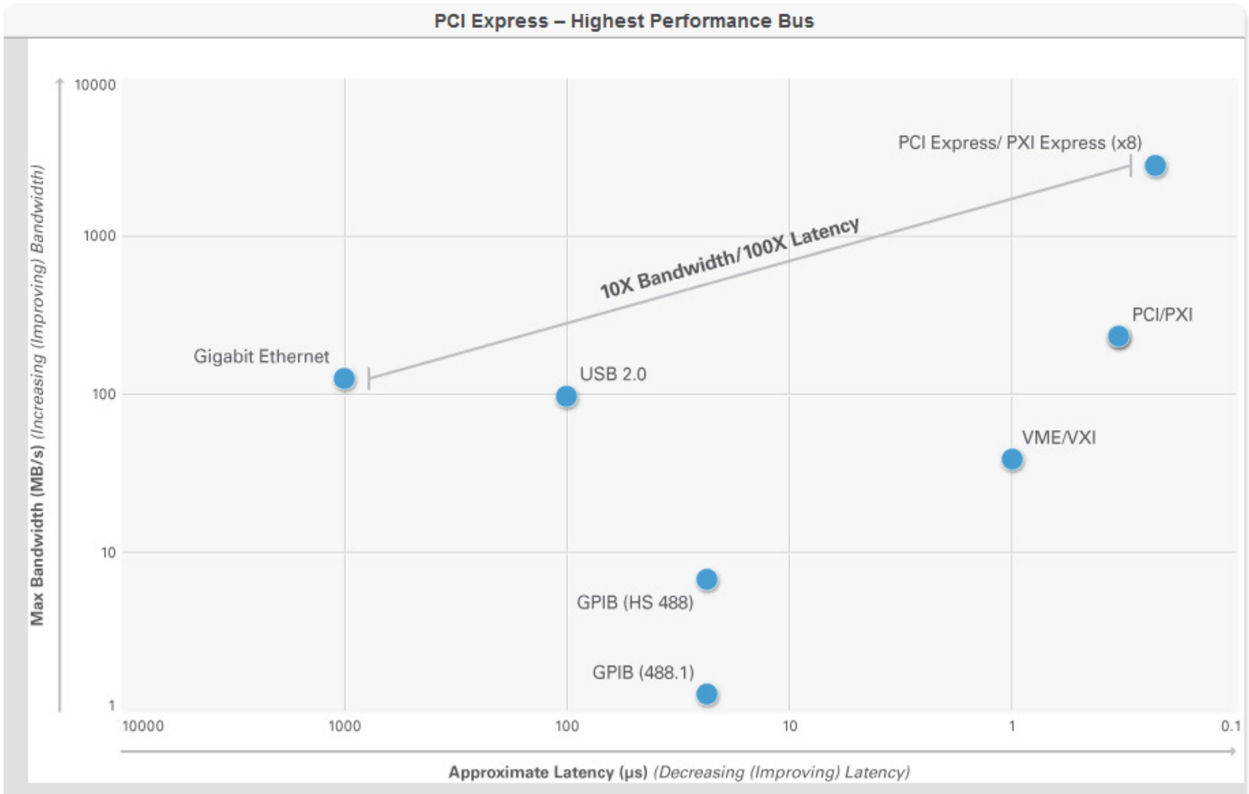

Figure 3. Bandwidth versus Latency for Instrumentation Buses.

Latency

Latency measures the delay in data transmission across the bus. By analogy, if we were to compare an instrumentation bus to a highway, bandwidth would correspond to the number of lanes and the speed of travel, while latency would correspond to the delay introduced at the on and off-ramps. A bus with low (meaning good) latency would introduce less delay between the time data was transmitted on one end and processed on the other end. Latency, while less observable than bandwidth, has a direct impact on applications where a quick succession of short, choppy commands are sent across the bus, such as in handshaking between a digital multimeter (DMM) and switch, and in instrument configuration.

Message versus Register-Based Communication

Buses that use message-based communication are generally slower because this mode of communication adds overhead in the form of command interpretation and padding around the data. With register-based communication, data transfer occurs by directly writing and reading binary data to and from hardware registers on the device, resulting in a faster transfer. Register-based communication protocols are most common to internal PC buses, where interconnects are physically shorter and the highest throughput is required. Message-based communication protocols are useful for transmitting data over longer distances and where higher overhead costs are acceptable.

Long-Range Performance

For remote monitoring applications and for systems that involve measurement over a large geographical area, range becomes important. Performance in this category can be viewed as a tradeoff with latency, because the error checking and message padding added to overcome physical limitations of sending data over longer cables can add delays to sending and receiving the data.

Instrument Setup and Software Performance

Ease of use in terms of instrument setup and software performance is the most subjective criterion examined here. Nonetheless, it is important to discuss. Instrument setup describes the “out of the box” experience and setup time. Software performance relates to how easily you can find interactive utilities or standard programming APIs such as VISA to communicate with and control the instrument.

Robustness of Connector

The physical connector for the bus affects whether it is suitable for industrial applications and whether additional effort will be required to “ruggedize” the connection between the instrument and the system controller.

Instrument Control Bus Comparison (GPIB, USB, PCI, PCI Express, and Ethernet/LAN/LXI)

GPIB

The first bus we will look at is the IEEE 488 bus, familiarly known as GPIB (general-purpose interface bus). GPIB is a proven bus designed specifically for instrument control applications. GPIB has been a robust, reliable communication bus for over 30 years and is still the most popular choice for instrument control because of its low latency and acceptable bandwidth. It currently enjoys the widest industry adoption, with a base of more than 10,000 instrument models with GPIB connectivity.

With a maximum bandwidth of about 1.8 MB/s, it is best suited for communicating with and controlling stand-alone instruments. The more recent, high-speed revision, HS488, increased bandwidth up to 8 MB/s. Transfers are message-based, often in the form of ASCII characters. Multiple GPIB instruments can be cabled together to a total distance of 20 m, and bandwidth is shared among all instruments on the bus. Despite relatively lower bandwidth, GPIB latency is significantly lower (better) than that of USB and especially Ethernet. GPIB instruments do not auto detect or auto configure when connected to the system; though GPIB software is among the best available, and the rugged cable and connector are suitable for the most demanding physical environments. GPIB is ideal for automating existing equipment or for systems requiring highly specialized instruments.

USB

USB (universal serial bus) has become popular in recent years for connecting computer peripherals. That popularity has spilled over into test and measurement, with an increasing number of instrument vendors adding USB device controller capabilities to their instruments.

Hi-Speed USB has a maximum transfer rate of 60 MB/s, making it an attractive alternative for instrument connectivity and control of stand-alone and some virtual instruments with data rates below 1 MS/s. Though most laptops, desktops, and servers may have several USB ports, those ports usually all connect to the same host controller, so the USB bandwidth is shared among all the ports. Latency for USB falls into the better category (between Ethernet at the slow end and PCI and PCI Express at the fast end), and cable length is limited to 5 m. USB devices benefit from auto detection, which means that unlike other technologies such as LAN or GPIB, USB devices are immediately recognized and configured by the PC when a user connects them. USB connectors are the least robust and least secure of the buses examined here. External cable ties may be needed to keep them in place.

USB devices are well-suited for applications with portable measurements, laptop or desktop data logging, and in-vehicle data acquisition. The bus has become a popular communication choice for stand-alone instruments due to its ubiquity on PCs and especially due to its plug-and-play ease of use. The USB Test and Measurement Class (USBTMC) specification addresses the communication requirements of a broad range of test and measurement devices.

PCI

PCI and PCI Express achieve the best bandwidth and latency specifications among all the instrumentation buses examined here. PCI bandwidth is 132 MB/s, with that bandwidth shared across all devices on the bus. PCI latency performance is outstanding; benchmarked at 700 ns, compared to 1 ms in Ethernet. PCI uses register-based communication. Unlike the other buses mentioned here, PCI does not cable to external instruments. Instead, it is an internal PC bus used for PC plug-in cards and in modular instrumentation systems such as PXI, so distance measures do not directly apply. Nonetheless, the PCI bus can be “extended” by up to 200 m by the use of NI fiber-optic MXI interfaces when connecting to a PXI system. Because the PCI connection is internal to the computer, it is probably fair to characterize the connector robustness as being constrained by the stability and ruggedness of the PC in which it resides. PXI modular instrumentation systems, which are built around PCI signaling, enhance this connectivity with a high-performance backplane connector and multiple screw terminals to keep connections in place. Once booted with PCI or PXI modules in place, Windows automatically detects and installs the drivers for modules.

An advantage that PCI (and PCI Express) share with Ethernet and USB is that they are universally available in PCs. PCI is one of the most widely adopted standards in the history of the PC industry. Today, every desktop PC has either PCI slots, PCI Express slots, or both. In general, PCI instruments can achieve lower costs, because these instruments rely on the power source, processor, display, and memory of the PC that hosts them, rather than incorporating that hardware in the instrument itself.

PCI Express

PCI Express is similar to PCI. It is the latest evolution of the PCI standard, much as Hi-Speed USB is to USB. Therefore, much of the above evaluation of PCI applies to PCI Express as well.

The main difference between PCI and PCI Express performance is that PCI Express is a higher bandwidth bus and gives dedicated bandwidth to each device. Of all the buses covered in this tutorial, only PCI express offers dedicated bandwidth to each peripheral on the bus. GPIB, USB, and LAN, divide bandwidth across the connected peripherals. Data is transmitted across point-to-point connections called lanes at 250 MB/s per direction for PCI Express Gen1. Each PCI Express link can be composed of a multiple lanes, so the bandwidth of the PCI Express bus depends on how it is implemented in the slot and device. A x1 (by 1) Gen1 link provides 250 MB/s; a x4 Gen1 link provides 1 GB/s; and a x16 Gen1 link provides 4 GB/s dedicated bandwidth. It is important to note that PCI Express achieves software backward compatibility, meaning that users moving to the PCI Express standard can preserve their software investments in PCI. PCI Express also extensible by external cabling. PCI Express continues to evolve as the PCI-SIG (PCI Special Interest Group) introduced PCI Express Gen2 which doubles the bandwidth of PCI Express Gen1.

High-speed, internal PC buses were designed for rapid communication. Consequently PCI and PCI Express are ideal bus choices for high-performance, data-intensive systems where large bandwidth is required, and for integrating and synchronizing several types of instruments.

Ethernet/LAN/LXI

Ethernet has long been an instrument control option. It is a mature bus technology and has been widely used in many application areas outside of test and measurement. 100BaseT Ethernet has a theoretical max bandwidth of 12.5 MB/s. Gigabit Ethernet, or 1000BaseT, increases the max bandwidth to 125 MB/s. In all cases, Ethernet bandwidth is shared across the network. At 125 MB/s Gigabit Ethernet is theoretically faster than Hi-Speed USB, but this performance quickly declines when multiple instruments and other devices are sharing network bandwidth. Communication along the bus is message based, with communication packets adding significant overhead to data transmission. For this reason, Ethernet has the worst latency of the bus technologies featured in this tutorial.

Nonetheless, Ethernet remains a powerful option for creating a network of distributed systems. It can operate at distances up to 85 to 100 m without repeaters and with repeaters has no distance limits. No other bus has this range of separation from the controlling PC or platform. As with GPIB, auto configuration is not available on Ethernet/LAN. Users must manually assign an IP address and subnet configuration to their instrument. Like USB and PCI, Ethernet/LAN connections are ubiquitous in modern PCs. This makes Ethernet ideal for distributed systems and remote monitoring. It is often used in conjunction with other bus and platform technologies to connect measurement system nodes. These local nodes may themselves be composed of measurement systems relying on GPIB, USB, and PCI. Physical Ethernet connections are more robust than USB connections, but less so than GPIB or PXI.

LXI (LAN eXtenstions for Instrumentation) is an emerging LAN-based standard. The LXI standard defines a specification for stand-alone instruments with Ethernet connectivity that adds triggering and synchronization features.

Conclusion: Instrument Bus Performance

Despite the conceptual convenience of designating a single bus or communication standard as the “ultimate” or “ideal” technology, history teaches us that several standards are likely to continue to coexist, since each bus technology has unique strengths and weaknesses. Table 1 compiles the performance criteria from the previous section. It should be clear that no single bus is superior across all measures of performance.

	Bandwidth (MB/s)	Latency (µs)	Range (m) (without extenders)	Setup and Installation	Connector Ruggedness
GPIB	1.8 (488.1) 8 (HS488)	30	20	Good	Best
USB	60 (Hi-Speed)	1000 (USB) 125 (Hi-Speed)	5	Best	Good
PCI	132	0.7	Internal PC bus	Better	Better Best (for PXI)
PCI Express	250 (x1) 4000 (x16)	0.7 (x1) 0.7 (x4)	Internal PC bus	Better	Better Best (for PXI)

Table 1. Bus Performance Comparison.

You can exploit the strengths of several buses and platforms by creating hybrid systems. Hybrid test and measurement systems combine components from modular instrumentation platforms such as PXI and VXI and stand-alone instruments that connect across GPIB, USB, and Ethernet/LAN. One key to creating and maintaining a hybrid system is implementing a system architecture that transparently recognizes multiple bus technologies and takes advantage of an open, multivendor computing platform, such as PXI, to achieve I/O connectivity.

The other key to successfully developing a hybrid system is ensuring that the software you choose at the driver, application, and test system management levels is modular. Though some vendors may offer vertical software solutions for specific instruments, the most useful system architecture is one which breaks up the software functions into interchangeable modular layers so that your system is neither tied to a particular piece of hardware or to a particular vendor. This layered approach provides the best code reuse, modularity, and longevity. For example, VISA (Virtual Instrument Software Architecture) is a vendor-neutral software standard for configuring, programming, and troubleshooting instrumentation systems comprising GPIB, VXI, serial (RS232/485), Ethernet, USB, and/or IEEE 1394 interfaces. It is a useful tool because the API for programming VISA functions is similar for a variety of communication interfaces.

With hybrid systems, you can combine the strengths of many types of instruments, including legacy equipment and specialized devices. Despite the appeal of finding a one-size-fits-all solution for instrumentation, reality requires that you fit the instruments and associated bus technologies to your specific application needs.

Determining the Required Instrumentation

Now that you have a better understanding for determining your test system measurement requirements and choosing your hardware platform, you are ready to start selecting the specific hardware instruments you need to conduct your measurements. This section features some best practices for choosing instruments for your tester.

Basing Instrument Choices on Measurement/Stimulus Rather Than Instrument Type

Test engineers often choose an instrument based on type rather than need. For example, many engineers select DMMs to make high-accuracy measurements even though in many applications, the accuracy of a data acquisition board may be sufficient. Such decisions often result in higher costs, so you should choose your instrument based on your measurement need rather than the instrument type.

Test Accuracy Ratio

Another best practice for choosing your test system instruments is to calculate the test accuracy ratio (TAR) to ensure that the accuracy of your measurement equipment is substantially larger than the accuracy of the component you are testing. If you do not meet this criterion, then you may see significant measurement error caused by both the device under test and the test equipment, making it impossible to know the true source of error. Because of this, engineers use TAR to determine the relative accuracy of the measurement equipment and the component under test. You can calculate TAR with the following formula:

TAR =Desired Accuracy of the Component Under Test/Accuracy of Measurement Equipment

Choosing Modular Instruments

Modular instrument is the term given to the modular hardware that fits into the virtual instrumentation architecture, as illustrated by Figure 4.

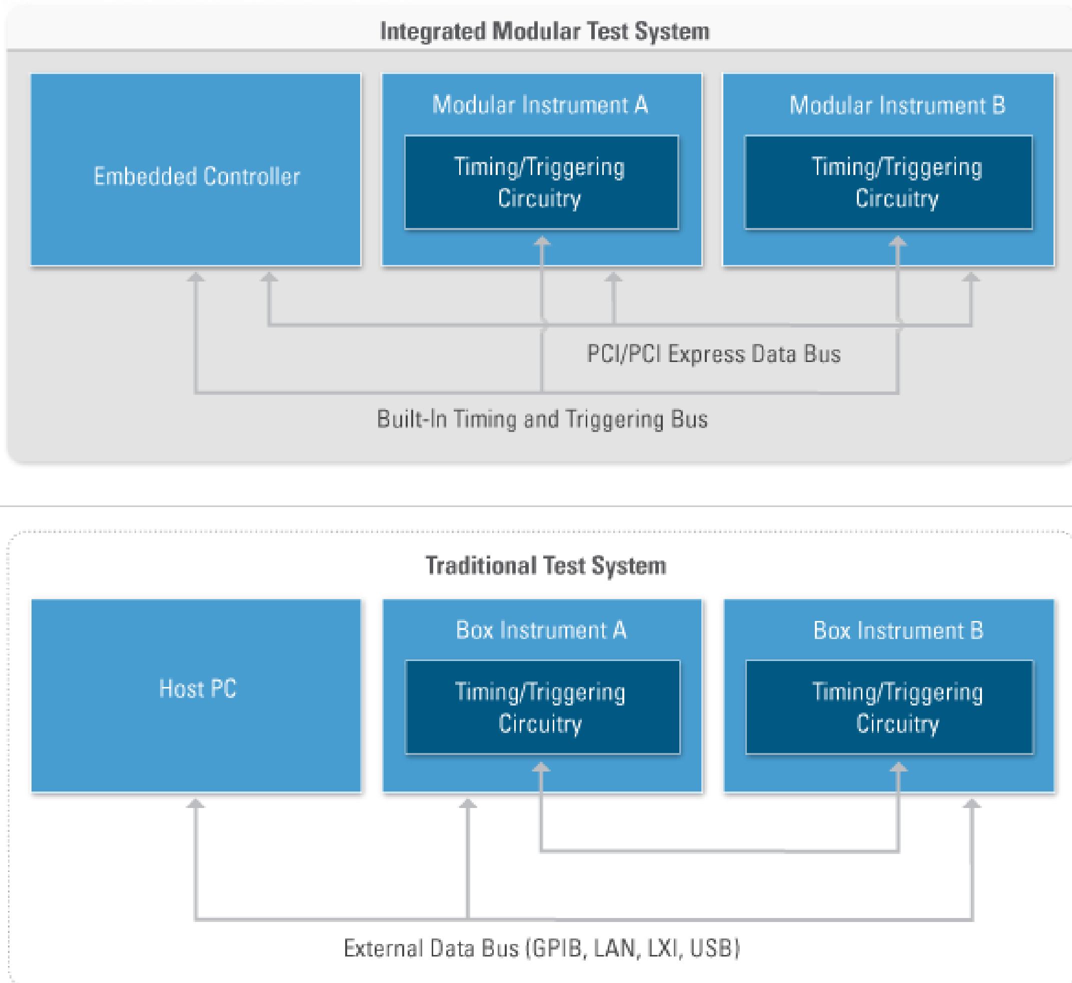

Figure 4. Comparing traditional and virtual instrumentation architectures.

To understand how modular instruments work, it is first important to understand the similarities and differences between the two approaches shown in Figure 4.

As can be seen, both approaches have measurement hardware, a chassis, a power supply, a bus, a processor, an OS, and a user interface. Because the approaches use the same basic components, the most obvious difference from a purely hardware standpoint is how the components are packaged. A traditional, or stand-alone, instrument puts all of the components in the same box for every discrete instrument. By contrast, in a well-designed modular instrumentation system, many of the components - such as the bus, power supply, OS and User Interface - are shared across instrument modules instead of duplicating these components for every instrument function. These instrument modules can also include different types of hardware, including oscilloscopes, function generators, digital, and RF.

While modular instruments can be designed for a variety of platforms, this tutorial will discuss various specifications with respect to modular instruments designed for the PCI Extension for Instrumentation (PXI) platform - a rugged platform for test, measurement, and control supported by more than 60 member companies.

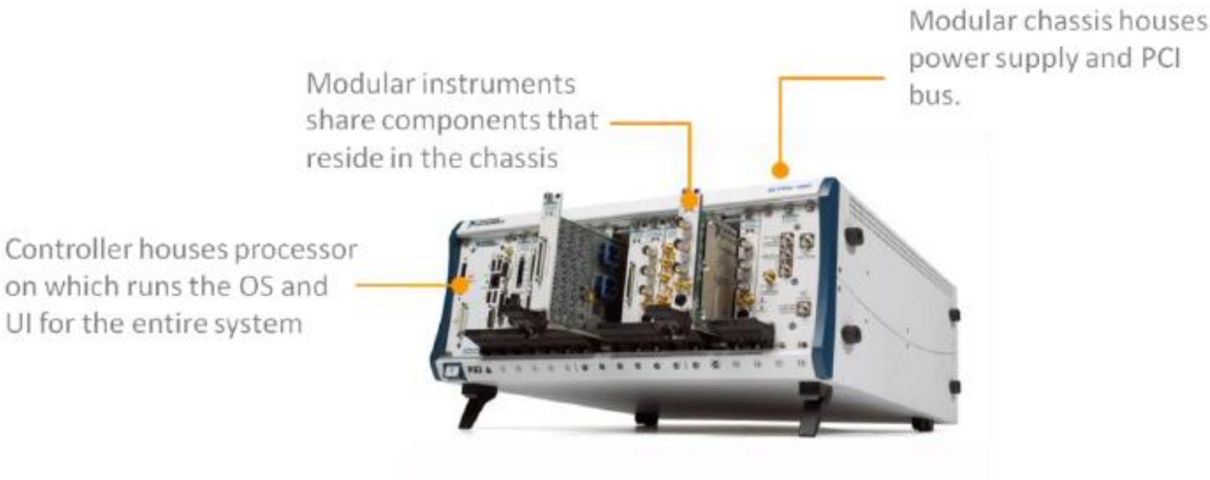

Figure 5. The PXI platform supports modular instrumentation.

Anatomy of a Modular Instrument

In order for you to choose the best possible set of instruments for your application, it is important for you to understand the various components that make up a modular instrument and also the impact that these components have on fundamental instrument specifications such as bandwidth, resolution, accuracy, and sampling rate.

This tutorial will describe the anatomy of a modular instrument with inputs and provide insight into how the components of such an instrument can impact various specifications.

Modular instruments typically consist of four main components – analog inputs, analog front-end, analog to digital converter (or digital to analog converter in the case of instruments that have outputs), and connection to chassis.

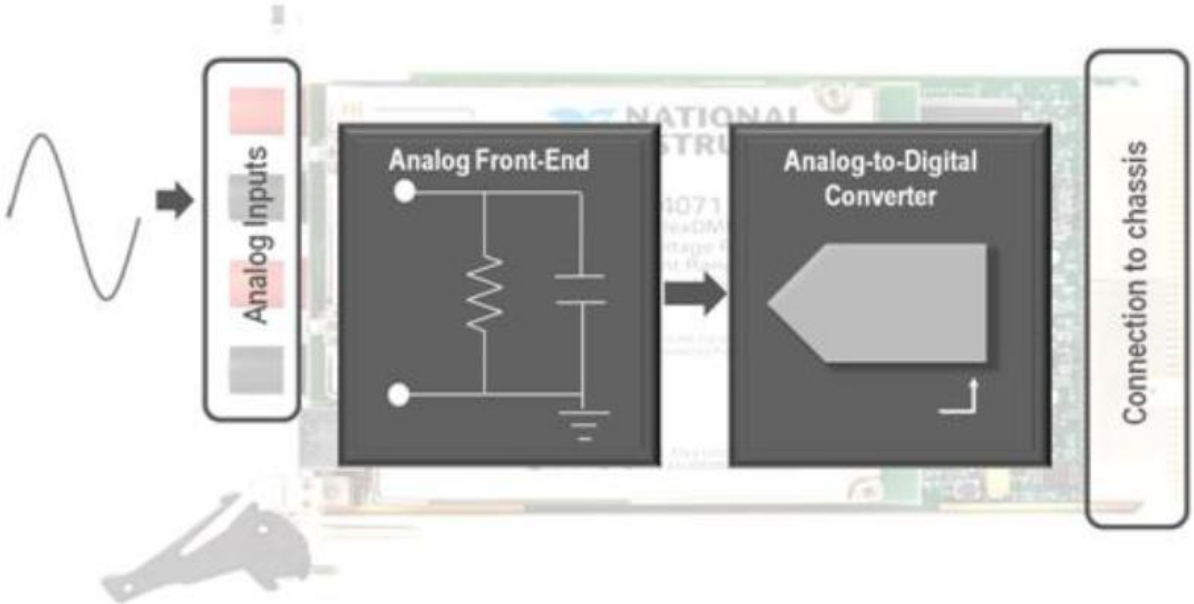

Figure 6. Anatomy of a typical PXI modular instrument.

Analog Inputs: All modular instruments have input connectors. This is where the analog signal enters the device. The type of connector, however, varies from instrument to instrument. DMMs for example use Banana connectors because provide a solid connection with the input signal and, thereby reduce noise and ensure accurate DC measurements. On the other hand, many digitizers use BNC connectors due to their higher bandwidth.

Figure 7. DMM connectors are optimized for accuracy. Digitizer connectors are optimized for Bandwidth.

Connection to Chassis: Modular instruments made for the PXI platform will also have a PXI or PXI Express (PXIe) connector at the back. This component connects the instrument to the chassis backplane and enables synchronization between multiple instruments in a chassis using the trigger lines on the backplane.

Analog Front-End: All instruments also possess an analog front-end. This is essentially the analog circuitry that is required to condition the input signal. The analog front-end is typically optimized either for bandwidth or accuracy. The front end of a DMM for example has an ovenstabilized onboard reference that is used to calibrate the DMM before every measurement in order to ensure maximum accuracy. The front-end is also optimized to reduce noise.

On the other hand, the analog-front end of higher speed instruments such as digitizers and arbitrary waveform generators are designed using components that are impedance matched in order to reduce attenuation at higher bandwidth values.

Typically, there is a tradeoff between designing an instrument's front-end for accuracy and bandwidth. Thus, instruments with front-ends that have high bandwidth typically have low accuracy. The same is also true for the opposite case.

Analog-to-digital converter (ADC): After the input signal passes through the front-end circuitry, it is converted in to digital values that can be read by the computer using an analog-to-digital converter or ADC. The instrument's ADC determines how fast it can sample an input signal. It can also help determine the instrument's resolution.

Note: In the case of instruments that output a signal, the ADC is replaced with a digital-to-analog converter or DAC.

Effects of Front End

The previous section of this paper discussed how the analog front-end of an instrument can impact its bandwidth and accuracy. This section will not define what the terms bandwidth and accuracy mean and also provide you with insight on how these specifications can impact your signal measurements.

Bandwidth: Bandwidth of an instrument is more clearly defined as the frequency at which a sinusoidal signal is attenuated by the analog front-end to 70% of its original amplitude. This is also commonly known as the 3 dB point. Bandwidth of an instrument depends on the bandwidth of its front-end, which takes the form of an RC circuit and acts as a low-pass filter.

Accuracy: There is always some uncertainty in any measurement made by an instrument. Accuracy is a measure of this uncertainty. To better understand accuracy, let's take a look at the example of the Omega chronometer watch series. Omega Chronometer watches mechanical movement's average daily variation range between -4/+6 seconds per day or 69.4 parts per million (ppm). This is an accuracy of 99.99%, the highest accuracy attainable by a mechanical movement.

Different instruments offer different accuracy for different measurement ranges and conditions. It is therefore very important to carefully evaluate the specifications listed in the datasheet to determine whether a particular product suits your measurement needs.

Effects of ADC/DAC

This section will describe in greater deal the impact that sampling rate and resolution, specifications impacted by the ADC of an instrument, can have on your measurements.

Sampling Rate

Sampling rate is the maximum speed at which the ADC of an instrument can convert the input signal into digital values that represent the voltage level. One thing to remember when considering the sampling rate of an instrument is the Nyquist Theorem which states that a signal must be sampled at a rate greater than twice the highest frequency component of interest in the signal to capture the highest frequency component of interest. Even at a rate of 2x, the signal will appear significantly distorted as can be seen in Figure 8, where a 100 MHz sine wave is being acquired by an ADC that can only sample at 200 MS/s or 200 MHz.

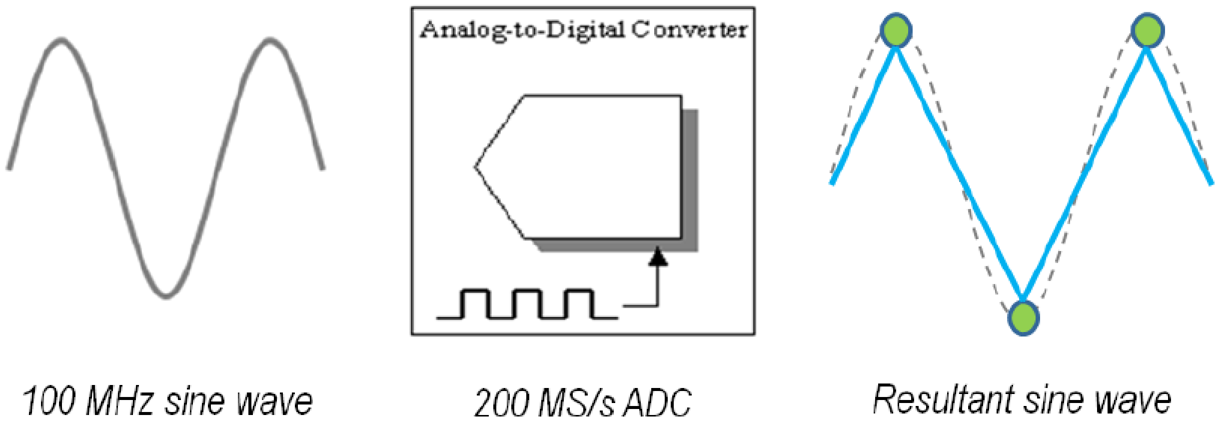

Figure 8: 100 MHz sine wave sampled by 200 MS/s appears distorted.

In order to attain an accurate representation of the signal, the instrument must have a sample rate which is at least 5x the speed of the signal.

Resolution

Resolution is another instrument specification that is directly impacted by the ADC. Resolution is defined as the smallest amount of input signal *change* that an instrument or sensor can detect reliably. Typically, in analog instruments such as digitizers, arbitrary waveform generators (arbs), and dynamic signal analyzers (DSA), resolution is represented in bits, while in the case of precision DC instruments such as DMMs, it is represented in digits.

Bits

The resolution of analog instruments (digitizers, arbs, DSA, etc.) is expressed in bits, and is a characteristic that is directly tied to the ADC (or DAC in the case of arbs) used in the instrument. An 18-bit digitizer, for example, uses an 18-bit ADC. In analog instruments, the smallest possible change in the input signal that can be detected, also known as the least significant bit (LSB) can be calculated using the following formula:

$$Least\ Significant\ Bit\ (LSB) = \frac{Input\ Range}{2^{number\ of\ bits\ on\ ADC/DAC}}$$

Digits

The resolution of precision DC instruments such as DMMs is represented in digits. The number of digits is used to specify the number of meaningful counts, or unique digitized values, the DMM is capable of producing. For a traditional benchtop DMM, the number of digits indicates how many decimal places the DMM can display on its digital readout. The number of digits is often specified as the number of full digits, digits that can take any value from 0 to 9, and a single overrange digit, referred to as the ½ digit. The overrange digit is limited to only specific values, generally 0 or 1. For example a 6½ digit DMM would have seven decimal places on its readout. The most significant digit on the display could take on a value of 0 or 1; the rest could range from 0 to 9.

Figure 9. Example reading obtained from a 6½-Digit DMM.

There is a loose relationship between digits and bits. For example for a noise-free DMM which uses a 12-bit ADC to digitize signals, the digits of resolution can be calculated using the following formula:

$$Digits\ of\ Resolution = \log_{10}(Number\ of\ LSB) = \log_{10}(2^{12}) = 3.61\ digits$$

In reality, however, a noise-free DMM does not exist. Noise may reduce the number of LSB, therefore reducing the number of digits. As a result, you need to account for the noise level when calculating the effective number of digits (ENOD) for a DMM. This can be done using the following formula:

$$ENOD = \log_{10}\left(\frac{Total\ Span}{\sqrt{12} \times RMS\ Noise}\right)$$

Anatomy of a Modular Hardware Architecture

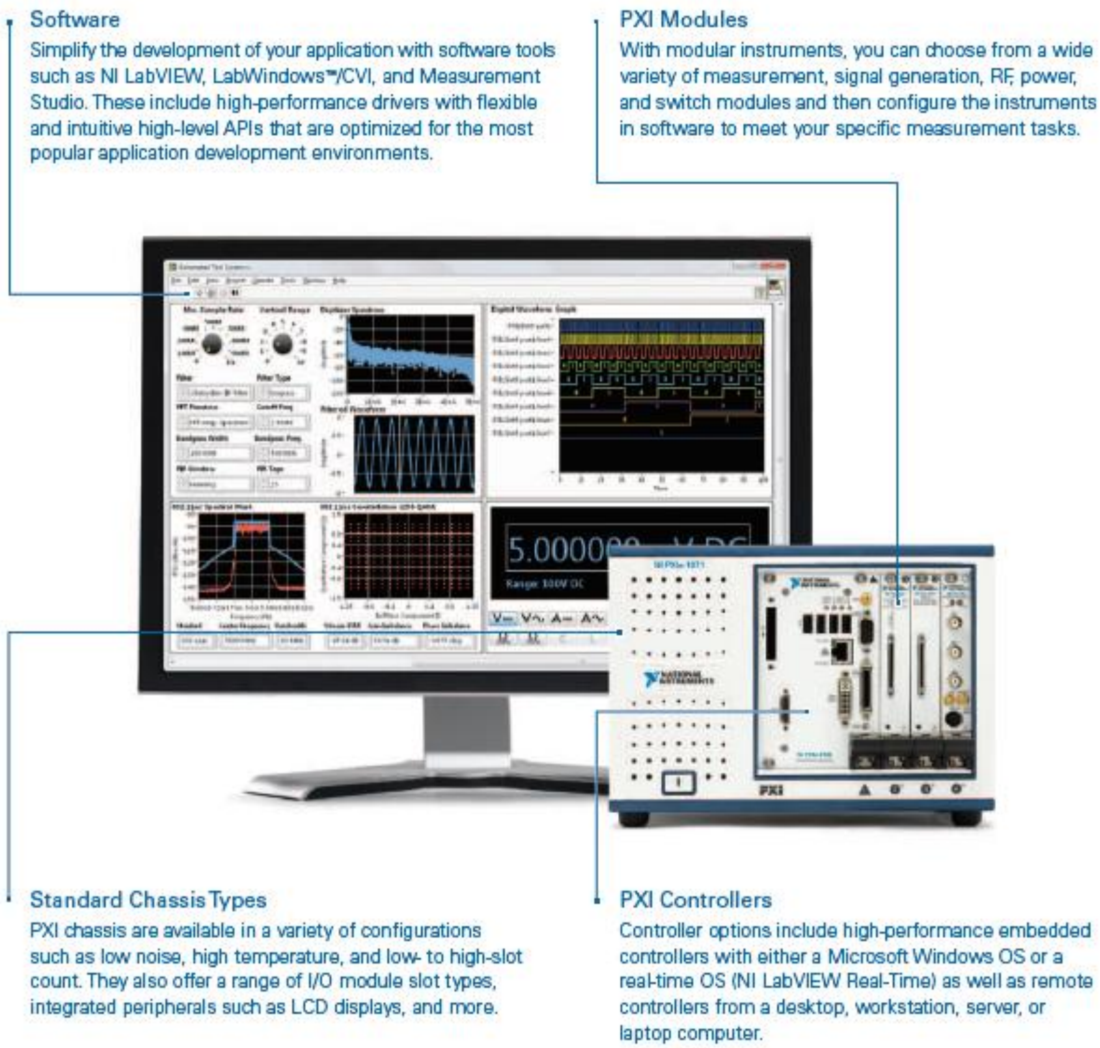

Figure 10. PXI Systems are Comprised of Three Basic Components – Chassis, System controller, and Peripheral Modules.

PXI Chassis

PXI chassis provides the rugged and modular packaging for the system. Chassis available in both 3U and 6U form factor, generally ranging in size from 4-slots to 18-slots, are available with special features such as DC power supplies and integrated signal conditioning. The chassis contains the high-performance PXI backplane, which includes the PCI bus and timing and triggering buses. Using these timing and triggering buses, users can develop systems for applications requiring precise synchronization.

PXI Controllers

As defined by the PXI Hardware Specification, all PXI chassis contain a system controller slot located in the leftmost slot of the chassis (slot 1). Controller options include remote controllers from a desktop, workstation, server, or a laptop computer and high-performance embedded

controllers with either a Microsoft OS (Windows 2000/XP) or a real-time OS (LabVIEW Real-Time).
PXI Embedded Controllers Embedded controllers eliminate the need for an external PC, therefore providing a complete system contained in the PXI chassis. PXI embedded controllers are typically built using standard PC components in a small, PXI package.

There are two types of PXI embedded controllers

- PXI Embedded Controllers with Windows
- PXI Embedded Real Time Controllers

PXI Embedded Controllers with Windows:

PXI embedded controllers with windows come with standard PC features such as integrated CPU, hard-drive, RAM, Ethernet, video, keyboard/mouse, serial, USB, and other peripherals, as well as Microsoft Windows and all device drivers already installed. Since the controllers have Microsoft Windows the user experience is no different than a PC or laptop computer. It has similar application software available in your PC or laptop computer such as Microsoft Office Word, Excel and PowerPoint.

PXI Embedded Real Time Controllers:

PXI Embedded Real Time Controllers also come with standard PC features along with a Real-Time OS such as LabVIEW Real-Time or VxWorks to deliver real-time, deterministic, and reliable I/O for measurement, automation, and control. Since RT Series PXI controllers are configured and programmed over the Ethernet, you can distribute a real-time application across the network and remotely monitor it. These controllers are designed for applications requiring deterministic and reliable performance and can be run under headless operation (i.e. no keyboard, mouse or monitor).

Figure 11: For example, the NI PXIe-8135 controller has the Intel Core i7-3610QE processor, with up to 16 GB of DDR3 RAM, a hard drive, and standard PC peripherals such as ExpressCard, USB 2.0, USB 3.0, Ethernet, serial, parallel and GPIB ports.

Embedded controllers are ideal for portable systems and contained "single-box" applications where the chassis is moved from one location to the next. For more information please refer to PXI controllers.

PXI Remote Controllers

There are two types of PXI remote controllers:

- Laptop control of PXI
- PC control of PXI

Laptop Control of PXI:

With ExpressCard MXI (Measurement eXtensions for Instrumentation) and PCMCIA CardBus interface kits, users can control PXI systems directly from laptop computers. During boot-up, the laptop computer will recognize all peripheral modules in the PXI system as PCI devices. Using ExpressCard MXI you can control your PXI system with a sustained throughput of up to 214 MB/s.

Figure 12: You can transparently control PXI Express systems from a laptop computer with either an ExpressCard

Users now have the advantage of mobile PXI systems for applications such as field tests, in-vehicle data logging, NVH and NDT with laptop control of PXI. You can purchase any ExpressCard MXI compatible laptop or PCMCIA CardBus compatible laptop to remotely control your PXI system.

PC Control of PXI:

With MXI-Express and MXI-4 interface kits, users can control PXI systems directly from desktop, workstation, or server computers. During boot-up, the computer will recognize all peripheral modules in the PXI system as PCI devices.

Using MXI-Express you can control your PXI system with a sustained throughput of up to 832 MB/s. With the MXI-Express 2-port interface kit, users will be able to control two PXI systems simultaneously from a single PC.

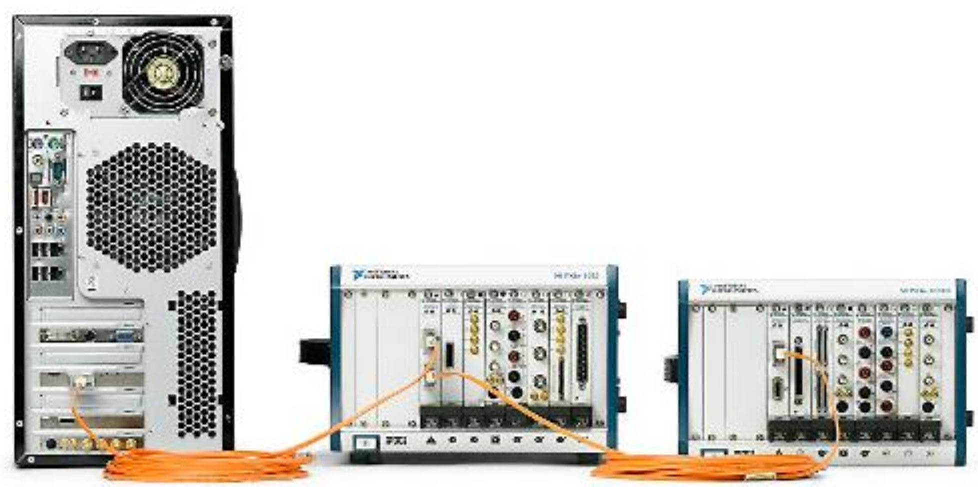

Figure 13: Control your PXI Express system directly from a PC with a transparent, high-speed link that requires no programming

With PXI remote controllers, you can maximize processor performance with minimized cost by using a desktop computer or laptop to remotely control a PXI system. Because all remote control products are software transparent, no additional programming is required.

PXI – Industry Standard for Modular Instrumentation

PXI, based on PCI and next-generation PCI Express, is the fastest-growing test and measurement standard since GPIB. PXI best meets modular instrumentation demands now and in the future, with more than 60 vendors in the PXI Systems Alliance, more than 1,500 products available today. Primarily, all instruments in a PXI system share the same power supply, chassis, and controller. Alternative approaches duplicate the power supply, chassis, and/or controller for every instrument, adding cost and size and decreasing reliability. With PXI, the controller can be a high-performance slot 0 embedded controller, desktop PC, laptop, or server-class machine. When you require faster processing, you can easily upgrade the controller of a PXI system. To reuse existing equipment, you can use PXI to control USB, GPIB, LAN/LXI, serial, and VXI instruments.

Modular instruments require a high-bandwidth, low-latency bus to connect instrument modules to the shared processor for performing user-defined measurements. PXI meets these needs with bandwidth up to 2 GB/s for each slot as specified by the PCI Express Gen 1 specification. Future generations of PCI Express will increase this further. Actual slot bandwidth per slot will vary by manufacturer depending on whether they use x4 or x8 PCI Express interfaces and whether they are Gen 1, 2, or 3.

Take a modular RF acquisition system for example. PXI can stream two channels of 100 MS/s, 16-bit IF data directly to a processor for computation. Neither LAN nor USB can meet these requirements, so these instruments always include an embedded, vendor-defined processor. Hence high-bandwidth standards such as PXI provide a truly software-defined approach required for modular instrumentation.

Why Customers Choose PXI?

Higher Throughput

Every application is unique and has very specific needs. However, bandwidth and latency are two important attributes of a platform for many applications. Latency tends to dominate single-point operations, such as digital multimeter/switch scanning, and bandwidth tends to dominate data streaming applications, such as waveform stimulus/response. PXI provides high speed for a wide range of applications with both high bandwidth and low latency via the PCI/PCI Express bus.

Timing and Synchronization

Many measurement and automation applications require advanced timing and synchronization capabilities that you cannot implement directly across PC standard I/O buses like PCI/PCI Express, Ethernet/LAN, USB, and so on. PXI offers advanced timing and synchronization features to meet your application needs:

- 100 MHz differential system reference clock
- 10 MHz reference clock signal
- Differential star trigger
- Star trigger bus with matched-length trigger traces to minimize intermodule delay and skew
- Trigger bus to send and receive high-speed timing and triggering signals
- Differential signals for multichassis synchronization
- Support for industry standards including GPS, 1588, and IRIG-B

System Reliability

The PXI specification defines requirements that make PXI systems well-suited for harsh environments. PXI features the high-performance IEC (International Electrotechnical Commission) connectors and rugged Eurocard packaging system used by CompactPCI. The PXI specification also defines specific cooling and environmental requirements to ensure operation in industrial environments. Modularity makes it easy to configure, reconfigure, and repair your PXI systems, resulting in very low mean time to repair (MTTR). Because PXI is modular, you can update individual modules and components without replacing the entire system.

Lower System Costs

Since PXI is a PC based platform it delivers the high-precision instrumentation, synchronization, and timing features at an affordable price. The low cost of PC components is only the beginning of the savings you gain from using PXI. With PXI, you use the same OS and application software like MS Excel, MS Word in your office and on the production floor. The familiarity of the software eliminates training costs and the need to retrain personnel every time you implement a new system. Because the foundation of PXI is PC technology, you benefit from low component costs, familiar software, and system reuse.