PXI™ Specification

PCI eXtensions for Instrumentation

An Implementation of CompactPCI™

Revision 1.0
August 20, 1997

NATIONAL INSTRUMENTS™
The Software is the Instrument®

IMPORTANT INFORMATION

Copyright

NOTICE

The *PXI Specification* is authored and copyrighted by National Instruments Corporation. The intent of National Instruments is for the *PXI Specification* to be an open industry standard supported by a wide variety of vendors and products. Vendors and users who are interested in developing PXI-compatible products or services, as well as parties who are interested in working with National Instruments to further promote PXI as an open industry standard are invited to contact National Instruments for further information.

National Instruments wants to receive your comments on this specification. Write to National Instruments, Attn: PXI Specification Committee, 6504 Bridge Point Parkway, Austin, TX 78730.

National Instruments makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. National Instruments shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Trademarks

LabVIEW™ and CVI™ are trademarks of National Instruments Corporation.

Product and company names are trademarks or trade names of their respective companies.

PXI Specification Revision History

This section is an overview of the revision history of the PXI specification. Specific individual additions/modifications to the document between revisions are denoted with diff marks, "|", in the right column of a line of text for which the change applies.

Revision 1.0, August 20, 1997

This is the first public revision of the PXI specification.

Contents

Figures

Tables

1. Introduction

This section describes the primary objectives and scope of the PCI eXtensions for Instrumentation (PXI) specification. It also defines the intended audience and lists relevant terminology and documents.

1.1. Objectives

PXI was created in response to the needs of a variety of instrumentation and automation users who require ever increasing performance, functionality, and reliability from compact rugged systems that are easy to integrate and use. Existing industry standards are leveraged by PXI to benefit from high component availability at lower costs. Most importantly, by maintaining software compatibility with industry-standard personal computers, PXI allows industrial customers to use the same software tools and environments with which they are familiar.

PXI leverages the electrical features defined by the widely adopted Peripheral Component Interconnect (PCI) specification. It also leverages the CompactPCI form factor, which combines the PCI electrical specification with rugged Eurocard mechanical packaging and high-performance connectors. This combination allows CompactPCI and PXI systems to have up to seven peripheral slots versus four in a desktop PCI system. Systems with more expansion slots can be built by using multiple bus segments with industry-standard PCI-PCI bridges. For example, a 13-slot PXI system can be built using a single PCI-PCI bridge. The PXI specification adds electrical features that meet the high-performance requirements of instrumentation applications by providing triggering, local buses, and system clock capabilities. PXI also offers two-way interoperability with CompactPCI products.

By implementing desktop PCI in a rugged form factor, PXI systems can leverage the large base of existing industry-standard software. Desktop PC users have access to different levels of software, from operating systems to low-level device drivers to high-level instrument drivers to complete graphical APIs. All of these software levels can be used in PXI systems. PXI defines software frameworks (for Microsoft Windows NT and Windows 95) for complete systems and requires appropriate device driver software for all PXI peripheral modules to ease system integration. Furthermore, PXI implements the Virtual Instrument Software Architecture (VISA). VISA is used to locate and communicate with serial, VXI, and GPIB peripheral modules. PXI extends VISA beyond these interfaces to allow for the location and control of PXI peripheral modules. This extension preserves the model for instrumentation software that has been adopted by the instrumentation community. The result is a very powerful software commonality that spans PXI, CompactPCI, desktop PCI, VXI, GPIB, and other instrumentation architectures.

The diagram in Figure 1.1 summarizes the scope of the PXI specification by depicting its mechanical, electrical, and software architectures.

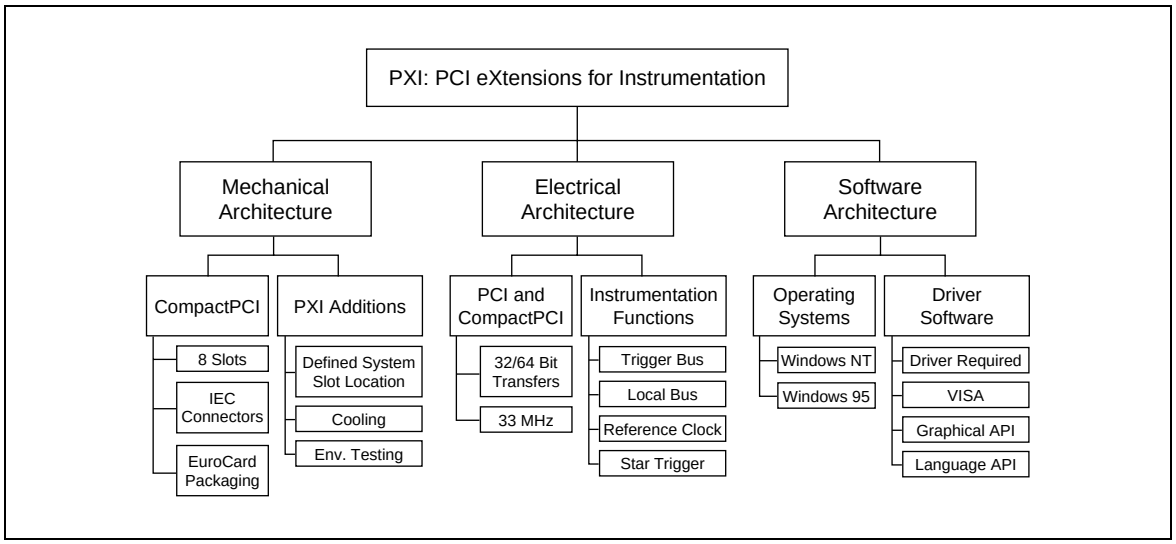

Figure 1.1 The PXI Architecture

1.2. Intended Audience and Scope

This specification is organized with a top-down approach whereby general descriptions precede the more detailed specifications found deeper in the subsections. This structure is intended to serve the needs of a variety of audiences from product developers to system integrators to end-users. Product developers may want to become familiar with all portions of this specification while end users may be interested in only the feature set description and perhaps the summaries of how these features are implemented. The goal of this specification is to serve as the central source of information relevant to all users and providers of PXI compatible systems.

The first section of this specification describes the features that PXI systems can offer and how these features can be applied to instrumentation. The subsequent sections cover the mechanical, electrical, and software requirements specific to implementing PXI features.

1.3. Background and Terminology

This section defines the acronyms and key words that are referred to throughout this specification. This specification uses the following acronyms:

API—Application Programming Interface

Eurocard—European Packaging Specifications (IEC 60297, IEEE 1101.1, IEEE 1101.10, IEEE 1101.11)

GPIB—General Purpose Interface Bus, IEEE 488

ISA—Industry Standard Architecture; desktop PC adapter board specification

PCI—Peripheral Component Interconnect; electrical specification defined by PCISIG

PCISIG—PCI Special Interest Group

PICMG—PCI Industrial Computer Manufacturers Group

PXI—PCI eXtensions for Instrumentation

VISA—Virtual Instrument Software Architecture

VITA—VMEbus International Trade Association

VME—Versa Module Europe; VMEbus specification governed by the VSO

VPP—VXI*plug&play* Specification

VSO—VITA Standards Organization

VXI—VME Extensions for Instrumentation

This specification uses several key words, which are defined as follows:

RULE: Rules SHALL be followed to ensure compatibility. A rule is characterized by the use of the words SHALL and SHALL NOT.

RECOMMENDATION: Recommendations consist of advice to implementers that will affect the usability of the final module. A recommendation is characterized by the use of the words SHOULD and SHOULD NOT.

PERMISSION: Permissions clarify the areas of the specification that are not specifically prohibited. Permissions reassure the reader that a certain approach is acceptable and will cause no problems. A permission is characterized by the use of the word MAY.

OBSERVATION: Observations spell out implications of rules and bring attention to things that might otherwise be overlooked. They also give the rationale behind certain rules, so that the reader understands why the rule must be followed.

MAY: A key word indicating flexibility of choice with no implied preference. This word is usually associated with a permission.

SHALL: A key word indicating a mandatory requirement. Designers SHALL implement such mandatory requirements to ensure interchangeability and to claim conformance with the specification. This word is usually associated with a rule.

SHOULD: A key word indicating flexibility of choice with a strongly preferred implementation. This word is usually associated with a recommendation.

1.4. Applicable Documents

This specification defines extensions to the base PCI and CompactPCI specifications referenced in this section. It is assumed that the reader has a thorough understanding of PCI and CompactPCI. The CompactPCI specification refers to several other applicable documents with which the reader may wish to become familiar. This specification refers to the following documents directly:

- *PCI Local Bus Specification*, Rev. 2.1
- *PICMG 2.0 R2.1 CompactPCI Specification*

- VXI*plug&play* Specifications (VPP-3.*x* and VPP-7)
- *Serialized IRQ Support for PCI Systems Specification*, Rev. 6.0, September 1, 1995, Compaq Computer et al.
- IEC 61326-1:1997, *Electrical equipment for measurement, control, and laboratory use—EMC requirements—Part I, General requirements*, International Electrotechnical Commission
- IEC 1010-1:1990 + A1:1992, *Safety requirements for electrical equipment for measurement, control, and laboratory use—Part 1, General requirements*, International Electrotechnical Commission
- IEC 60068-1, *Environmental testing*, International Electrotechnical Commission
- *Hardware Design Guide for Microsoft Windows 95*, Microsoft Corporation
- *PC 97 Hardware Design Guide*, Microsoft Corporation

2. PXI Architecture Overview

This section presents an overview of a PXI system's features and capabilities by summarizing the mechanical, electrical and software architectures defined by this specification.

2.1. Mechanical Architecture Overview

PXI supports the two form factors that are depicted in Figure 2.1. The 3U form factor defines modules that are 100 by 160 mm (3.94 by 6.3 in.) and have two interface connectors. J1 carries the signals required for the 32-bit PCI local bus and J2 carries the signals for 64-bit PCI transfers and the signals for implementing PXI electrical features. The 6U form factor defines modules that are 233.35 by 160 mm (9.19 by 6.3 in.) and may carry up to three additional connectors for future expansion of the PXI specification.

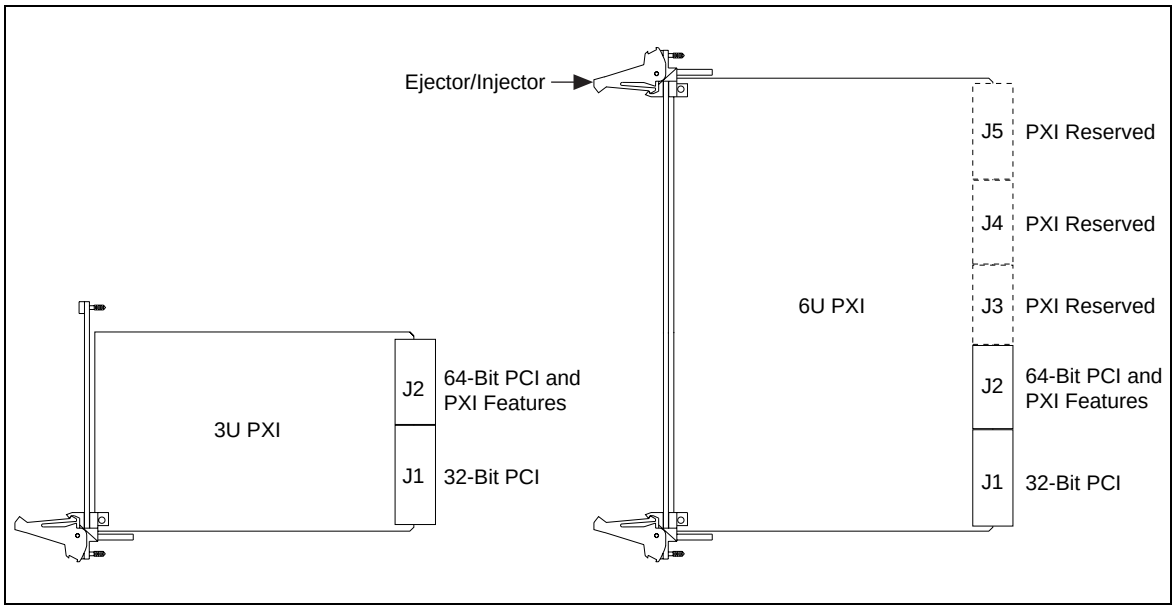

Figure 2.1 PXI Peripheral Module Form Factors and Connectors

Figure 2.2 presents an example of a typical PXI system to help illustrate the following keywords (in *italics*). A PXI system is composed of a *chassis* that supports the PXI *backplane* and provides the means for supporting the *system controller module* and the *peripheral modules*. The chassis must have one *system slot* and may have one or more *peripheral slots*. Any number of *controller expansion slots* may be available to the left of the system slot. The optional *star trigger controller*, when used, must reside next to the system controller module. If a star trigger controller is not used in a system, a peripheral module can be installed in the slot next to the system controller module. The backplane carries the *interface connectors* (P1, P2, ...) and provides the interconnection between the controller and peripheral modules. A maximum of

seven peripheral modules can be used in a single PXI *bus segment*. PCI-PCI bridges can be used to add bus segments for additional expansion slots.

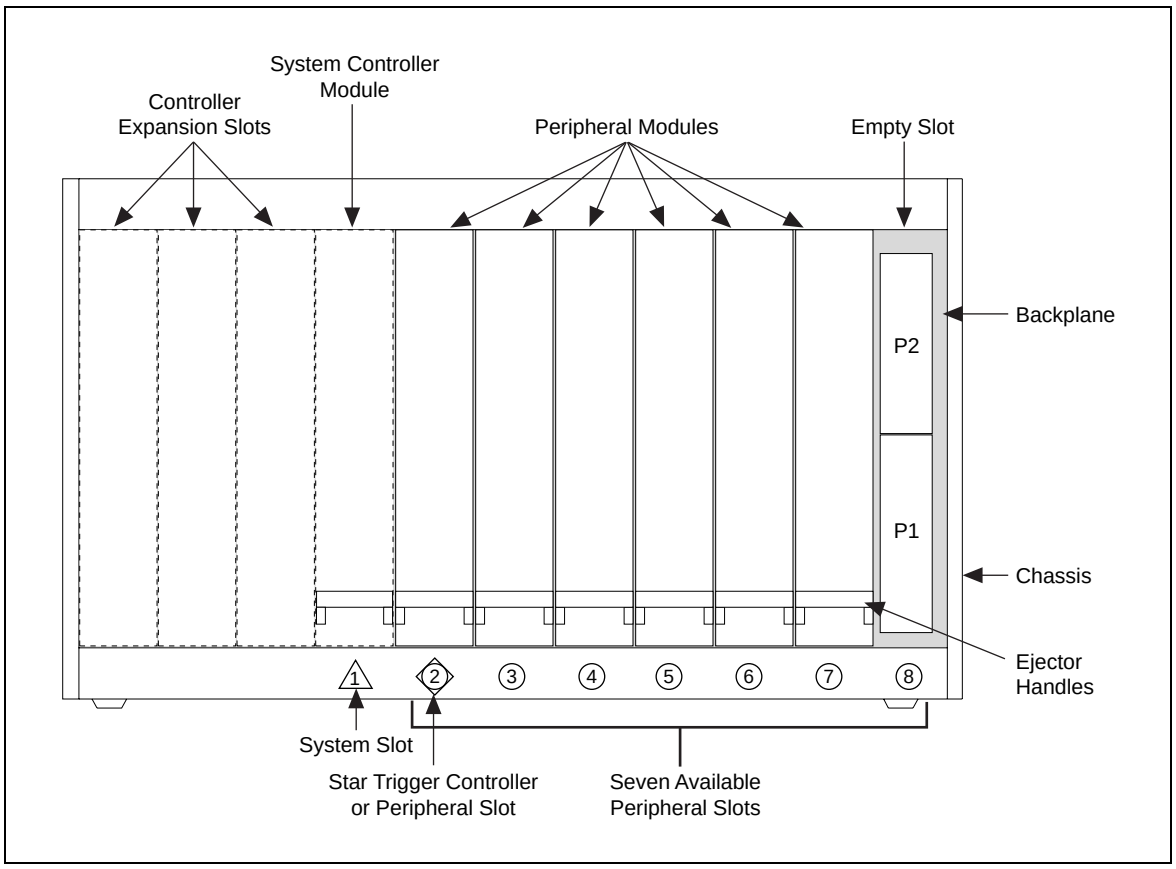

Figure 2.2 Example of a 3U PXI System (Single Bus Segment)

2.1.1. System Slot Location

PXI defines the system slot location to be on the left end of the PCI bus segment. This defined arrangement is a subset of the numerous possible configurations allowed by CompactPCI (a CompactPCI system slot may be located anywhere on a backplane). Defining a single location for the system slot simplifies integration and increases the degree of compatibility between PXI controllers and chassis. Furthermore, the PXI specification requires that if necessary the system controller module should expand to the left into what are defined as controller expansion slots. Expanding to the left prevents system controllers from using up valuable peripheral slots.

PXI also defines the location of a star trigger controller slot that can accept a peripheral module or a special star trigger module that can provide individual triggers to all other peripheral modules. The star trigger signals are routed from the star trigger slot to each peripheral slot on all PXI backplanes.

2.1.2. Additional Mechanical Features

All mechanical specifications defined in PICMG 2.0 R2.1 (CompactPCI Specification) apply directly to PXI systems; however, PXI includes additional requirements that simplify system integration. Forced cooling in PXI chassis is required and the airflow direction is defined. Environmental testing for operating and storage temperature ratings of all PXI products is required and must be clearly documented by suppliers.

2.1.3. Interoperability with CompactPCI

Interoperability among PXI compatible products and standard CompactPCI products is a very important feature provided by this specification. Many PXI-compatible systems will require components that do not implement PXI-specific features. For example, a user may want to use a standard CompactPCI network interface module in a PXI chassis. Likewise, some users may choose to use a PXI compatible module in a standard CompactPCI chassis. In these cases, the user will not be able to use PXI-specific functions but will still be able to use the basic functions of the module. Note that interoperability between PXI-compatible products that use the J2 connector for PXI-defined signals and other application-specific implementations of CompactPCI chassis (which may define other signal definitions for sub-buses on the P2 backplane connector) is not guaranteed. Of course, both CompactPCI and PXI leverage the PCI local bus, thereby ensuring software and electrical compatibility as depicted in Figure 2.3.

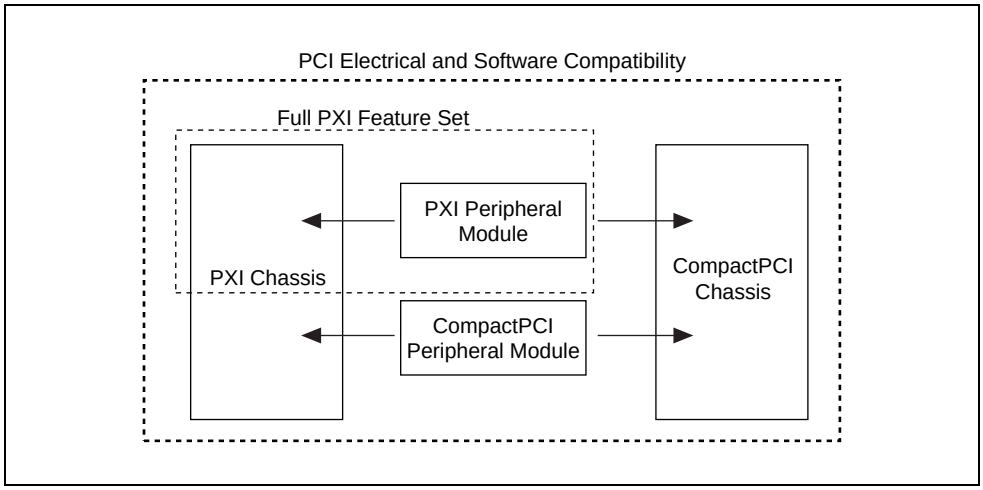

Figure 2.3 PXI and CompactPCI Interoperability

2.2. Electrical Architecture Overview

Many instrumentation applications require system timing capabilities that cannot be directly implemented across standard desktop computer backplanes such as ISA, PCI, or PCMCIA. PXI uses the standard PCI bus and adds specific signals for instrumentation including bused trigger lines, slot-specific triggers, a dedicated

system clock, and slot-to-slot local buses to address the need for advanced timing, synchronization, and side-band communication.

2.2.1. Peripheral Component Interconnect (PCI) Features

PXI offers the same performance features defined by the desktop PCI specification, with one notable exception. A PXI system can have up to eight slots per segment (one system slot and seven peripheral slots), whereas a desktop PCI system can have only five per segment (one motherboard or system slot and four peripheral slots). The capability to have three additional peripheral slots is defined in the CompactPCI specification upon which PXI draws. Otherwise, all the features of PCI transfer into PXI:

- 33 MHz performance
- 32- and 64-bit data transfers
- 132 Mbytes/s (32-bit) and 264 Mbytes/s (64-bit) peak data rates
- System expansion via PCI-PCI bridges
- 3.3 V migration
- Plug and Play capability

2.2.2. Local Bus

The PXI local bus is a daisy-chained bus that connects each peripheral slot with its adjacent peripheral slots to the left and right. Thus, the right local bus of a given peripheral slot connects to the left local bus of the adjacent slot, and so on. Each local bus is 13 lines wide and can be used to pass analog signals between modules or to provide a high-speed side-band digital communication path that does not affect the PXI bandwidth.

Local bus signals can range from high-speed TTL signals to analog signals as high as 42 V. Keying of adjacent modules is implemented by initialization software that prohibits the use of incompatible modules. This software uses the configuration information specific to each peripheral module to evaluate compatibility before enabling local bus circuitry. This method provides a flexible means for defining local bus functionality that is not limited by hardware keying.

The local bus lines for the leftmost peripheral slot of a PXI backplane are used for the star trigger. Figure 2.4 schematically shows a complete PXI system demonstrating the local buses.

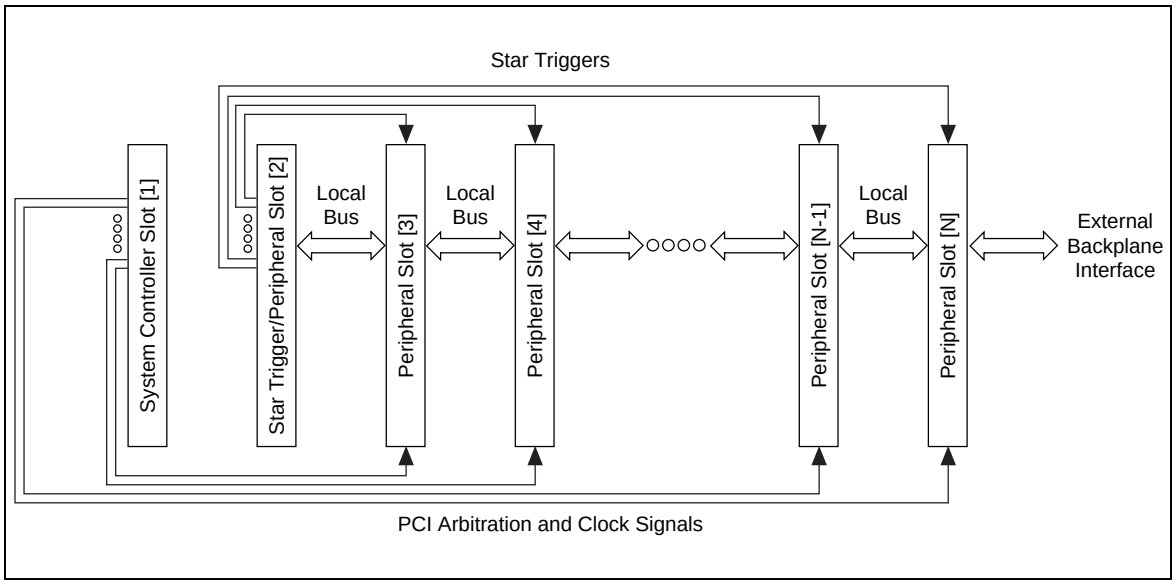

Figure 2.4 PXI Local Bus Routing

2.2.3. System Reference Clock

The PXI 10 MHz system clock (PXI_CLK10) is distributed to all peripheral modules in a system. This common reference clock can be used for synchronization of multiple modules in a measurement or control system. The PXI backplane specification defines implementation guidelines for PXI_CLK10. As a result, the low skew qualities afforded by this reference clock make it ideal for precise multimodule synchronization by using trigger bus protocols to qualify individual clock edges.

2.2.4. Trigger Bus

The eight PXI bused trigger lines are highly flexible and can be used in a variety of ways. For example, triggers can be used to synchronize the operation of several different PXI peripheral modules. In other applications, one module can control carefully timed sequences of operations performed on other modules in the system. Triggers may be passed from one module to another, allowing precisely timed responses to asynchronous external events that are being monitored or controlled. The number of triggers that a particular application requires varies with the complexity and number of events involved.

2.2.5. Star Trigger

The PXI star trigger bus offers ultra-high performance synchronization features to users of PXI systems. The star trigger bus implements a dedicated trigger line between the first peripheral slot (adjacent to the system slot) and the other peripheral slots. A star trigger controller can be installed in this slot and can be used to provide very precise trigger signals to other peripheral modules. Systems that do not require this advanced trigger can install any standard peripheral

module in this slot. Through the required use of line-length equalization techniques for routing the star triggers, PXI systems can meet demanding triggering requirements for which bused triggers are not appropriate. Note that the star trigger can be used to communicate information back to the star trigger controller, as in the case of reporting a slot's status, as well as responding to information provided by the star trigger controller.

This trigger architecture for PXI gives two unique advantages in augmenting the bused trigger lines. The first advantage is a guarantee of a unique trigger line for each module in the system. For large systems, this eliminates the need to combine multiple module functions on a single trigger line or to artificially limit the number of trigger times available. The second advantage is the low-skew connection from a single trigger point. The PXI backplane defines specific layout requirements such that the star trigger lines provide matched propagation time from the star trigger slot to each module for very precise trigger relationships between each module.

2.2.6. System Expansion with PCI-PCI Bridge Technology

A PXI system can be built with more than one bus segment by using standard PCI-PCI bridge technology. The bridge device takes up one PCI load on each of the bus segments that it links together. Thus, a system with two bus segments offers 13 expansion slots for PXI peripheral modules.

(2 bus segments) x (8 slots per segment) - (1 system controller slot)
- (2 slots for PCI-PCI Bridge) = 13 available expansion slots

Similarly, a three-bus segment system would offer 19 expansion slots for PXI peripheral modules.

The trigger architecture defined by PXI has implications for systems with multiple bus segments. The PXI trigger bus provides connectivity within a single bus segment and does not allow physical connection to an adjacent bus segment. This maintains the high performance characteristics of the trigger bus and allows multisegment systems to partition instruments into logical groups. Multiple segments may be logically linked by providing buffers between physical segments. The star trigger provides the means to independently access all 13 peripheral slots in a two-segment system for applications in which a high number of instruments require synchronization and controlled timing.

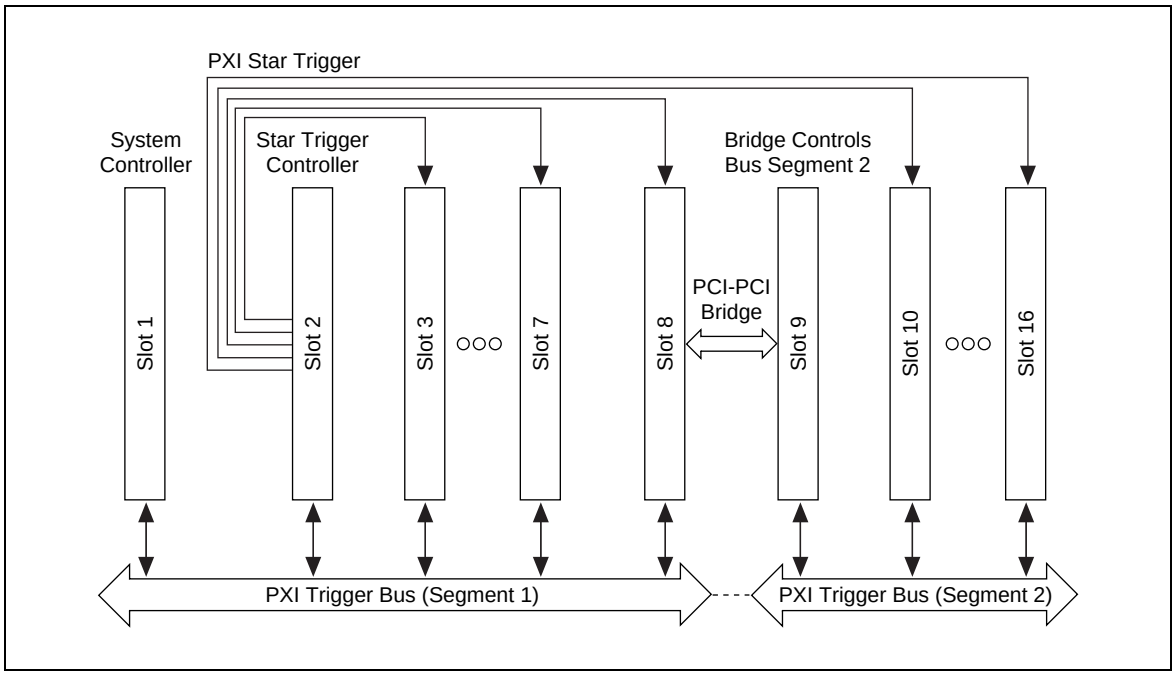

Figure 2.5 PXI Trigger Architecture for Two Bus Segments

2.3. Software Architecture Overview

Like other bus architectures, PXI defines standards that allow products from multiple vendors to work together at the hardware interface level. Unlike many other specifications, however, PXI defines software requirements in addition to bus-level electrical requirements, to further ease integration. These requirements include the support of standard operating system frameworks such as Windows NT and Windows 95 (WIN32) and the support of instrumentation software standards developed by the VXI*plug&play* Systems Alliance (VPP and VISA). Appropriate drivers for all peripheral modules are also required. Clearly, the PXI software specification is motivated by the benefits achieved through leveraging existing desktop software technology.

2.3.1. Common Software Requirements

The PXI specification presents software frameworks for PXI systems including Microsoft Windows NT and Windows 95. A PXI controller operating in either framework must support the currently available operating system and must support future upgrades. The benefit of this requirement is that the controller therefore also supports the most popular industry-standard application programming interfaces, such as Microsoft and Borland C++, Visual Basic, LabVIEW, and LabWindows®/CVI.

PXI also requires that all peripheral modules have device driver software that runs in the appropriate framework. Hardware vendors for other industrial buses that do not have software standards often do not provide any software drivers for

their devices. The customer is often given only a manual, which describes how to write software to control the device. The cost to the customer, in terms of engineering effort to support these devices, can be enormous. PXI removes this burden by requiring that manufacturers, rather than customers, develop the driver software.

2.3.2. Virtual Instrument Software Standards

PXI systems are required to provide the VISA software standard as the mechanism for locating and controlling GPIB, VXI, serial, and PXI instruments. Bringing the VISA standard to PXI preserves an instrumentation user's investment in software. VISA provides the link from PXI to a VXI chassis and instruments and to standalone GPIB and serial instruments. VISA provides a standard mechanism to locate, configure, and control PXI modules in a user's system.

2.3.3. Other Software Requirements

PXI also requires that certain software components be made available by peripheral module and chassis vendors. Initialization files that define a system's configuration and capabilities are required for PXI components. The system software uses this information to ensure proper configuration of a system. For example, this mechanism is used to identify whether or not adjacent peripheral modules have a compatible local bus. If any information is missing, the local bus circuitry is not enabled and the functionality cannot be accessed.

3. Mechanical Requirements

This section discusses the additional mechanical requirements for PXI systems. It discusses the location of the system slot, the interoperability of the controller with the chassis, the PXI logo, environmental testing, cooling, grounding, and guidelines for minimizing electromagnetic interference (EMI).

3.1. CompactPCI Mechanical Requirements

Both 3U (100 by 160 mm, or 3.94 by 6.3 in.) and 6U (233.35 by 160 mm, or 9.19 by 6.3 in.) form factor modules can implement PXI features as defined in Section 4, *Electrical Requirements*.

RULE: All mechanical requirements defined in the CompactPCI specification (PICMG 2.0 R2.1) SHALL apply to 3U and 6U PXI compatible modules.

The following sections define additional mechanical requirements and recommendations that are included to ease system integration.

3.2. System Slot Location and Rules

All CompactPCI and PXI compatible systems require a backplane/chassis and a system controller module in the system slot. This requirement allows users to mix and match different controllers with different chassis. However, because the CompactPCI specification allows a system slot to be located anywhere relative to peripheral slots, the possibility for confusion and incompatibility exists. To address this problem the following rules must be followed for PXI-compatible systems:

RULE: The system slot SHALL be defined as the leftmost PXI slot in a basic single bus segment PXI chassis/backplane. For documentation purposes this slot is counted as one *controller* slot.

RECOMMENDATION: If the system controller module requires more than one slot width, it SHOULD extend to the LEFT of the system slot in full slot increments (one slot equals 20.32 mm, or 0.8 in.) into additional controller expansion slots.

OBSERVATION: These additional controller slots are for physical expansion of the system controller module only and cannot support peripheral modules. These slots DO NOT have connectors that interface to the PCI bus routed on the backplane.

OBSERVATION: Extending the system slot module to the LEFT allows all PXI peripheral slots to be utilized.

RECOMMENDATION: The system controller module SHOULD NOT extend to the RIGHT of the system slot into peripheral slots.

OBSERVATION: If a system controller module expands to the right, the number of usable PXI peripheral slots is compromised and access to the star trigger slot is eliminated.

RULE: Every PXI system controller module SHALL clearly document how many controller expansion slots (to the left of the system slot) and peripheral slots it occupies.

RULE: Every PXI chassis SHALL clearly document how many peripheral and controller expansion slots are available.

OBSERVATION: The two preceding rules help ensure that end users can easily determine whether a particular controller-chassis pair is compatible and how many peripheral slots are left available.

Figure 3.1 depicts slot designations in a PXI system.

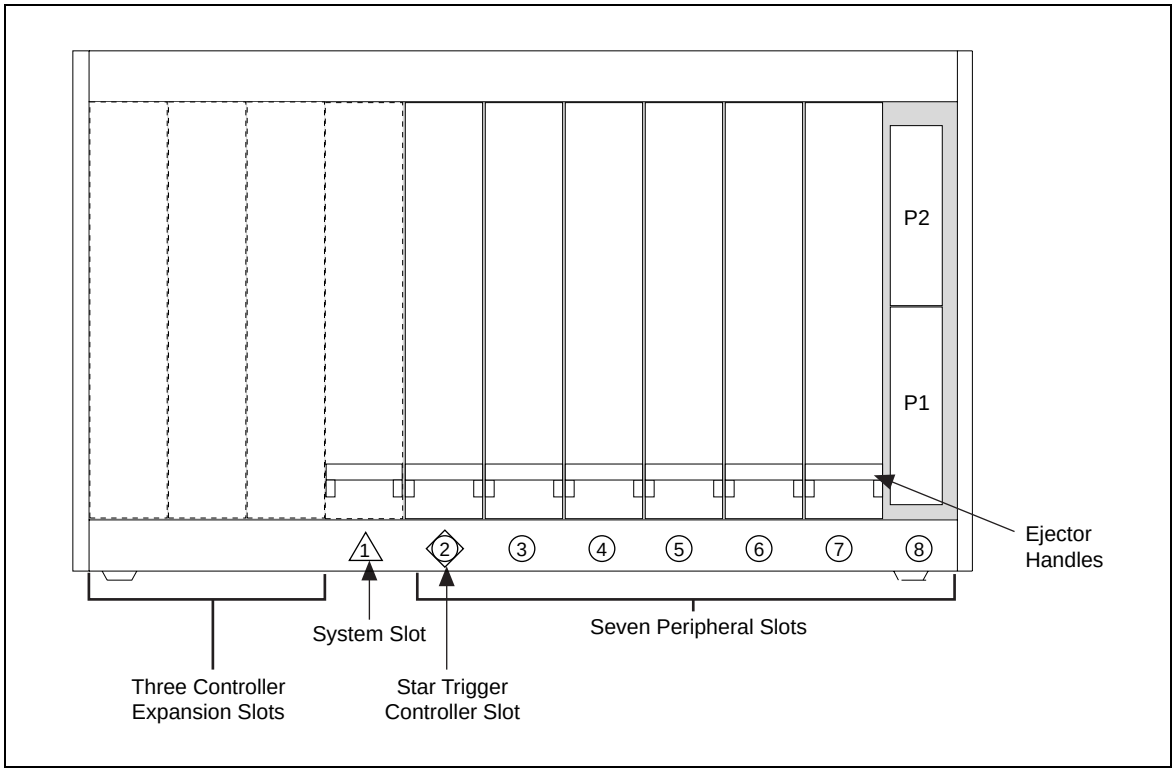

Figure 3.1 Example of PXI Slot Designations for a Chassis

3.3. Logos and Compatibility Glyphs

RULE: A PXI chassis SHALL clearly mark each physical slot with a unique number. The number SHALL be located within the compatibility glyph.

OBSERVATION: The preceding rule applies to a chassis with multiple bus segments as well as to a single-segment chassis.

RECOMMENDATION: Products claiming full compliance with the PXI specification SHOULD use the PXI logo as defined below on either the front panel or the injector/ejector handle.

RULE: If the PXI logo is used, it SHALL NOT be altered in any way other than scale. The logo SHALL NOT incorporate any additions.

Figure 3.2 shows the PXI logo. Logo artwork can be obtained by contacting National Instruments.

Figure 3.2 PXI Logo

PERMISSION: The PXI logo MAY be used as a substitute for the CompactPCI logo defined in PICMG 2.0 R2.1. Additionally, a product MAY display both the CompactPCI logo and the PXI logo.

RULE: Modules and backplanes SHALL use the compatibility glyphs as defined in the PICMG 2.0 R2.1 (CompactPCI specification).

RULE: Modules and backplanes SHALL use the glyph shown in Figure 3.3 (square at 45°) for the star trigger slot/module. This slot is always located next to the system slot. Because this slot can also support standard peripheral modules, the peripheral module glyph (circle) SHALL also be used for this slot.

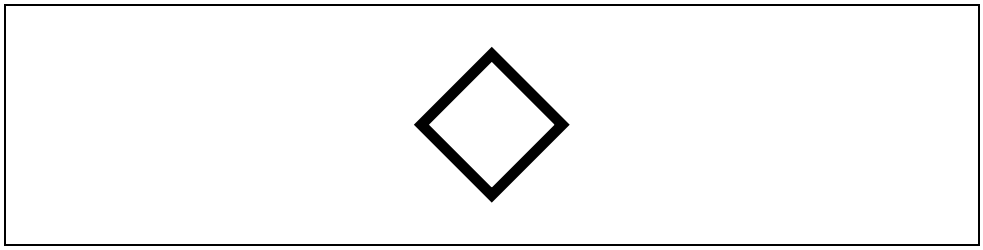

Figure 3.3 Star Trigger Slot Glyph

3.4. Environmental Testing

RULE: PXI chassis, system controller modules, and peripheral modules SHALL be tested for storage and operating temperature ranges.

RECOMMENDATION: PXI chassis, system controller modules, and peripheral modules SHOULD be tested for humidity, vibration, and shock.

RECOMMENDATION: All environmental testing SHOULD be carried out according to the procedures described in IEC 60068.

RULE: Test results and reports generated for environmental testing SHALL be made available to end users of PXI systems. All manufacturers of PXI chassis, system controller modules, and peripheral modules SHALL supply operating and storage temperature ratings for their products.

RULE: If a manufacturer chooses to use environmental testing procedures other than those described in IEC 60068, then these procedures, in addition to the test results and reports, SHALL be documented and made available to the customer.

OBSERVATION: It is the system integrator's responsibility to select modules and chassis appropriate for the application's environmental requirements.

3.5. Cooling Specifications

3.5.1. Plug-in Module Requirements

RULE: Plug-in modules SHALL be designed to allow a suitable airflow path from bottom to top as shown in Figure 3.4.

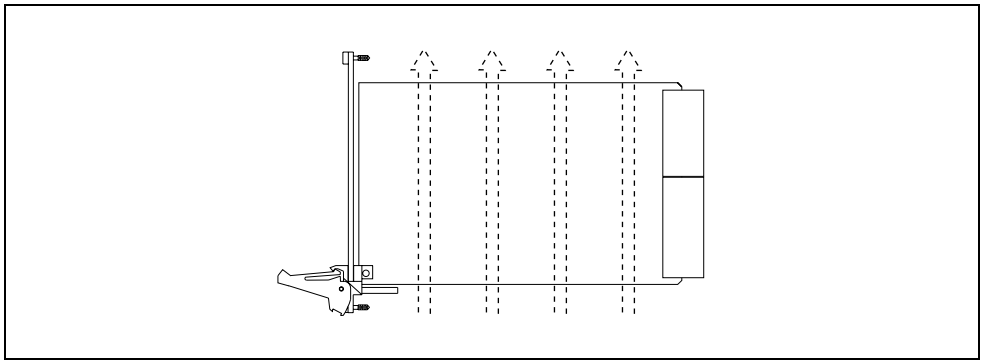

Figure 3.4 Cooling Airflow Direction in a PXI System

RULE: Plug-in module manufacturers SHALL document and make available to the customer the nominal wattage generated by the module under normal operating conditions.

3.5.2. Chassis Requirements

RULE: Chassis SHALL provide active cooling that flows from the bottom to the top of a plugged-in module as shown in Figure 3.4.

RULE: Chassis manufacturers SHALL document and make available to the customer the maximum total wattage that a given chassis can dissipate and the maximum wattage dissipated for the worst-case slot. Furthermore, the manufacturer SHALL document and make available to the customer the specific test procedure used to determine these wattage values.

3.6. Chassis and Module Grounding Requirements and EMI Guidelines

RULE: Chassis SHALL provide a direct (low-impedance) means for connecting the chassis ground to earth ground.

OBSERVATION: A grounding stud may be used to provide direct access to chassis ground.

RECOMMENDATION: As defined in PICMG 2.0 R2.1, PXI modules SHOULD use metalized shell connectors for EMI/RFI protection. The shell SHOULD be electrically connected to the front panel through a low impedance path in accordance with IEEE 1101.10.

RULE: As defined in PICMG 2.0 R2.1, PXI modules SHALL NOT connect chassis ground through a low-impedance path to logic ground used onboard.

3.7. Regulatory Requirements

The following standards assure uniform performance and international portability of PXI systems and modules. Allowances are made for devices not intended for international sale. All regulatory compliance information must be clearly documented for the user. Subsequently issued standards or amendments to these standards shall apply.

3.7.1. Requirements for EMC

RULE: Testing SHALL be performed, either by the manufacturer or a competent laboratory, marked accordingly, and documented showing compliance to the following EMC standard(s).

IEC 61326-1:1997, *Electrical equipment for measurement, control, and laboratory use—EMC requirements—Part I, General requirements*[1]:

- Localized EMC standards may be substituted if sale and use are restricted accordingly.[2]
- Use current edition of IEC CISPR-11, Group 1, Class A or Class B Limits, at 10 m.
- Continuous unmonitored operation is assumed unless otherwise specified by manufacturer.

[1] At the time of this writing, IEC 61326-1 was not yet normalized for CE Marking, but its first edition has been published by the IEC as a bonafide standard. The ISM emission and generic immunity standards may be substituted during any legally allowed transition period if documented accordingly.

3.7.2. Requirements for Electrical Safety

RULE: Testing SHALL be performed, either by a competent organization or qualified manufacturer, marked accordingly, and documented showing compliance to the following electrical safety standard(s). Strictly extra low voltage (SELV) devices do not need formal agency testing, though the basic requirements still apply, such as material flammability specified, DC power outputs fused or limited, and so on.

IEC 1010-1:1990 + A1:1992, *Safety requirements for electrical equipment for measurement, control, and laboratory use—Part 1, General requirements*:

- IEC 950 and amendments are acceptable for applications restricted to the office use only.
- Localized safety standards may be substituted if sale and use are restricted accordingly. [2]

3.7.3. Particular Requirements for Chassis

RULE: Chassis manufacturers, or a designated party, SHALL demonstrate EMC compliance with a commonly available controller. At minimum the controller SHALL apply a 33 MHz clock to the backplane, or higher as the technology commonly supports, unless the marketing and use of the chassis is restricted to specific kinds of system controllers or platforms. A reasonably common processor speed is sufficient for this test. The complete controller system, including hard drive, floppy drive, serial, parallel, keyboard, mouse, and video ports (as offered with the controller) will be exercised with typical peripherals. Filling remaining slots, if any, is not required. Special shielding or suppression options, if used to achieve compliance, SHALL be documented and available to the user.

RULE: The chassis SHALL be qualified for electrical safety as listed previously. Manufacturer claims of safety compliance are not sufficient without independent verification and regular inspection by a competent agency.

[2] The use of localized EMC or safety standards must be clearly documented for the benefit of the user. The standards used may include standards in force during legally allowed transitional periods of new standards or amendments. Currently, manufacturer declarations must list all standards used for declaring compliance and conveniently meet this requirement when the declarations are included with the user documentation.

4. Electrical Requirements

This chapter discusses the detailed electrical requirements for developing PXI-compatible peripheral modules, controllers, and backplanes. It discusses all signals required for PXI systems and their related implementation requirements. It also discusses appropriate connector pinouts, power supply requirements, and 6U form factor implementation issues.

4.1. PXI Signal Groups

This section details all the signals required for PXI systems, including those mapped directly from CompactPCI on P1/J1, the 64-bit signals duplicated on P2/J2, and the PXI-specific signals located on P2/J2. The PXI-specific signals include those required for the trigger bus, the reference clock, the local buses, and the star trigger.

4.1.1. P1/J1: Signals

RULE: All signals on the P1/J1 connector of a PXI module and the mating P1 connector of a PXI backplane SHALL meet all the requirements of the PICMG 2.0 R2.1 specification (CompactPCI) for both peripheral modules and system modules.

RULE: Backplane devices that are not PCI-PCI bridges and peripheral slots SHALL have their IDSEL lines connected to one of AD[25:31].

RULE: Backplane PCI-PCI bridges SHALL have their IDSEL lines connected to AD lines other than AD[25:31].

To provide enhanced legacy ISA support, PXI allows for a serial interrupt mechanism on the INTP or INTS line. This serial mechanism allows for ISA interrupts to be generated on any peripheral module.

PERMISSION: The system controller module MAY use INTP or INTS as a serialized IRQ line for support of multiple legacy ISA interrupts from multiple modules. Refer to revision 6.0 of the *Serialized IRQ Support for PCI Systems Specification* for details of the protocol.

OBSERVATION: If the system controller module uses INTP or INTS as a serialized IRQ, peripheral modules in the given PXI backplane will not be able to successfully drive the serialized IRQ line (INTP or INTS) as a traditional ISA interrupt.

4.1.2. P2/J2: Signals

The PICMG 2.0 R2.1 (CompactPCI) specification defines an open pinout for P2/J2, which allows for custom back-panel I/O implementations. This pinout allows sub-buses on the P2 section of the backplane.

The PXI back-panel I/O definition of signals on the P2/J2 connector consists of signals mapped from the CompactPCI 64-bit connector pinout and new signals defined by this specification. The only differences between the PXI pinout and the CompactPCI 64-Bit connector pinout are on signals that are reserved or not used in the CompactPCI specification (PICMG 2.0 R2.1). Therefore, all modules that meet the requirements of the CompactPCI 64-bit specification will function in a PXI system. Furthermore, all PXI modules will work in a system that meets the requirements of the CompactPCI 64-bit specification, but without the complete set of benefits of PXI.

4.1.2.1. Signals from CompactPCI 64-Bit Connector Specification

To allow for applications that require higher bandwidth than 32-bit PCI can provide, PXI has incorporated the pieces of the CompactPCI 64-bit specification that are necessary to support 64-bit operation and has used the remaining resources for instrumentation. This section describes the signals that are mapped from the CompactPCI 64-bit specification. The remaining signals are described in subsequent sections.

System Slot

RULE: The following signals SHALL meet all the requirements of the PICMG 2.0 R2.1 specification for the 64-bit CompactPCI pinout on the system slot: GND, V(I/O), AD[32:63], C/BE[4:7]#, DEG#, FAL#, PRST#, SYSEN#, CLK[1:6], GNT[1:6]#, REQ[1:6]#, and the RSV pins.

RULE: The system slot interface in a PXI system SHALL be as defined in the PICMG 2.0 R2.1 Specification (CompactPCI), except as noted in the following rule:

RULE: The system controller module SHALL provide full support for a REQ#/GNT# pair to each slot in the given backplane (up to seven slots). This support SHALL NOT be selectable. Instead, it SHALL be available to all slots in a system at once.

RECOMMENDATION: When using external logic on the system controller module to convert one REQ/GNT pair to two pair, one SHOULD make sure the timing meets all PCI setup/hold requirements. The unassertion of an additional grant SHOULD be in the same clock as that of the original grant so that the requesting master does not start a cycle prematurely. A cycle can start prematurely if the system controller module grants the bus to a module while a cycle is pending and then removes the grant before the cycle is complete. It might be necessary for the system controller module to park the PCI bus to the system slot.

OBSERVATION: The PXI implementation of back-panel I/O uses the pins that are BRSV (bused reserved) in the CompactPCI 64-bit pinout for directly accessing PXI features on system modules.

Peripheral Module

RULE: The following signals SHALL meet all the requirements of the PICMG 2.0 R2.1 specification for the 64-bit CompactPCI pinout on the peripheral slots: GND, V(I/O), AD[32:63], C/BE[4:7]#, DEG#, FAL#, and PRST#, and SYSEN#.

OBSERVATION: The PXI implementation of back-panel I/O uses the pins that are BRSV, CLK[1:6], GNT[1:6]#, REQ[1:6]#, and RSV in the CompactPCI 64-bit pinout for instrumentation features on peripheral modules. Note that the CLK[1:4], GNT[1:6]# and REQ[1:6]# signals are not implemented on CompactPCI peripheral modules and slots.

Backplane

RULE: The following signals SHALL meet all the requirements of the PICMG 2.0 R2.1 specification for the 64-bit CompactPCI pinout on the backplane: GND, V(I/O), AD[32:63], C/BE[4:7]#, DEG#, FAL#, and PRST#, and SYSEN#.

RULE: As in the CompactPCI specification, the backplane SHALL route CLK[1:4], GNT[1:6]#, and REQ[1:6]# from the system slot to the appropriate pins of the J1 connectors on the peripheral slots.

RULE: The backplane SHALL leave CLK5 and CLK6 of the system slot disconnected.

RULE: The backplane SHALL leave all the RSV lines on the system slot disconnected.

OBSERVATION: The PXI implementation of back-panel I/O uses the pins that are BRSV, CLK[1:6], GNT[1:6]#, REQ[1:6]#, and RSV (reserved) on peripheral slots and the BRSV pins on the system slot in the CompactPCI 64-bit pinout for instrumentation features on the backplane. The CLK[1:6], GNT[1:6]# and REQ[1:6]# signals are not implemented on CompactPCI backplane peripheral slots.

OBSERVATION: The PXI implementation of back-panel I/O routes all of the bused reserved signals from the CompactPCI 64-bit pinout as bused signals on the PXI backplane. However, most of the signals are not reserved in the PXI signal definition and are explained in the Section 4.1.2.5, *Trigger Bus*.

4.1.2.2. PXI Reserved and Bused Reserved Signals

System Slot

The following signals are reserved for future PXI use: PXI_BRSVA15, PXI_BRSVB4, PXI_RSVA22, PXI_RSVB22, PXI_RSVC22, PXI_RSVD22, and PXI_RSVE22.

RULE: A system controller module SHALL NOT connect to the PXI_BRSV signals.

Peripheral Slots

The following signals are reserved for PXI use: PXI_BRSVA15, PXI_BRSVB4, PXI_RSVA22, PXI_RSVB22, PXI_RSVC22, PXI_RSVD22, and PXI_RSVE22.

RULE: A peripheral module SHALL NOT connect to the PXI_RSV signals.

Backplane

The following signals are reserved for PXI use: PXI_BRSVA15, PXI_BRSVB4, PXI_RSVA22, PXI_RSVB22, PXI_RSVC22, PXI_RSVD22, and PXI_RSVE22.

RULE: A backplane SHALL NOT connect to the PXI_RSV signals.

RULE: A backplane SHALL bus each PXI_BRSV signal to each slot.

OBSERVATION: PXI_BRSV and PXI_RSV signals are implemented on the backplane exactly like CompactPCI BRSV and RSV signals.

4.1.2.3. Local Buses

PXI implements a daisy-chained local bus between adjacent peripheral modules. The local bus is a user-definable bus (13 lines wide) that can be used for a wide variety of applications. The range of applications may vary from passing an analog signal between two modules to high-speed data movement that does not affect the PXI bandwidth. For most slots, the functionality of the local bus is user-definable; however, there are specific implementations that the PXI specification defines.

Note that the system controller module does not implement the local bus as it uses those pins for PCI arbitration and clocking functionality. The slot adjacent to the system slot uses its available local bus left signals for the star trigger functionality defined later. Similarly, the rightmost slot of the bus has no slot to its right; hence, the local bus right pins can either be unused or

routed for a chassis-specific implementation. One example of chassis-specific implementation is routing the local bus to another adjacent PXI bus segment. The remainder of the slots can use the local bus for any specified purpose. Refer to Figure 2.4 for a diagram of local bus routing.

For system configuration purposes, the chassis configuration of the local bus can be determined by the initialization (`.ini`) file of the chassis. The `.ini` file lists each slot and its location in the system. The system configuration software uses this information to determine whether or not a system is put together properly.

The following rules apply to all implementations of the local bus:

Peripheral Module

RULE: A peripheral module SHALL NOT drive more than ± 42 V onto any local bus line.

RULE: A peripheral module SHALL NOT drive more than 200 mA DC current into any local bus line.

PERMISSION: A peripheral module MAY connect its local bus left to its local bus right if passing the local bus to an additional module is required. This connection should be performed with caution because the length and/or characteristic impedance specifications of the local bus might be violated.

PERMISSION: On a peripheral module, a local bus signal MAY be connected to ground.

OBSERVATION: The preceding permission allows local bus implementations to have improved grounding and/or shielding.

RULE: On a peripheral module, if a local bus signal is not connected to ground, it SHALL power up in a high-impedance state. All local bus signals (except grounded ones) SHALL remain in a high-impedance state until system configuration software has determined that the local bus signals are compatible with the chassis and other peripheral modules in the system.

PERMISSION: A peripheral module MAY pull-up a local bus signal to V (I/O) to prevent an input from powering up in an unstable state.

RULE: Each peripheral module SHALL have a maximum input leakage current of 100 μA on each local bus line.

Backplane

RULE: On each bus segment, a backplane SHALL route PXI_LBR[0:12] from each slot in column A of Table 4.1 to the PXI_LBL[0:12] of the corresponding slot in column B, provided that both slots exist on the bus segment.

Table 4.1 Local Bus Routings

A	B
IDSEL = 31	IDSEL = 30
IDSEL = 30	IDSEL = 29
IDSEL = 29	IDSEL = 28
IDSEL = 28	IDSEL = 27
IDSEL = 27	IDSEL = 26
IDSEL = 26	IDSEL = 25

RULE: The backplane SHALL NOT make any other local bus routings.

OBSERVATION: There is no local bus connection to the system slot.

RULE: The backplane SHALL NOT provide any termination or buffering of local bus signals. Each signal SHALL be implemented as a direct connection between adjacent peripheral slots.

RULE: The backplane SHALL NOT route the local bus between two bus segments.

RULE: Bus segment boundaries SHALL be documented.

OBSERVATION: Documenting bus segment boundaries allows users to optimize systems by grouping boards that frequently talk to each other on the same segment. It also allows users to know when adjacent physical slots do not have a local bus connection.

RULE: The signal length of local bus signals between slots SHALL be less than 3 in. and SHALL be matched within 1 in. between all local bus traces. The characteristic impedance of each trace SHALL be 65 Ω ± 10%.

PERMISSION: A slot at the end of a PXI segment or next to the system slot has one local bus that is not routed to another peripheral slot. The backplane MAY route these local buses for other uses.

RULE: A star trigger slot SHALL NOT implement a local bus left. Instead, the remaining pins SHALL be routed to support the star triggers.

RULE: If a chassis implements an external backplane interface with the local bus right of a slot at the end of a PXI segment, it SHALL be the highest numbered slot in the chassis.

OBSERVATION: A chassis will have at most one slot with an external backplane interface.

4.1.2.4. Reference Clock: PXI_CLK10

The PXI backplane is responsible for providing a common reference clock for synchronization of multiple modules in a measurement or control system. The variable frequency of the PCI system clock limits its usefulness in this application. PXI_CLK10 is a 10 MHz clock provided independently to each peripheral slot. PXI_CLK10 can be used to run multiple modules in synchronization from a common reference. The low skew qualities make it ideal for qualifying trigger protocols.

RULE: The PXI_CLK10 provided by the backplane SHALL be a 10 MHz TTL signal. Its accuracy SHALL be ± 100 ppm or better over the specified operating temperature and time.

OBSERVATION: Because of drift over time, the oscillator chosen MAY need to be 50 ppm accurate or better to meet the requirement of the preceding rule.

RULE: The PXI_CLK10 signal SHALL have a 50% ± 5% duty cycle when measured at the 2.0 V transition point.

RULE: The clock to each peripheral slot SHALL be driven by an independent buffer that has a source impedance matched to the backplane.

RULE: The backplane SHALL provide for a skew of less than 1 ns between slots.

RECOMMENDATION: A backplane SHOULD allow PXI_CLK10 to be derived from an external source to allow for a more accurate reference.

OBSERVATION: The star trigger slot has defined a pin for providing an external clock.

RULE: If the PXI_CLK10 is switched between sources, the minimum pulse width created SHALL be no less than 30 ns and the minimum time between successive edges of the same polarity SHALL be no less than 80 ns.

OBSERVATION: The preceding rule is intended to prevent a state machine from being corrupted by glitches in the clock during transition.

4.1.2.5. Trigger Bus

The PXI trigger bus provides intermodule synchronization and communication. The trigger bus lines can be used for trigger or clock transmission. A few standard triggering protocols are defined to facilitate interoperability, but use of the trigger bus is not limited to the defined protocols. The PXI trigger bus lines may be used as general purpose intermodule bused lines using other manufacturer-defined protocols.

The PXI trigger bus consists of the following eight signals:

PXI_TRIG[0:7]

Clock Transmission

Variable frequency clock transmission allows multiple modules to share a timebase that is not a derivative of the PXI_CLK10. For example, two data acquisition modules using a 44.1 kS/s CD audio sampling rate could share a clock that is a multiple of the 44.1 kHz. Type A drivers, defined later, prevent degradation of the clock jitter over the bused transmission medium.

PERMISSION: PXI_TRIG[0:7] MAY be used for variable frequency clock transmission.

RECOMMENDATION: To facilitate interoperability between sources and receivers of a variable frequency clock, PXI_TRIG[7] SHOULD be used to source and receive such clocks.

PXI Trigger Protocols

Some standard protocols are defined in the following sections.

PXI Asynchronous Trigger

The PXI asynchronous trigger protocol is a single-line broadcast trigger. Figure 4.1 and Table 4.2 show the timing parameters.

RULE: A PXI asynchronous trigger source SHALL meet the timing requirements listed in Table 4.2.

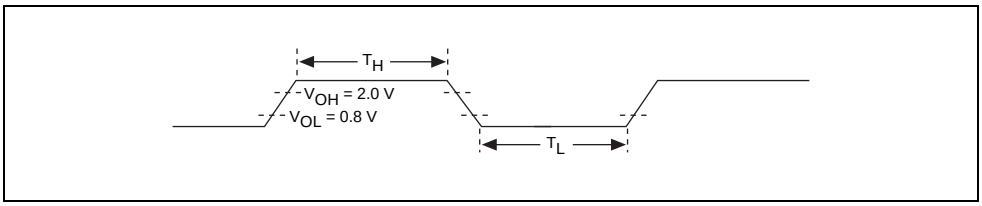

Figure 4.1 PXI Asynchronous Trigger Timing

Table 4.2 PXI Asynchronous Trigger Timing Parameters

Symbol	Parameter	Min	Max
T_h	pulse width high	18 ns	—
T_l	pulse width low	18 ns	—

Note:
The pulse width is measured at the driver with a 50 pf equivalent load.

OBSERVATION: The preceding rule guarantees PXI asynchronous trigger receivers a minimum of 10 ns pulse widths on PXI asynchronous trigger pulses.

PXI Synchronous Trigger

The PXI synchronous trigger protocol can be used to synchronize PXI_CLK10 derived timed operations on module clusters. A PXI_TRIG line is driven by a module and the participating modules respond to this line synchronously at the next PXI_CLK10 rising edge. The reference clock is PXI_CLK10.

Figure 4.2 shows PXI synchronous trigger timing.

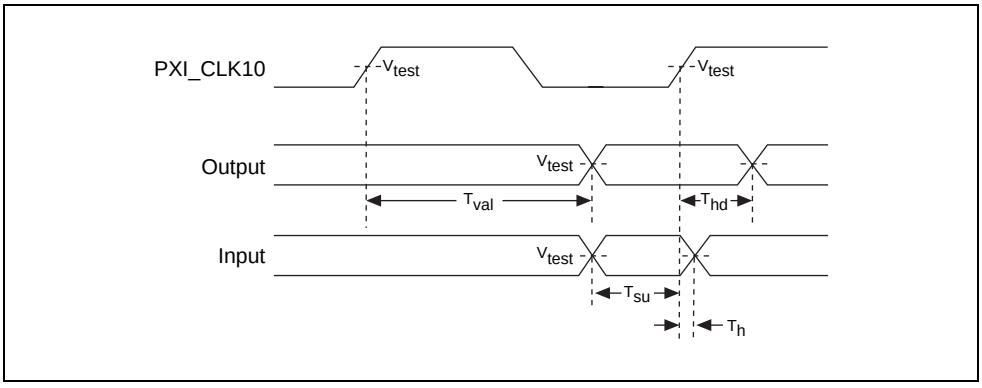

Figure 4.2 PXI Synchronous Trigger Timing

RULE: A PXI synchronous trigger source SHALL meet the timing requirements listed in Table 4.3.

Table 4.3 lists PXI synchronous trigger timing requirements.

Table 4.3 PXI Synchronous Trigger Timing

Symbol	Parameter	Min	Max
T_{hd}	Output hold time from PXI_CLK10	2 ns	—
T_{val}	Output signal valid from PXI_CLK10	—	65 ns
T_{su}	Input set up time to PXI_CLK10	23 ns	—
T_h	Input hold time from PXI_CLK10	0 ns	—

Note:
Minimum times are measured with 0 pf equivalent load. Maximum times are measured with 50 pf equivalent load.

OBSERVATION: The output signal valid timing from PXI_CLK10 allows either rising or falling edges of PXI_CLK10 to be used for sourcing PXI synchronous triggers.

Backplane

RULE: For each PXI segment in a PXI chassis, the PXI chassis SHALL bus the PXI_TRIG[0:7] signal to each PXI slot (system and peripheral) in that segment. A chassis SHALL NOT directly connect PXI_TRIG buses from different PXI segments. If a system slot controls multiple PXI segments, it SHALL NOT directly connect PXI trigger buses from different segments.

OBSERVATION: Trigger buses from multiple segments are physically disconnected to maintain signal integrity and allow for incident wave switching of Type A trigger drivers. However, the buses may be logically connected by buffering the signals between segments.

RULE: PXI_TRIG[0:7] SHALL be fast Schottky diode terminated at both ends of the bus segment on the backplane to + 5 V and ground as shown in Figure 4.3.

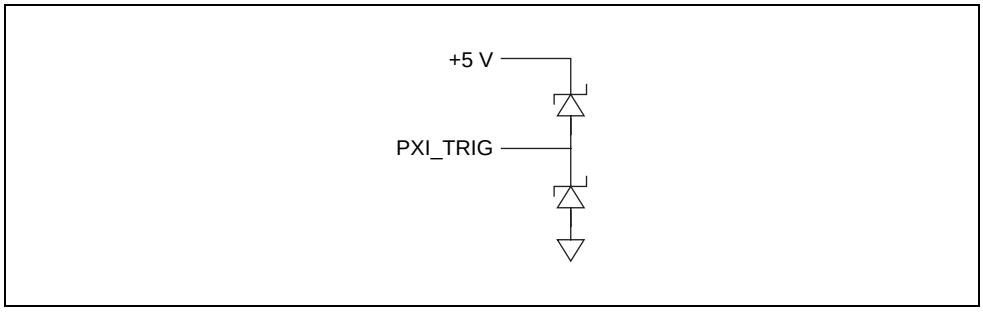

Figure 4.3 PXI Trigger Bus Termination

RULE: The *unloaded* characteristic impedance for the backplane $Z_{l,min}$ SHALL be 75 Ω ± 10% using a stripline transmission line geometry.

OBSERVATION: The higher backplane impedance lowers the driver strength requirements for incident wave switching.

RULE: The signal trace lengths of PXI trigger bus signals SHALL be less than 10 in. and matched within 1 in. between all trigger bus signals.

Peripheral or System Module

RULE: Printed circuit board trace lengths for PXI trigger bus signals SHALL be less than or equal to 1.5 in.

PERMISSION: A PXI module (system or peripheral) MAY leave any number of PXI trigger bus signals unconnected.

RECOMMENDATION: The following recommendations improve interoperability between modules using the PXI trigger bus. For triggering applications, if a PXI module (system or peripheral) connects to a subset of the PXI trigger signals, it SHOULD connect to PXI_STAR and PXI_TRIG[0:*n*-2] where *n* is the number of trigger lines on the peripheral module. For sourcing or receiving clocks, PXI_TRIG[7] SHOULD be used.

RULE: Upon power up, the PXI_TRIG[0:7] lines and drivers SHALL remain in a high impedance state until configured by software.

RULE: PXI_TRIG[0:7] I/O buffers SHALL be compliant with the DC specifications listed in Table 4.4.

Table 4.4 DC Specifications

Symbol	Parameter	Condition	Min	Max	Notes
Vih	Input High Voltage	—	2.0 V	Vcc + 0.5 V	1
Vil	Input Low Voltage	—	-0.5 V	0.8 V	—
Il	Leakage Current	0 < Vin < Vcc	—	± 70 µA	1, 2
Voh	Output High Voltage	Iout = - 2 mA	2.4 V	—	—
Vol	Output Low Voltage	Iout = 4 mA	—	0.55 V	—
Cpin	Input, Output, Bidirectional Pin Capacitance	—	—	10 pf	—

Notes:
1. Vcc refers to the 5 V power supply rail.
2. Leakage current includes output leakage for bidirectional buffers in a high-impedance state.

OBSERVATION: The DC specifications are met by 5 V tolerant LVTTL-compatible I/O buffers.

PERMISSION: To prevent floating inputs, PXI_TRIG[0:7] lines used on a PXI module (system or peripheral) MAY be pulled up on the module with a pull-up resistance whose value is shown in Table 4.5.

Table 4.5 Pull-Up Resistor Values

Signaling Rail	Rmin	Rmax
5 V	11 kΩ	[Vccmin - Vx]/[Iih]*
3.3 V	19 kΩ	[Vccmin - Vx]/[Iih]*

* Vx = 2.4 V, which is the desired voltage on the bus in a non-driven state, and Iih is the maximum leakage current for the device buffer.

RECOMMENDATION: Type A drivers SHOULD be used for clock transmission over the PXI trigger bus. Type A drivers are capable of incident wave switching on rising edges, preventing jitter degradation due to transmission line effects. Refer to Table 4.6 for Type A driver specifications.

Table 4.6 Type A, High Current Driver, AC Specifications

Symbol	Parameter	Condition	Min	Max
Vsoh(AC)	Switching voltage high		2.3 V	—
Ioh(AC)	High source current	@Vsoh(AC)	75 mA	—

RECOMMENDATION: Intermediate voltage levels (Vol ≤ V ≤ Voh) may be present on the trigger bus. Schmitt trigger inputs SHOULD be used to guard against sensing multiple transitions on edge-sensitive inputs when the voltage on the bus is at intermediate levels due to transmission line effects on the bus.

RECOMMENDATION: Rising edges SHOULD be used as active edges in asynchronous trigger protocols.

4.1.2.6. Star Trigger

In addition to the bused PXI triggers, the PXI bus has included an independent trigger (PXI_STAR) for each slot that is oriented in a star configuration from the star trigger slot. The star trigger slot is adjacent to the system slot and uses the 13 left local bus signals as the star triggers. This allows a single star trigger slot to control or monitor triggers in two PCI bus segments. In systems with more than two PCI segments, one star trigger slot is designated for two bus segments.

The PXI specification does not specify the functionality of the star trigger slot module or even require that the system be constructed with a star trigger module. Typical uses would include triggering multiple modules independently with low skew, monitoring a trigger from peripheral slots, and routing triggers between slots. One star trigger slot pin is dedicated to allow an external 10 MHz frequency standard to be routed as PXI_CLK10.

Backplane

RULE: Slot 2 in a chassis SHALL be a star trigger slot.

RULE: A chassis SHALL NOT have more than one star trigger slot.

OBSERVATION: The star trigger slot may also be used as a generic peripheral slot with the exception that the left side local bus is unavailable because these pins connect to the star triggers. The star trigger slot does not have a PXI_STAR; instead, the pin in the star trigger slot is used for an external frequency reference.

RULE: The PXI backplane SHALL route the signals from the star trigger slot to each peripheral slot according to Table 4.7, with a trace impedance of 65 Ω ± 10%. The mapping of the connections is described in the chassis`.ini` file.

Table 4.7 Star Trigger Mapping

Star Trigger Signal	Physical Peripheral Slot
PXI_STAR0	3
PXI_STAR1	4
PXI_STAR2	5
PXI_STAR3	6
PXI_STAR4	7
PXI_STAR5	8
PXI_STAR6	9
PXI_STAR7	10
PXI_STAR8	11
PXI_STAR9	12
PXI_STAR10	13
PXI_STAR11	14
PXI_STAR12	15

RULE: The PXI_STAR line lengths SHALL be matched in propagation delay to within 1 ns, and the delay from the star trigger slot to each peripheral module SHALL NOT exceed 5 ns.

Peripheral Module

RULE: The driver of PXI_STAR, which may be the star trigger module or a peripheral module, SHALL have a source impedance of 65 Ω ± 10% to match the backplane impedance.

RULE: When a peripheral module or star trigger controller drives a PXI_STAR, the signaling levels SHALL NOT exceed 5 V.

RULE: A peripheral module SHALL NOT drive its PXI_STAR when reset.

PERMISSION: A peripheral module MAY pull-up the PXI_STAR signal to prevent an unstable input.

RULE: The leakage current of a peripheral module connected to PXI_STAR SHALL NOT exceed 650 μA.

OBSERVATION: The same trigger protocols defined for the PXI TTL trigger bus MAY be used on the PXI_STAR signals.

PERMISSION: The PXI_CLK10_IN signal of the star trigger slot MAY be used to provide an external 10 MHz reference for PXI_CLK10.

RULE: A module in the star trigger slot SHALL NOT drive the PXI_STAR/CLK10_IN signal, except to provide a reference for PXI_CLK10.

OBSERVATION: The presence of a periodic signal on the PXI_CLK10_IN signal MAY be used to indicate that an external reference is to be used.

4.1.3. Electrical Guidelines for 6U

Larger 6U size modules are desirable for the extra module space and possibly for future additional functionality that can be provided through the J3, J4, and J5 connectors. In many cases, the extra space on a 6U module is needed only for extra circuitry, and the module requires only the J1 and J2 connectors for PXI. J3, J4, and J5 are reserved for future revisions of the PXI specification.

RULE: 6U PXI peripheral modules SHALL implement only J1 and J2. J3, J4, and J5 SHALL NOT be loaded on 6U peripheral modules.

4.2. Connector Pin Assignments (J1/P1 and J2/P2)

To help in reviewing the tables in this section and locating the appropriate specification for signal requirements, Table 4.8 lists all signals alphabetically by original specification (PXI, CompactPCI, or PCI).

Table 4.8 PXI System Signal Groups

System	Signals		
PXI	PXI_BRSV	PXI_LBL[0:12]	PXI_STAR
	PXI_CLK10	PXI_LBR[0:12]	PXI_STAR[0:12]
	PXI_CLK10_IN	PXI_RSV	PXI_TRIG[0:7]
CompactPCI	BRSV	FAL#	PRST#
	CLK[0:6]	GNT#[0:6]	REQ#[0:6]
	DEG#	INTP	RSV
	ENUM#	INTS	SYSEN#
PCI	ACK64#	IRDY#	STOP#
	AD[0:63]	LOCK#	TCK
	C/BE[0:7]#	M66EN	TDI
	CLK	PAR	TDO
	DEVSEL#	PAR64	TMS
	FRAME#	PERR#	TRDY#
	GND	REQ#	TRST#
	GNT#	REQ64#	V(I/O)
	IDSEL	RST#	3.3 V
	INTA#	SBO#	5 V
	INTB#	SDONE	+12 V
	INTC#	SERR#	-12 V
	INTD#		

4.2.1. General Peripheral Slots

Table 4.9 gives the peripheral slot pinout for the J1 and J2 connector. PXI-specific signals are shown in **bold**.

RULE: Peripheral modules and backplane peripheral slots SHALL use the pinout in Table 4.9.

Table 4.9 Generic Peripheral Slot Pinout

22	GND	**PXI_RSVA22**	**PXI_RSVB22**	**PXI_RSVC22**	**PXI_RSVD22**	**PXI_RSVE22**	GND	
21	GND	**PXI_LBR0**	GND	**PXI_LBR1**	**PXI_LBR2**	**PXI_LBR3**	GND	
20	GND	**PXI_LBR4**	**PXI_LBR5**	**PXI_LBL0**	GND	**PXI_LBL1**	GND	
19	GND	**PXI_LBL2**	GND	**PXI_LBL3**	**PXI_LBL4**	**PXI_LBL5**	GND	
18	GND	**PXI_TRIG3**	**PXI_TRIG4**	**PXI_TRIG5**	GND	**PXI_TRIG6**	GND	P2
17	GND	**PXI_TRIG2**	GND	PRST#	**PXI_STAR**	**PXI_CLK10**	GND	/
16	GND	**PXI_TRIG1**	**PXI_TRIG0**	DEG#	GND	**PXI_TRIG7**	GND	J2
15	GND	**PXI_BRSVA15**	GND	FAL#	**PXI_LBL6**	**PXI_LBR6**	GND	
14	GND	AD[35]	AD[34]	AD[33]	GND	AD[32]	GND	
13	GND	AD[38]	GND	V(I/O)	AD[37]	AD[36]	GND	C
12	GND	AD[42]	AD[41]	AD[40]	GND	AD[39]	GND	O
11	GND	AD[45]	GND	V(I/O)	AD[44]	AD[43]	GND	N
10	GND	AD[49]	AD[48]	AD[47]	GND	AD[46]	GND	N
9	GND	AD[52]	GND	V(I/O)	AD[51]	AD[50]	GND	E
8	GND	AD[56]	AD[55]	AD[54]	GND	AD[53]	GND	C
7	GND	AD[59]	GND	V(I/O)	AD[58]	AD[57]	GND	T
6	GND	AD[63]	AD[62]	AD[61]	GND	AD[60]	GND	O
5	GND	C/BE[5]#	GND	V(I/O)	C/BE[4]#	PAR64	GND	R
4	GND	V(I/O)	**PXI_BRSVB4**	C/BE[7]#	GND	C/BE[6]#	GND	
3	GND	**PXI_LBR7**	GND	**PXI_LBR8**	**PXI_LBR9**	**PXI_LBR10**	GND	
2	GND	**PXI_LBR11**	**PXI_LBR12**	SYSEN#	**PXI_LBL7**	**PXI_LBL8**	GND	
1	GND	**PXI_LBL9**	GND	**PXI_LBL10**	**PXI_LBL11**	**PXI_LBL12**	GND	
25	GND	5V	REQ64#	ENUM#	3.3V	5V	GND	
24	GND	AD[1]	5V	V(I/O)	AD[0]	ACK64#	GND	
23	GND	3.3V	AD[4]	AD[3]	5V	AD[2]	GND	
22	GND	AD[7]	GND	3.3V	AD[6]	AD[5]	GND	
21	GND	3.3V	AD[9]	AD[8]	M66EN	C/BE[0]#	GND	
20	GND	AD[12]	GND	V(I/O)	AD[11]	AD[10]	GND	
19	GND	3.3V	AD[15]	AD[14]	GND	AD[13]	GND	P1
18	GND	SERR#	GND	3.3V	PAR	C/BE[1]#	GND	/
17	GND	3.3V	SDONE	SBO#	GND	PERR#	GND	J1
16	GND	DEVSEL#	GND	V(I/O)	STOP#	LOCK#	GND	
15	GND	3.3V	FRAME#	IRDY#	GND	TRDY#	GND	
12–14	Key Area							C
11	GND	AD[18]	AD[17]	AD[16]	GND	C/BE[2]#	GND	O
10	GND	AD[21]	GND	3.3V	AD[20]	AD[19]	GND	N
9	GND	C/BE[3]#	IDSEL	AD[23]	GND	AD[22]	GND	N
8	GND	AD[26]	GND	V(I/O)	AD[25]	AD[24]	GND	E
7	GND	AD[30]	AD[29]	AD[28]	GND	AD[27]	GND	C
6	GND	REQ#	GND	3.3V	CLK	AD[31]	GND	T
5	GND	BRSVP1A5	BRSVP1B5	RST#	GND	GNT#	GND	O
4	GND	BRSVP1A4	GND	V(I/O)	INTP	INTS	GND	R
3	GND	INTA#	INTB#	INTC#	5V	INTD#	GND	
2	GND	TCK	5V	TMS	TDO	TDI	GND	
1	GND	5V	-12V	TRST#	+12V	5V	GND	
Pin	**Z**	**A**	**B**	**C**	**D**	**E**	**F**	

4.2.2. System Slot

Table 4.10 gives the system slot pinout for the J1 and J2 connectors. PXI-specific signals are shown in **bold**.

RULE: System modules and backplane system slots SHALL use the pinout in Table 4.10.

Table 4.10 System Slot Pinout

22	GND	**PXI_RSVA22**	**PXI_RSVB22**	**PXI_RSVC22**	**PXI_RSVD22**	**PXI_RSVE22**	GND	
21	GND	CLK6	GND	RSV	RSV	RSV	GND	
20	GND	CLK5	GND	RSV	GND	RSV	GND	
19	GND	GND	GND	RSV	RSV	RSV	GND	
18	GND	**PXI_TRIG3**	**PXI_TRIG4**	**PXI_TRIG5**	GND	**PXI_TRIG6**	GND	P2
17	GND	**PXI_TRIG2**	GND	PRST#	REQ6#	GNT6#	GND	/
16	GND	**PXI_TRIG1**	**PXI_TRIG0**	DEG#	GND	**PXI_TRIG7**	GND	J2
15	GND	**PXI_BRSVA15**	GND	FAL#	REQ5#	GNT5#	GND	
14	GND	AD[35]	AD[34]	AD[33]	GND	AD[32]	GND	
13	GND	AD[38]	GND	V(I/O)	AD[37]	AD[36]	GND	C
12	GND	AD[42]	AD[41]	AD[40]	GND	AD[39]	GND	O
11	GND	AD[45]	GND	V(I/O)	AD[44]	AD[43]	GND	N
10	GND	AD[49]	AD[48]	AD[47]	GND	AD[46]	GND	N
9	GND	AD[52]	GND	V(I/O)	AD[51]	AD[50]	GND	E
8	GND	AD[56]	AD[55]	AD[54]	GND	AD[53]	GND	C
7	GND	AD[59]	GND	V(I/O)	AD[58]	AD[57]	GND	T
6	GND	AD[63]	AD[62]	AD[61]	GND	AD[60]	GND	O
5	GND	C/BE[5]#	GND	V(I/O)	C/BE[4]#	PAR64	GND	R
4	GND	V(I/O)	**PXI_BRSVB4**	C/BE[7]#	GND	C/BE[6]#	GND	
3	GND	CLK4	GND	GNT3#	REQ4#	GNT4#	GND	
2	GND	CLK2	CLK3	SYSEN#	GNT2#	REQ3#	GND	
1	GND	CLK1	GND	REQ1#	GNT1#	REQ2#	GND	
25	GND	5V	REQ64#	ENUM#	3.3V	5V	GND	
24	GND	AD[1]	5V	V(I/O)	AD[0]	ACK64#	GND	
23	GND	3.3V	AD[4]	AD[3]	5V	AD[2]	GND	
22	GND	AD[7]	GND	3.3V	AD[6]	AD[5]	GND	
21	GND	3.3V	AD[9]	AD[8]	M66EN	C/BE[0]#	GND	
20	GND	AD[12]	GND	V(I/O)	AD[11]	AD[10]	GND	
19	GND	3.3V	AD[15]	AD[14]	GND	AD[13]	GND	P1
18	GND	SERR#	GND	3.3V	PAR	C/BE[1]#	GND	/
17	GND	3.3V	SDONE	SBO#	GND	PERR#	GND	J1
16	GND	DEVSEL#	GND	V(I/O)	STOP#	LOCK#	GND	
15	GND	3.3V	FRAME#	IRDY#	GND	TRDY#	GND	
12–14	Key Area							C
11	GND	AD[18]	AD[17]	AD[16]	GND	C/BE[2]#	GND	O
10	GND	AD[21]	GND	3.3V	AD[20]	AD[19]	GND	N
9	GND	C/BE[3]#	IDSEL	AD[23]	GND	AD[22]	GND	N
8	GND	AD[26]	GND	V(I/O)	AD[25]	AD[24]	GND	E
7	GND	AD[30]	AD[29]	AD[28]	GND	AD[27]	GND	C
6	GND	REQ#	GND	3.3V	CLK	AD[31]	GND	T
5	GND	BRSVP1A5	BRSVP1B5	RST#	GND	GNT#	GND	O
4	GND	BRSVP1A4	GND	V(I/O)	INTP	INTS	GND	R
3	GND	INTA#	INTB#	INTC#	5V	INTD#	GND	
2	GND	TCK	5V	TMS	TDO	TDI	GND	
1	GND	5V	-12V	TRST#	+12V	5V	GND	
Pin	**Z**	**A**	**B**	**C**	**D**	**E**	**F**	

4.2.3. Star Trigger Slot

Table 4.11 gives the peripheral slot pinout for the J1 and J2 connectors of the star trigger slot. PXI-specific signals are shown in **bold.**

RULE: Star trigger modules and backplane star trigger slots SHALL use the pinout in Table 4.11.

Table 4.11 Star Trigger Slot Pinout

22	GND	**PXI_RSVA22**	**PXI_RSVB22**	**PXI_RSVC22**	**PXI_RSVD22**	**PXI_RSVE22**	GND	
21	GND	**PXI_LBR0**	GND	**PXI_LBR1**	**PXI_LBR2**	**PXI_LBR3**	GND	
20	GND	**PXI_LBR4**	**PXI_LBR5**	**PXI_STAR0**	GND	**PXI_STAR1**	GND	
19	GND	**PXI_STAR2**	GND	**PXI_STAR3**	**PXI_STAR4**	**PXI_STAR5**	GND	
18	GND	**PXI_TRIG3**	**PXI_TRIG4**	**PXI_TRIG5**	GND	**PXI_TRIG6**	GND	P2
17	GND	**PXI_TRIG2**	GND	PRST#	**PXI_CLK10_IN**	**PXI_CLK10**	GND	/
16	GND	**PXI_TRIG1**	**PXI_TRIG0**	DEG#	GND	**PXI_TRIG7**	GND	J2
15	GND	**PXI_BRSVA15**	GND	FAL#	**PXI_STAR6**	**PXI_LBR6**	GND	
14	GND	AD[35]	AD[34]	AD[33]	GND	AD[32]	GND	
13	GND	AD[38]	GND	V(I/O)	AD[37]	AD[36]	GND	C
12	GND	AD[42]	AD[41]	AD[40]	GND	AD[39]	GND	O
11	GND	AD[45]	GND	V(I/O)	AD[44]	AD[43]	GND	N
10	GND	AD[49]	AD[48]	AD[47]	GND	AD[46]	GND	N
9	GND	AD[52]	GND	V(I/O)	AD[51]	AD[50]	GND	E
8	GND	AD[56]	AD[55]	AD[54]	GND	AD[53]	GND	C
7	GND	AD[59]	GND	V(I/O)	AD[58]	AD[57]	GND	T
6	GND	AD[63]	AD[62]	AD[61]	GND	AD[60]	GND	O
5	GND	C/BE[5]#	GND	V(I/O)	C/BE[4]#	PAR64	GND	R
4	GND	V(I/O)	**PXI_BRSVB4**	C/BE[7]#	GND	C/BE[6]#	GND	
3	GND	**PXI_LBR7**	GND	**PXI_LBR8**	**PXI_LBR9**	**PXI_LBR10**	GND	
2	GND	**PXI_LBR11**	**PXI_LBR12**	SYSEN#	**PXI_STAR7**	**PXI_STAR8**	GND	
1	GND	**PXI_STAR9**	GND	**PXI_STAR10**	**PXI_STAR11**	**PXI_STAR12**	GND	
25	GND	5V	REQ64#	ENUM#	3.3V	5V	GND	
24	GND	AD[1]	5V	V(I/O)	AD[0]	ACK64#	GND	
23	GND	3.3V	AD[4]	AD[3]	5V	AD[2]	GND	
22	GND	AD[7]	GND	3.3V	AD[6]	AD[5]	GND	
21	GND	3.3V	AD[9]	AD[8]	M66EN	C/BE[0]#	GND	
20	GND	AD[12]	GND	V(I/O)	AD[11]	AD[10]	GND	
19	GND	3.3V	AD[15]	AD[14]	GND	AD[13]	GND	P1
18	GND	SERR#	GND	3.3V	PAR	C/BE[1]#	GND	/
17	GND	3.3V	SDONE	SBO#	GND	PERR#	GND	J1
16	GND	DEVSEL#	GND	V(I/O)	STOP#	LOCK#	GND	
15	GND	3.3V	FRAME#	IRDY#	GND	TRDY#	GND	
12–14	Key Area							C
11	GND	AD[18]	AD[17]	AD[16]	GND	C/BE[2]#	GND	O
10	GND	AD[21]	GND	3.3V	AD[20]	AD[19]	GND	N
9	GND	C/BE[3]#	IDSEL	AD[23]	GND	AD[22]	GND	N
8	GND	AD[26]	GND	V(I/O)	AD[25]	AD[24]	GND	E
7	GND	AD[30]	AD[29]	AD[28]	GND	AD[27]	GND	C
6	GND	REQ#	GND	3.3V	CLK	AD[31]	GND	T
5	GND	BRSVP1A5	BRSVP1B5	RST#	GND	GNT#	GND	O
4	GND	BRSVP1A4	GND	V(I/O)	INTP	INTS	GND	R
3	GND	INTA#	INTB#	INTC#	5V	INTD#	GND	
2	GND	TCK	5V	TMS	TDO	TDI	GND	
1	GND	5V	-12V	TRST#	+12V	5V	GND	
Pin	**Z**	**A**	**B**	**C**	**D**	**E**	**F**	

4.3. Chassis Power Supply Specifications

Because minimum power supply requirements are specified for PXI chassis, module designers can design modules knowing that they will work in any PXI system.

RULE: The power supply SHALL provide at least the required amounts of current given in Tables 4.12 and 4.13.

RECOMMENDATION: The power supply SHOULD provide at least the recommended amounts of current in Tables 4.12 and 4.13.

Table 4.12 5 V Backplane Power Requirements and Recommendations

	5 V		3.3 V		+12 V	-12 V
	System Slot	**Peripheral Slot**	**System Slot**	**Peripheral Slot**	**All Slots**	**All Slots**
Required Current	4 A	2 A	0 A	0 A	0.5 A	0.1 A
Recommended Current	6 A	2 A	6 A	0 A	0.5 A	0.1 A

Table 4.13 3.3 V Backplane Power Requirements and Recommendations

	5 V		3.3 V		+12 V	-12 V
	System Slot	**Peripheral Slot**	**System Slot**	**Peripheral Slot**	**All Slots**	**All Slots**
Required Current	0.5 A	0.5 A	6 A	3 A	0.5 A	0.1 A
Recommended Current	6 A	2 A	6 A	3 A	0.5 A	0.1 A

OBSERVATION: By supplying the recommended amounts of power, a PXI system can accommodate a much wider range of system slot designs.

OBSERVATION: Having the system supply 3.3 V conserves space and heat in the system slot module because 5 V does not have to be converted to 3.3 V on the module.

PERMISSION: Power supplies MAY provide additional current.

OBSERVATION: Each generation of processors requires more power than the previous generation. Providing copious amounts of power and cooling to the system slot of a chassis can extend the product applicability in the future.

RULE: All modules that draw more current than the amount required of the power supply SHALL document and make available to the customer their current requirements.

RECOMMENDATION: All modules SHOULD document and make available to the customer their current requirements, to allow users to maximize the capabilities of their power supply.

RULE: A peripheral or system slot module SHALL NOT draw more than 1 A of current from any power pin or return more than 1 A of current through any ground pin.

5. Software Frameworks and Requirements

This section discusses the software features associated with a PXI system. It gives an overview of the general motivating factors behind the PXI software specification, along with specific software frameworks.

5.1. Overview

Like other bus architectures before it, PXI defines standards that allow products from multiple vendors to work together at the bus level. The PXI specification goes on to mandate software requirements in addition to these bus level requirements. Other buses that have failed to designate software standards have seen the market fragment into competing standards from multiple vendors.

The PXI hardware specification is motivated by the benefits achieved by leveraging existing desktop hardware and software technology into the instrumentation world.

5.2. Motivation

Low-cost, rugged, reliable computer systems are needed in instrumentation and automation applications. The demands of reducing product cost and time to market, while increasing reliability, are severely straining custom-built systems. Hardware vendors have implemented many modular multivendor solutions to tackle these problems. However, the majority of costs associated with any system development are more likely related to the software development time, rather than the hardware development time.

PXI, unlike other systems, defines *software frameworks* as part of its specification. These software frameworks ensure that a user has a complete, multivendor system solution from the start.

The technical forces driving the PXI software frameworks are the same forces that are driving the hardware standard—the immense amounts of capital being applied toward the development of desktop PCs. By aligning with these software developments, PXI systems are poised to initiate a new generation of instrumentation. The frameworks that have been chosen for this initial specification reflect the dominance of these operating systems on current desktop PCs. As other operating systems become widely accepted and offer the same degree of software leverage as the current frameworks, they may be added to the supported PXI frameworks. Each framework is required to support the VISA software standard. VISA provides an industry-standard mechanism for locating, configuring, and controlling PXI modules.

The currently supported frameworks are the Windows 95 and Windows NT frameworks. Many organizations have aligned their entire offices around Microsoft operating systems, because of the reduction in training and support costs that come

from a common, familiar computing environment across the office. These same benefits extend to the factory floor or test department.

Thousands of 32-bit applications are running today on the Windows 95 and Windows NT platforms, ranging from technical and engineering applications to complete manufacturing and financial solutions. All of these tools can be leveraged to improve instrumentation systems.

The ever-increasing interconnection between computers that design, produce, and test goods is driving the requirement that these systems easily interact with each other. The use of a common operating system across these computers means that a richer set of tools is available for exchanging and sharing data. All of the work being done to interconnect the desktop—ActiveX, COM, ODBC, and so on—is now available to interconnect machines on the factory floor to each other and to the rest of the corporation.

5.3. Framework Definition

The software frameworks define PXI system software requirements for both system controller modules and PXI peripheral modules. System controller modules and PXI peripheral modules have to meet certain requirements for operating system and tool support in order to be considered compliant with a given PXI software framework.

RULE: PXI system controller modules and PXI peripheral modules SHALL support at least one of the following frameworks (Windows 95 or Windows NT).

5.4. Windows 95 System Framework

5.4.1. Introduction

This section defines the specific requirements for the Windows 95 (WIN95) system framework. It defines all of the unique components that must exist to support this framework. It also describes the optional recommended components.

5.4.2. Overview of the Framework

The WIN95 system framework defines a system based on the popular IBM PC personal computer architecture and its compatibles and is based on the Windows 95 operating system from Microsoft. The design requirements for Windows 95 are specified in the *Hardware Design Guide for Microsoft Windows 95*, hereafter referred to as PC 95.

5.4.2.1. Controller Requirements

This section defines the system requirements for the WIN95 framework for the system controller module.

RECOMMENDATION: If a system controller module claims conformance to the PXI WIN95 system framework, it SHOULD meet the PC 95 design requirements.

RULE: As the requirements for compliance with the Microsoft PC 9*x* specifications evolve, PXI system controller modules SHALL conform to these new requirements.

PERMISSION: System controller modules that are certified prior to future changes in the PC 9*x* specifications MAY still be considered PXI-compliant.

RULE: If a system controller module claims conformance to the PXI WIN95 system framework, it SHALL ship with a VISA implementation that supports the PXI bus.

5.4.2.2. PXI Peripheral Module Requirements

Hardware vendors for other industrial buses that do not have software standards often do not provide any software drivers for their modules. The customer is often given only a manual describing how to write software to control the module. The cost to the customer, in terms of engineering effort to support these modules, is huge. PXI removes this burden by requiring that manufacturers, rather than customers, develop the driver software.

RULE: If a peripheral module claims conformance to the PXI WIN95 system framework, it SHALL meet the PC 95 design requirements.

RECOMMENDATION: PXI peripheral modules that are instrumentation class modules SHOULD provide a user-level interface that is supported under the development environments specified in Table 5.1.

Table 5.1 Development Environments Supported by PXI Modules under Windows 95

Product	Company	Revision
LabVIEW	National Instruments	4.0
LabWindows/CVI	National Instruments	4.0
Visual Basic	Microsoft	5.0
Visual C/C++	Microsoft	5.0
Turbo C/C++	Borland	5.0

OBSERVATION: Other system tools MAY be supported in addition to these tools.

5.5. Windows NT System Framework

5.5.1. Introduction

This section defines the specific requirements for the Windows NT (WINNT) system framework. It defines all of the unique components that must exist to support this framework. It also describes the optional recommended components.

5.5.2. Overview of the Framework

The WINNT system framework defines a system based on the popular IBM PC personal computer architecture and its compatibles, and is based on the Windows NT operating system from Microsoft. Microsoft does not have a comprehensive design document for the Windows NT operating system as it does for Windows 95. However, the Microsoft *PC 97 Hardware Design Guide* does provide guidelines for future versions of Windows NT.

5.5.3. Controller Requirements

This section defines the system requirements for the WINNT framework for the system controller module.

RULE: If a system controller module claims conformance to the PXI WINNT system framework, it SHALL run Windows NT version 4.0 or greater.

RULE: The system controller module SHALL be based on the 80x86 architecture.

OBSERVATION: Processors other than the 80x86 that are supported under Windows NT may be added in additional frameworks.

RULE: If a system controller module claims conformance to the PXI WINNT system framework, it SHALL ship with a VISA implementation that supports the PXI bus.

5.5.3.1. PXI Peripheral Module Requirements

Hardware vendors for other industrial buses that do not have software standards often do not provide any software drivers for their modules. The customer is often given only a manual describing how to write software to control the module. The cost to the customer, in terms of engineering effort to support these modules, is huge. PXI removes this burden by requiring that manufacturers, rather than customers, develop this software.

RULE: If a peripheral module claims conformance to the PXI WINNT system framework, it SHALL provide software for installing, configuring, and controlling this module under Windows NT.

Because Microsoft has not released comprehensive design guidelines for Windows NT, the preceding rule is not as strict as the corresponding rule for the PXI WIN95 framework. The intention of this rule is to provide the same degree of integration and ease of use in Windows NT-based PXI systems as is required in Windows 95-based PXI systems. Failure to provide the appropriate software support in the Windows NT framework will result in an unacceptable user experience.

RECOMMENDATION: PXI peripheral modules that are instrumentation class modules SHOULD provide a user-level interface that is supported under the development environments specified in Table 5.2.

Table 5.2 Development Environments Supported by PXI Modules under Windows NT

Product	Company	Revision
LabVIEW	National Instruments	4.0
LabWindows/CVI	National Instruments	4.0
Visual Basic	Microsoft	5.0
Visual C/C++	Microsoft	5.0
Turbo C/C++	Borland	5.0

OBSERVATION: Other system tools MAY be supported in addition to these tools.

5.6. Support for Existing Instrumentation Standards

The challenge for developing PXI instrumentation systems is to provide a platform that extends the capabilities of new test systems, while supporting existing instrumentation standards. The VXI*plug&play* System Alliance provides an industry-standard mechanism for communicating with GPIB, VXI, and serial instrumentation through a series of specifications. These specifications define communications standards (*VISA*), instrument programming interfaces (*instrument drivers*), and interactive user interfaces (*soft front panels*). These specifications have also been grouped into frameworks that include frameworks for Windows 95 and Windows NT.

RULE: A PXI module that controls a VISA supported interface SHALL provide the VISA software as a mechanism for communicating with that interface.

The use of the VISA standard in PXI preserves the user's investment in existing instrumentation software. VISA provides the link from PXI to a VXI chassis and instruments and standalone GPIB and serial instruments.

RECOMMENDATION: Instrumentation class PXI peripheral modules SHOULD provide instrument drivers and soft front panels that are consistent with the VXI*plug&play* instrument driver specifications (VPP-3.*x* and VPP-7).

The goal of supporting VXI*plug&play* instrument drivers and soft front panels is to provide a familiar development environment to test and measurement customers. Test and measurement customers have come to expect VXI and GPIB instruments to have soft front panels and instrument drivers. VXI*plug&play* support for native PXI instruments provides a seamless software path between VXI and PXI based systems.

RULE: All PXI system controller modules SHALL be supplied with VISA.

VISA will be used as the software interface for system configuration, slot enumeration, and access to instrumentation features including the trigger bus and the star trigger.

5.7. System Implementation Issues

5.7.1. System Configuration and Initialization Files

The PXI specification allows many variations of PXI chassis and system modules. To assist system integrators, the manufacturers of PXI chassis and system modules must document the capabilities of their products. The minimum documentation requirements are contained in `.ini` files, which consist of ASCII text. The `.ini` files can be read by the system integrator, and can also be used by configuration utilities and device drivers if needed.

The `.ini` files help the integrator determine if a peripheral module installed in a particular slot may enable circuitry that uses a local bus. For example, in a chassis, the rightmost peripheral slot can use its local bus right signals to implement an external interface to another bus (such as SCXI). If a peripheral module that uses its local bus right for another purpose is installed in that slot, the local bus circuitry cannot be enabled.

The `.ini` files also help the integrator determine the physical location of peripheral modules. For example, in a system with four identical data acquisition modules installed, the device driver for the module identifies the modules as four logical entities. The user may need to know the physical slot location of logical module number 2. Utility software can use the `.ini` files to determine the slot location. The `.ini` files effectively provide a method of obtaining slot enumeration without requiring additional pins on each slot.

The documentation for a chassis is contained in a `chassis.ini` file. The information in this file is combined with information about the system module into a single system initialization file, `pxisys.ini`. The system module manufacturer will either provide a `pxisys.ini` file for the particular chassis model that contains the system module, or provide a utility that can read an arbitrary `chassis.ini` file and generate the corresponding `pxisys.ini` file.

Device drivers and other utility software will read the `pxisys.ini` (PXI System Initialization) file to determine information about the system. The device drivers should have no need to directly read the `chassis.ini` file.

OBSERVATION: Issues related to configuring systems consisting of multiple chassis are beyond the scope of this specification. However, the `.ini` files described in this section may be useful in addressing such issues.

5.7.2. General .ini File Format

The `.ini` files consist of lines of ASCII text. Each `.ini` file contains one or more sections, and each section contains one or more tag lines. Each tag line describes a specific property of the section.

A `.ini` file might begin with the following lines:

```
# This line is a comment
[Section1]
IsSpecialSection = No
[Section2]
IsSpecialSection = Yes
```

RULE: Each `.ini` file SHALL contain only ASCII text.

RULE: Each `.ini` file SHALL contain only the following types of lines: comment line, section header, and tag line.

RULE: A comment line SHALL begin with the '#' character.

RULE: A section header line SHALL begin with the '[' character and end with the ']' character. Text between the two brackets SHALL identify the type of section.

RULE: A tag line SHALL consist of the following three fields: tag, '=' character, and value. The three fields SHALL be each separated by a single space character.

5.7.3. PXI System Initialization (pxisys.ini) File

A `pxisys.ini` file describes the PXI system configuration. In this context, a system configuration is a particular system module model installed in a particular chassis model. The `pxisys.ini` file contains information that

describes the physical location of installed peripheral modules. The file also describes the capabilities of the peripheral slots in the chassis.

RULE: A system module manufacturer SHALL either provide a `pxisys.ini` file for each supported system configuration or provide a utility that can generate the `pxisys.ini` file.

OBSERVATION: A utility to create a `pxisys.ini` file would read the `.ini` file for the chassis and the `.ini` file for the system module and generate a `pxisys.ini` file.

OBSERVATION: The `pxisys.ini` file is usually read only by PXI peripheral device drivers.

RULE: A `pxisys.ini` file SHALL have one section for each slot in the system. The section heading line for each section SHALL be '[Slot*n*]', where *n* is a decimal integer such that $0 \leq n \leq 999$.

RULE: Each physical slot SHALL be labeled so that the user can see the slot number, and this slot number SHALL be used in the heading line for the slot.

RULE: Each backplane PCI device SHALL be assigned a unique slot number to identify it in the `.ini` file.

RULE: Each section SHALL contain one of each tag line type described in Table 5.3.

OBSERVATION: A backplane PCI-PCI bridge module will have two Slot entries in the `.ini` file—one for the peripheral PCI module on the upstream bus, and one for the system PCI module on the downstream bus.

Table 5.3 Tag Line Descriptions

Tag	Valid Values	Description
`IDSEL`	*n*, where *n* is the number of the PCI address line that connects to the slot's IDSEL pin.	This tag shows the backplane IDSEL connection to this slot.
`SecondaryBusNumber`	*n*, where *n* is a decimal integer such that $0 \leq n \leq 255$	If this slot is a PCI-PCI bridge, this is the PCI bus number on the secondary interface of the bridge as defined by the Configuration Cycle section of the PCI specification. If this slot is not a bridge, set the field to `0`.

(continues)

Table 5.3 Tag Line Descriptions (Continued)

Tag	Valid Values	Description
`ExternalBackplaneInterface`	None (other)	If this slot routes to an external backplane interface, this tag specifies the name of that interface.
`PCIBusNumber`	*n*, where *n* is a decimal integer such that $0 \le n \le 255$	The PCI bus number as defined by the Configuration Cycle section of the PCI spec. The `PCIBusNumber` and `PCIDeviceNumber` uniquely identify a module in a system.
`PCIDeviceNumber`	*n*, where *n* is a decimal integer such that $0 \le n \le 31$	The PCI device number as defined by the Configuration Cycle section of the PCI specification. The `PCIBusNumber` and `PCIDeviceNumber` uniquely identify a module in a system.

RULE: The `PCIBusNumber` and `SecondaryBusNumber` fields SHALL contain the PCI bus number as defined by the Configuration Cycle section of the PCI specification when the only PCI-PCI bridges present are those on the controller and the chassis.

OBSERVATION: If modules with bridges are installed in the chassis, config space registers on the modules can be queried to determine the PCI bus number shifting caused by the additional bridges.

5.7.4. Chassis Initialization (chassis.ini) File

A `chassis.ini` file describes the functionality of a PXI chassis. A PXI chassis may include slots that accept peripheral modules, slots that accept system modules, and backplane PCI modules that behave as system modules or peripheral modules.

RULE: A chassis manufacturer SHALL provide a `chassis.ini` file for each chassis model produced.

RULE: A `chassis.ini` file SHALL have one section for each slot in the chassis. The section heading line for each section SHALL be '[Slot*n*]', where *n* is a decimal integer such that $0 \le n \le 999$.

RULE: Each physical slot SHALL be labeled so that the user can see the slot number, and this slot number SHALL be used in the heading line for the slot.

RULE: Each backplane PCI device SHALL be assigned a unique slot number to identify it in the `.ini` file.

RULE: Each section SHALL contain one IDSEL tag line, as described in Table 5.3.

RULE: A `chassis.ini` file SHALL also contain the tag lines shown in Table 5.4 in each section:

Table 5.4 Additional Tag Line Descriptions for chassis.ini

Tag	Valid Values	Description
`SlotNumberOfOtherHalfOfBridge`	None *n*, where *n* is a `SlotNumber`	If this slot is part of a PCI-PCI bridge, this tag field shows the `SlotNumber` of the other PCI module contained in this PCI-PCI bridge.
`SystemSlotNumber`	*n*, where *n* is the `SlotNumber` of the system slot.	This tag shows the `SlotNumber` of the system slot for this PCI bus segment.

5.7.5. Other Chassis Requirements

To assist configuration software in locating the slots, this specification places restrictions on the numbering of system slots.

RULE: For any two physical system slots in the same row, the system slot physically to the left SHALL have a lower slot number. Any system slot SHALL have a `SlotNumber` lower than all system slots in a rows physically below the slot.