Introduction

PCI eXtensions for Instrumentation or PXI is a computer based hardware and software platform for test and measurement systems. Developed in the late 1990's as an open industry standard based on the compact PCI (cPCI) computer bus, PXI provides a basis for complex, rugged, modular instrumentation systems. The PXI standard is governed by the PXI systems Alliance (PXISA) which maintains and controls the evolution of the standard to insure interoperability of instruments from hundreds of vendors.

PXI Express or PXIe is a subset of the PXI standard that replaces PXI's parallel data bus with a high speed serial interface. PXIe provides the most advantages for modular instruments like digitizers or arbitrary waveform generators which often need to transfer large amounts of data. For example, state-of-the-art PXIe products from Spectrum incorporate a fast interface that can stream at rates of up to 1.7 GB/s.

A Little History

The Peripheral component Interconnect (PCI) bus was originally designed to be a commercial computer bus. In the late 1990s the CompactPCI (cPCI) bus was created by the PCI Industrial Computers Manufacturers Group (PICMG) by taking the desktop PCI specification and developing an industrial Eurocard-based module for it. This enabled the benefit of the PCI bus while providing a rugged form factor that is suitable for industrial testing applications. The cPCI physical design utilizes a chassis with a passive backplane that includes power supplies and cooling. It features front loading cards with top and bottom support rails that provide extra stability.

While instruments can make use of the cPCI bus they may also require additional functionality in the form of timing, trigger, and synchronization signals. To achieve this PXI joined industrial PC technology from the cPCI specification with integrated synchronization and triggering. The result is a rugged, modular platform with a quantum improvement over older bus structures that has enabled PXI to become the industry standard for measurement and automation applications.

The modular design brings a number of advantages. Firstly, it makes installing PXI or PXIe based instruments easy. In addition, front panel access allows signals to be connected to the modules in an easier way than is the case with traditional computer add-in cards. PXI systems also have cooling fans mounted at the base of their chassis to blow air vertically over the PXI cards, which gives improved temperature ratings. Common to both the PXI and cPCI buses is the use of standardized 3U or 6U card sizes that are typically interconnected via a passive backplane.

PXI and PXIe Hardware

The hardware implementation of PXI/PXIe consists of a chassis, controller or computer interface, and peripheral cards.

Chassis

Cards are housed in a chassis which includes a passive backplane, power sources, cooling and provision for a controller. Hybrid chassis use a passive motherboard which allows them to accept cPCI, PXI or PXIe peripheral cards. Figure 1 shows a typical PXIe hybrid chassis. This eighteen slot chassis is a National Instrument PXIe-1075. Icons at the bottom of the chassis front panel indicate the functions of each of the slots. The left most slot (slot 1) is reserved

for the system controller, if used, or remote PXIe controller. Slot 10 is a system timing slot which can be used by a PXIe system timing module or a PXIe peripheral module. Slots 2 through 5 and 15 through 18 are hybrid card slots that can handle hybrid compatible PXI, PXIe or 32 bit cPCI peripheral modules. Slots 6 through 9 and 11 through 14 are dedicated PXIe slots.

The main difference between PXI and cPCI are the addition of timing and synchronization features. There are eight shared trigger bus lines, a differential star trigger (featuring board to board distribution with low timing skew) and a 10MHz system reference clock that can be used for timing synchronization of multiple cards.

The 10 MHz reference clock is strictly defined in the PXI standard to assure a low skew distribution (< 1ns) to all peripherals. The clock can be used for timing synchronization or as a stable edge reference for

Figure 1: standard 18 slot PXIe chassis from National Instruments

qualifying trigger events. Most PXI chassis make provision for an external 10 MHz clock input which the internal clock source can phase lock to.

A PXIe chassis also offers a 100 MHz high frequency differential clock called PXIe_CLK100 which is supplied to the system timing and hybrid peripheral slots. The skew for this clock is less than 100ps. If used, the PXIe_CLK100 lines must be terminated on the peripheral. There is also a differential synchronization signal, PXIe_SYNC100. This signal is synchronous to PXIe_CLK100 and asserts one out of every 10 clocks to indicate the phase relationship of the 10 MHz and 100 MHz reference clocks.

The PXI standard defines a trigger bus consisting of eight separate lines. They allow trigger, synchronization, and timing signals to be passed from one module to another module where one module may act as a master passing timing or synchronization to the others that may act as slaves. It's important to note that the trigger bus has limited bandwidth and it is not recommended for clock frequencies higher than 20 MHz.

The star trigger path is used for applications where a high speed trigger with short delay and minimum skew are required. The star trigger is routed from the star trigger slot (slot 2 in the PXI chassis) using a star topology to each of the remaining cards slots. The line lengths are designed to ensure that the propagation delays are matched to within 1 ps.

Another PXI bus is the local bus. This bus consists of a 13 line daisy chained bus from one slot to the adjacent slots. The bus allows both digital and analog signals to be passed from card to card. This bus supports signal levels to 42 Volts.

The PXIe chassis also must supply power to each slot. Voltages of +3.3, +5, +12 and -12 are available at all slots with current limits appropriate for the slot function.

Controller
The PXIe chassis includes provision for a controller. The controller must be located in the leftmost slot (slot1). The controller can be an embedded computer or a link to an external computer or laptop.

Embedded controllers are dedicated computers with all required processors, RAM, hard disk drive, and all related peripherals to assure that the PXI system can run independently. The Embedded controllers most often run Windows as an operating system although other operating systems can be employed. As in all computer systems appropriate driver software must be installed.
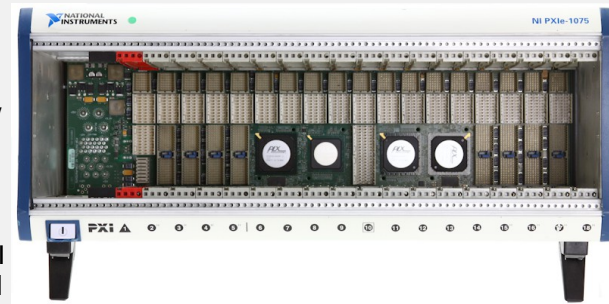
Peripheral Cards and Modules

PXIe systems can accept a large variety of PXI, PXIe, or cPCI cards in appropriate chassis slots. The standard specifies two different size modules 3U and 6U. The 3U module uses a card with dimensions of 100 by 160 mm (3.94 by 6.3 inches). The 6U card is 233.35 by 160mm (9.19 by 6.3 inches). The smaller 3U module is the most commonly used configuration. Instruments needing more space are accommodated by stacking multiple 3U cards resulting in modules of greater than one slot width.

Figure 2: outline and dimension of a 3U PXIe module

Both card types are fitted with locking mechanical ejectors (a single one in the 3U modules and dual ejectors for the larger 6U modules). To further improve the mechanical robustness 3U modules also include retaining screws to lock the module in place.

Generally, the 3U modules are populated with two connectors depending on the family (PXI, PXIe, or cPCI).

Figure 3 shows a high-speed, high-resolution Spectrum M4x.4451-x4 PXIe digitizer. This digitizer uses a 3U card configuration with a 2 slot width

The M4x.44xx-x4 series digitizers offer both high speed and high resolution. The series includes PXIe cards with either two or four synchronous channels. Each channel has its own dedicated ADC which can sample at rates from 130 MS/s up to 500 MS/s and are available with either 14 bit or 16 bit resolution as summarized in Table 1.
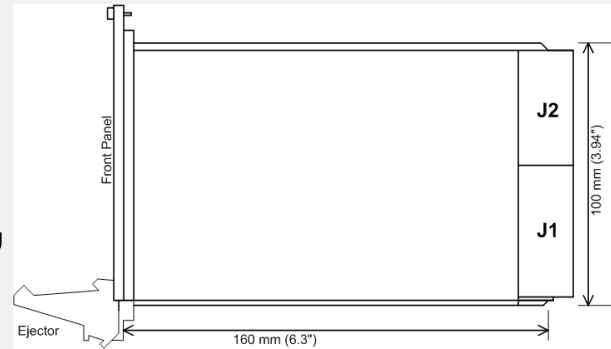
Model Number	Number of Channels	ADC Resolution	Acquisition Memory	Maximum Sample Rate
M4x.4410-x4	2	16	2 GSamples	130 MS/s
M4x.4411-x4	4	16	2 GSamples	130 MS/s
M4x.4420-x4	2	16	2 GSamples	250 MS/s
M4x.4421-x4	4	16	2 GSamples	250 MS/s
M4x.4450-x4	2	14	2 GSamples	500 MS/s
M4x.4451-x4	4	14	2 GSamples	500 MS/s

Table 1: Summary of M4x.44xx-x4 digitizer product specifications

▶ An Introduction to PXIe

The M4x series PXI Express cards use a PCI Express x4 Gen 2 connection. They can be used in every PXI Express (PXIe) slot, as well as in any PXI hybrid slot with Gen 1, Gen 2 or Gen 3. The PCI Express x4 Gen 2 interface offers outstanding data streaming performance. The interface and Spectrum's optimized drivers enable data transfer rates in excess of 1.7 GB/s so that signals can be acquired, stored and analyzed at the highest speeds.

The M4x.44xx series digitizers use two backplane connectors. The topmost connector handles the PXIe interface which allows the digitizer to stream data at rates up to 1.7 GB/s. The lower connector contains the PXI specific signals like the local and trigger buses and the timing reference signals. Spectrum digitizers currently can use the PXI trigger bus or the Star trigger. They do not currently support Local bus connections.

Figure 3: Spectrum M4x.4451-x4, 4 channel 500 MS/s 14 bit digitizer with PXIe interface

The front panel contains the connectors for the input channels, trigger, clock, and digital I/O signals. The module ejector is located at the bottom of the front panel. Because this is a two slot wide peripheral there are two retaining screws on the top of the front panel.

Software

The PXI/PXIe standard also stipulates the software as well as the hardware. In general, it specifies the number of Windows operating systems that must be supported. Other operating systems such as Linux can also be used. The key factors in the selection of an operating system is that it can connect to the PXI bus and that there are suitable drivers available.

Modules can be controlled in either of two ways in the PXI system, this can be message or register based. Register based communications is the most common method. In register based control the controller accesses the registers within the modules which control their actions. The message based communication, as the name implies, controls and reads data from the peripherals by exchanging text based messages. Messages have a well-defined format and range of arguments. Message based control is more typical of standalone instruments but there are some PXI modules which use this method.

Instrument drivers simplify the communication and control of PXI/PXIe peripherals. An instrument driver is a type of computer software that allows interaction with hardware devices. It constitutes an interface for communications between the operating system and application specific software. It allows the transmission of commands and/or the reception of data. The main reason for using a driver is that it frees users from worrying about the low-level details of device control.

The PXI/PXIe standard relies on VISA drivers. VISA is the Virtual Instrument Software Architecture, a standard managed by the IVI Foundation (http://www.ivifoundation.org). VISA defines a software layer that allows users to control instruments without needing to understand the workings of the interface bus being used. It provides a way of writing instrument drivers using a consistent approach. This means that, even when dealing with instruments from different manufacturers in the same chassis, interoperability becomes far easier.

▶ An Introduction to PXIe

IVI, or Interchangeable Virtual Instrument is a driver technology that provides a standard application programming interface (API) for communications with instruments. As it builds on top of VISA this allows users to potentially interchange instruments without changing their source code. The IVI Foundation also manages the IVI specification.

Instrument drivers are essential if users wish to write their own programs with commonly used programming languages. In the case of the Spectrum M4x.44xx-x4 series digitizers both Windows and Linux operating systems are supported (32 bit and 64 bit versions). Programs can be created in a variety of languages including Visual C++, Borland C++ Builder, LabWindows/CVI, LabVIEW, Borland Delphi, Visual Basic, VB.NET, C#, J#, Python and IVI. Support for Linux per-compiled kernel modules are included for the most common Linux distributions like RedHat, Fedora, Suse, Ubuntu LTS or Debian.

Controlling a multi-peripheral system requires system integration software such as LabVIEW, LabWindows/CVI, or MATLAB. These programs, with suitable drivers, offer support for multiple modular instruments operating as a system. They are capable of controlling such a system as well as gathering, measuring, and analyzing measurement data.

LabVIEW Example

LabVIEW, from National Instruments, is the most common graphical programming language for measurement applications and is very well supported by digitizer hardware by means of a dedicated LabVIEW driver. The driver combines different digitizer functions into functional blocks and makes them available within LabVIEW. The LabVIEW driver package consists of several different libraries and some open example virtual instruments (VI's) showing the use of the driver. Besides these libraries all driver functions can also be directly called.

Figure 4 shows an example that gives you the chance to use the analog channels of the digitizer card as a simple oscilloscope. It shows the

Figure 4: LabVIEW oscilloscope example for all Spectrum Digitizers.

virtual front panel of an oscilloscope that allows single acquisition or multiple acquisitions of data on up to four channels.
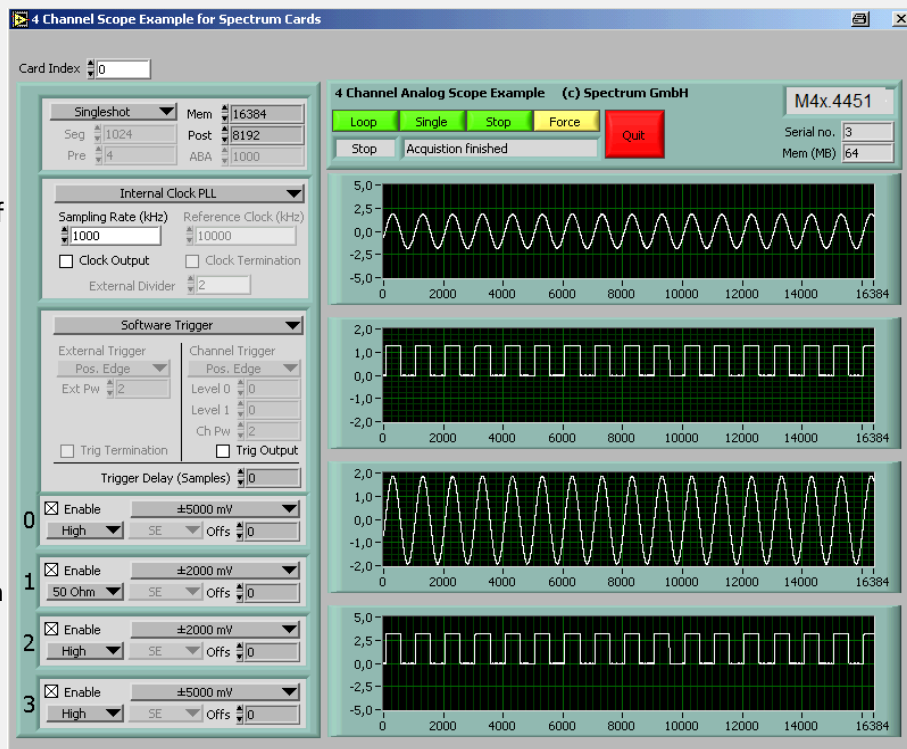
All clock settings, trigger modes, sources and all the input channel settings can be setup and changed via the interface. The example works for all acquisition cards with up to 4 channels, independent of the analog resolution and the maximum sampling speed of the digitizer.

All examples are delivered with open and editable diagrams allowing the programmer to use

it as a base for their own programming or as an example to better understand how the LabVIEW driver works.

The user interface was created with the intent to allow a fast start up with basic functions and provide a useful tool for LabVIEW users using a digitizer for the first time.

Device Specific Software

Peripheral instrument suppliers also offer instrument specific control and analysis software for their modules. Spectrum offers SBench 6, an easy to use graphical operating software for Spectrum modules such as digitizers and arbitrary waveform generators. SBench 6 is a valuable tool for checking the peripheral cards performance and assisting with the unit's initial setup. A base version of SBench 6 is supplied with the unit making it is possible to test the card, display acquired data and make some basic

Figure 5: SBench 6 Acquisition of an ultrasound signal using M4x.4450 2 channel 500 MS/s digitizer

measurements. The professional version of SBench 6 adds advanced measurement functions, such as filtering, averaging, FFTs and X/Y display, archiving functions like import and export utilities as well as support for all acquisition modes including data streaming.

Figure 5 provides an example of the SBench6 user interface and a typical measurement. In this example an ultrasound pulse has been captured and digitized. A number of measurements, including maximum value, frequency, minimum value, and peak to peak have been made on the acquired pulse. A Fast Fourier Transform (FFT) has also been performed.
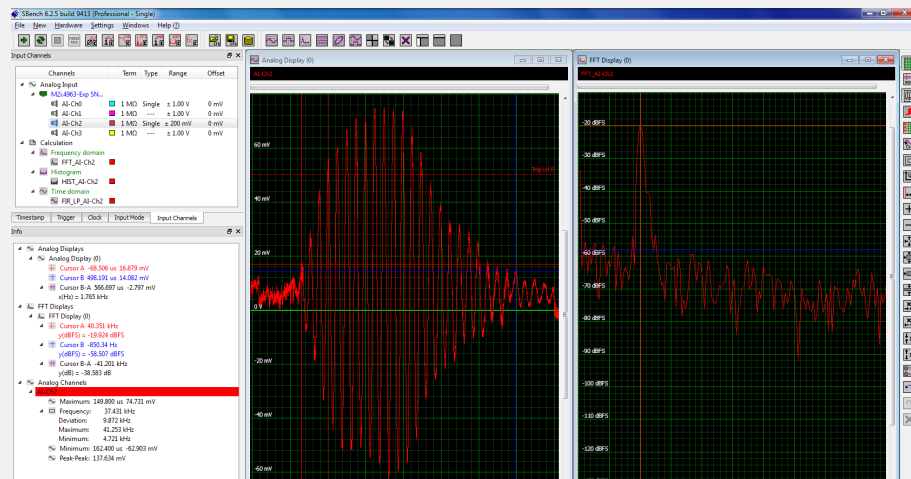
Conclusion

Modular PXIe instruments like the M4x.4451-x4 digitizer are rugged instruments meant for industrial environments. The PXI standard guarantees inter-operability with other PXI, PXIe, or cPCI instruments and peripheral modules enabling the creation of multi-module test systems. Modules communicate via high speed internal connections for concise triggering and complex interactions. With their high speed communication, fast data transfer and access to powerful processing technology PXIe systems can be used to greatly improve measurement speed and productivity in most automated testing environments. Furthermore, standardized instrument drivers and software tools make it possible for rapid system configuration and easy customization.