NAT7210™
Reference Manual

June 1995 Edition

Part Number 370875A-01

National Instruments Corporate Headquarters
6504 Bridge Point Parkway
Austin, TX 78730-5039
(512) 794-0100
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices:
Australia 03 9 879 9422, Austria 0662 45 79 90 0, Belgium 02 757 00 20,
Canada (Ontario) 519 622 9310, Canada (Québec) 514 694 8521,
Denmark 45 76 26 00, Finland 90 527 2321, France 1 48 14 24 24,
Germany 089 741 31 30, Hong Kong 2645 3186, Italy 02 48301892,
Japan 03 5472 2970, Korea 02 596 7456, Mexico 5 202 2544,
Netherlands 03480 33466, Norway 32 84 84 00, Singapore 2265886,
Spain 91 640 0085, Sweden 08 730 49 70, Switzerland 056 20 51 51,
Taiwan 02 377 1200, U.K. 01635 523545

Limited Warranty

The NAT7210™ integrated circuit ("equipment") is warranted against defects in material and workmanship under normal use and service for a period of one (1) year from the date of shipment from the National Instruments factory. During this period of one year, National Instruments shall at its sole option either repair, replace, or credit the Buyer for defective equipment if: (i) Buyer returns the equipment to National Instruments, FOB the National Instruments factory in Austin, Texas; (ii) Buyer notifies National Instruments promptly upon discovery of any defect in writing, including a detailed description of the defect; and (iii) upon examination of the returned equipment, National Instruments is satisfied that the circuit is defective and that the cause of such defect is not alteration or repair by someone other than National Instruments, neglect, accident, misuse, improper installation, or use contrary to any instructions issued by National Instruments.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. Prior to issuance of an RMA by National Instruments, Buyer shall allow National Instruments the opportunity to inspect the equipment on-site at Buyer's facility.

This warranty expires one year from date of original shipment regardless of any warranty performance during that warranty period. The warranty provided herein is Buyer's sole and exclusive remedy for nonconformity of the equipment or for breach of any warranty. THE ABOVE IS IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED. NATIONAL INSTRUMENTS SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. BUYER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE BUYER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. National Instruments recommends against the use of its products as critical components in any life support devices or systems whose failure to perform can reasonably be expected to cause significant injury to a human. Buyer assumes all risk for such application and agrees to indemnify National Instruments for all damages which may be incurred due to use of the National Instruments standard devices in medical or life support applications. Any action against National Instruments must be brought within one year after the cause of action accrues.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

Copyright

Trademarks

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

Chapter 5
Software Considerations 5-1

Figures

Tables

About This Manual

This manual describes the programmable features of the NAT7210 and contains information that is suitable for programmers and engineers who wish to write software for the NAT7210.

This manual assumes that you are already familiar with general IEEE 488 concepts.

Organization of This Manual

This manual is organized as follows:

- Chapter 1, *Introduction and General Description*, explains the features and capabilities of the NAT7210.
- Chapter 2, *NAT7210 Architecture*, discusses the internal hardware architecture of the NAT7210.
- Chapter 3, *7210-Mode Interface Registers*, contains NAT7210 address maps and detailed descriptions of the NAT7210 interface registers in 7210 mode.
- Chapter 4, *9914-Mode Interface Registers*, contains NAT7210 address maps and detailed descriptions of the NAT7210 interface registers in 9914 mode.
- Chapter 5, *Software Considerations*, explains important NAT7210 programming considerations, including chip initialization, Talkers and Listeners, message reception, and holdoffs.
- Chapter 6, *Controller Software Considerations*, explains important system and GPIB Controller considerations.
- Chapter 7, *Hardware Considerations*, explains important NAT7210 hardware-interfacing considerations, including a description of the pins.
- Appendix A, *Common Questions*, lists common questions and answers.
- Appendix B, *Introduction to the GPIB*, discusses the history of the GPIB, GPIB hardware configurations, and serial polling.
- Appendix C, *Standard Commands for Programmable Instruments (SCPI)*, discusses the SCPI document, the required SCPI commands, and SCPI programming.
- Appendix D, *Multiline Interface Command Messages*, lists the multiline interface messages and describes the mnemonics and messages that correspond to the interface functions.

- Appendix E, *Mnemonics Key*, defines the mnemonics (abbreviations) that this manual uses for functions, remote messages, local messages, states, bits, registers, integrated circuits, and system functions.
- Appendix F, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products and manuals.
- The *Glossary* contains an alphabetical list and a description of the terms, including abbreviations, acronyms, metric prefixes, mnemonics, and symbols, that this manual uses.
- The *Index* contains an alphabetical list of the key terms and topics that this manual uses, and it includes the page number where you can locate each term and topic.

Conventions Used in This Manual

This manual uses the following conventions.

italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
bold italic	Bold italic text denotes a note, caution, or warning.
`monospace`	Text in this font denotes programming examples.
IEEE 488 and IEEE 488.2	IEEE 488 and IEEE 488.2 refer to the ANSI/IEEE Standard 488.1-1987 and ANSI/IEEE Standard 488.2-1992, respectively, which define the GPIB.

The *Glossary* lists abbreviations, acronyms, metric prefixes, mnemonics, symbols, and terms.

Related Documentation

The following documents contain information that you may find helpful as you read this manual.

- *40-Pin IEEE 488.2 Controller Chip: Drop-In Replacement for NEC µPD7210 NAT7210APD*
- ANSI/IEEE Standard 488.1-1987, *IEEE Standard Digital Interface for Programmable Instrumentation*
- ANSI/IEEE Standard 488.2-1992, *IEEE Standard Codes, Formats, Protocols, and Common Commands*

You may obtain the two ANSI/IEEE documents through the Institute of Electrical and Electronics Engineers, 345 East 47th Street, New York, New York 10017.

You may obtain more information about Standard Commands for Programmable Instruments from the SCPI Consortium, 8380 Hercules Drive, Suite P3, La Mesa, CA 91942.

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix F, *Customer Communication*, at the end of this manual.

Chapter 1
Introduction and General Description

This chapter explains the features and capabilities of the NAT7210.

The NAT7210 is an IEEE 488.2 Controller chip designed to perform all the interface functions defined in the ANSI/IEEE Standard 488.1-1987 and the additional requirements and recommendations of the ANSI/IEEE Standard 488.2-1987. The NAT7210 manages the IEEE 488 interface functions with a set of control and status registers that increase the throughput of driver software and simplify hardware and software design. The NAT7210 performs complete IEEE 488 Talker, Listener, and Controller functions and is software compatible with the NEC μPD7210 and TI TMS9914A chips.

The NAT7210 can be characterized as a bus translator: it converts messages and signals from the CPU into appropriate GPIB messages and signals. In GPIB terminology, the NAT7210 implements GPIB board and device functions to communicate with the central processor and memory. For the computer, the NAT7210 is an interface to the outside world.

IEEE 488 Capabilities

The National Instruments NAT7210 has the features necessary to provide a high-performance IEEE 488 interface. Table 1-1 lists the capabilities of the NAT7210 in terms of the IEEE 488 standard codes.

Table 1-1. NAT7210 IEEE 488 Interface Capabilities

Capability Code	Description
SH1	Complete Source Handshake Capability
AH1	Complete Acceptor Handshake Capability; DAC and RFD Holdoff on Certain Events
T5	Complete Talker Capability: • Basic Talker • Serial Poll • Talk-Only Mode • Unaddressed on MLA • Send END or EOS

(continues)

Table 1-1. NAT7210 IEEE 488 Interface Capabilities (Continued)

Capability Code	Description
TE5	Complete Extended Talker Capability: • Basic Extended Talker • Serial Poll • Talk-Only Mode • Unaddressed on MSA & LPAS • Send END or EOS
L3	Complete Listener Capability: • Basic Listener • Listen-Only Mode • Unaddressed on MTA • Detect END or EOS
LE3	Complete Extended Listener Capability: • Basic Extended Listener • Listen-Only Mode • Unaddressed on MSA & TPAS • Detect END or EOS
SR1	Complete Service Request Capability
RL1	Complete Remote/Local Capability
PP1	Remote Parallel Poll Configuration
PP2	Local Parallel Poll Configuration
DC1	Complete Device Clear Capability
DT1	Complete Device Trigger Capability
C1 through C5	Complete Controller Capability: • System Controller • Send IFC and Take Charge • Send REN • Respond to SRQ • Send Interface Messages • Received Control • Parallel Poll • Take Control Synchronously or Asynchronously
E2	Three-State Drivers (Open-Collector Drivers During Parallel Polls)

The NAT7210 has complete Source and Acceptor Handshake capability. It can operate as a basic Talker or an extended Talker and can respond to a Serial Poll. If you place it in talk-only mode, it is unaddressed to talk when it receives its listen address. The NAT7210 GPIB interface can also operate as a basic Listener or an extended Listener. If you place it in listen-only mode, it is unaddressed to listen when it receives its talk address. The NAT7210 can request service from a Controller.

Device Clear and Trigger capability is included in the interface; the interpretation is software dependent.

Other GPIB features include the following:

- Messages not sent when there are no Listeners
- Automatic detection of EOS and/or NL messages
- Automatic bus synchronization detection
- Programmable data transfer rates (T1 delays as short as 350 ns)
- Programmable GPIB transceiver support
- Automatic processing of IEEE 488 commands and read-undefined commands
- Ability to use six addressing modes:
  - Automatic single or dual primary addressing detection
  - Automatic single primary with single secondary address detection
  - Single or dual primary with multiple secondary addressing
  - Multiple primary addressing

CPU Interface Capabilities

- Software compatible with NEC μPD7210 and TI TMS9914A Controller chips
- DMA interface to the host system
- Flexible interrupt capabilities
- Uses only eight bytes of address space

Typical System Interface

Figure 1-1 shows a block diagram of a typical application that uses the NAT7210 to implement an IEEE 488.2 interface.

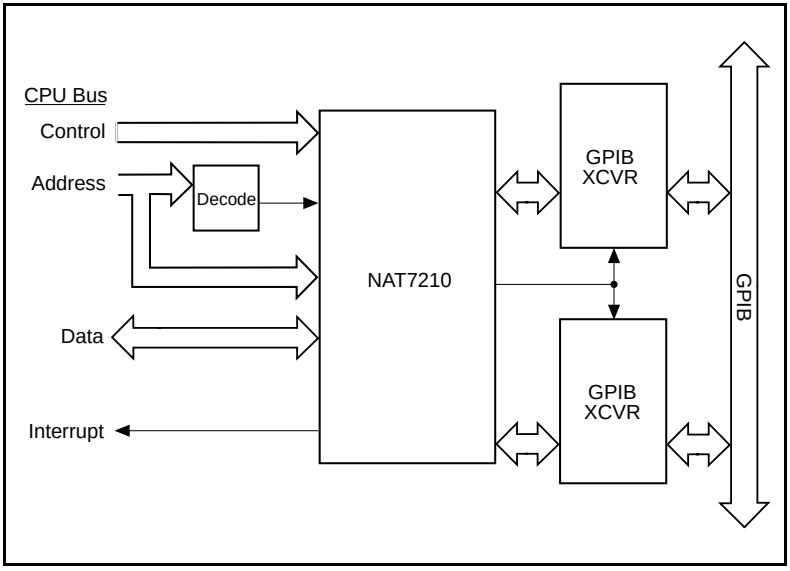


Figure 1-1. NAT7210 Implementation Block Diagram

In all applications, the NAT7210 must be connected to the GPIB via IEEE 488 compliant transceivers such as the 75160 and 75162, which are available from National Semiconductor and other vendors. The NAT7210 has control signals that let it easily interface to several different types of transceivers.

Chapter 2
NAT7210 Architecture

This chapter discusses the internal hardware architecture of the NAT7210.

The NAT7210 includes the following major components:

- *Read/Write Control* converts the CPU interface signals to read and write signals for each internal NAT7210 register.

- *Internal NAT7210 Registers* configure and control the operation of the NAT7210. They transfer data between the NAT7210 and the GPIB, report status information, and set the operating modes. Chapter 3, *7210-Mode Interface Registers*, and Chapter 4, *9914-Mode Interface Registers*, describe each register in detail.

- *Interface Functions* implement the interface functions described in the IEEE 488.1 standard. Some internal registers control the interface functions, and you can use other internal registers to monitor the status of interface functions. The interface functions drive and receive the GPIB control signals and generate the signals to control the GPIB transceivers.

- *Message Decoders* receive the GPIB data lines and decode the GPIB commands that affect the operation of the interface functions.

Figure 2-1 contains a block diagram of the NAT7210.

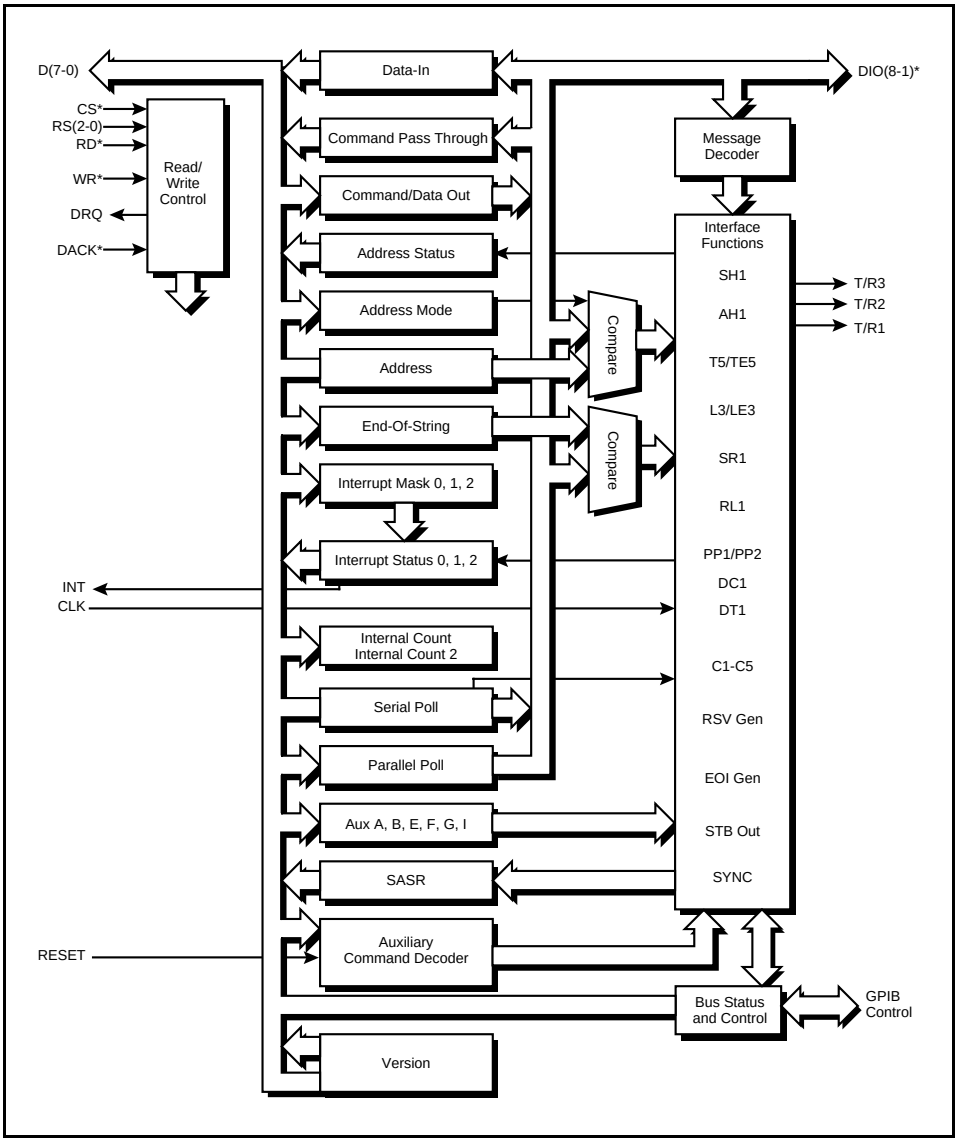


Figure 2-1. NAT7210 Block Diagram

NAT7210 Modes

The NAT7210 has two basic modes of operation: 7210 mode and 9914 mode. In 7210 mode, the NAT7210 is software compatible with the µPD7210 IEEE 488 Controller. The NAT7210 has many registers and features that are not present in the µPD7210. In 9914 mode, the NAT7210 is software compatible with the TMS9914A IEEE 488 Controller. The NAT7210 has many registers and features that are not present in the TMS9914A.

Note: ***Throughout this manual, 7210 mode refers to the NEC µPD7210 software compatibility mode, and 9914 mode refers to the TI TMS9914A software compatibility mode.***

Changing the NAT7210 Mode

Figure 2-2 illustrates how you change the mode of the NAT7210.

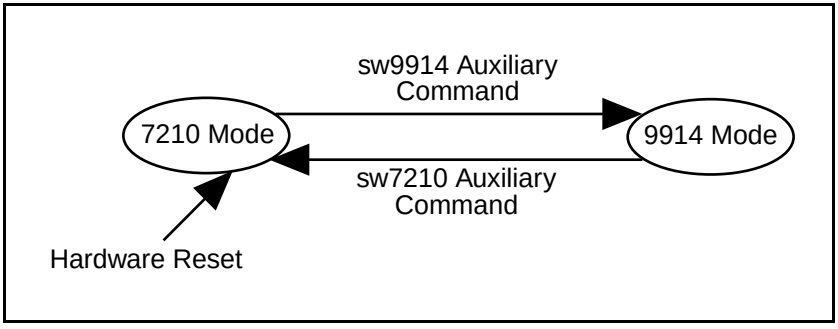


Figure 2-2. Changing the NAT7210 Mode

Notice that the NAT7210 is in 7210 mode after a hardware reset. To change from 7210 mode to 9914 mode, write the sw9914 auxiliary command to the (7210 mode) Auxiliary Mode Register (AUXMR). To change from 9914 mode to 7210 mode, write the sw7210 auxiliary command to the (9914 mode) Auxiliary Command Register (AUXCR).

Chapter 3
7210-Mode Interface Registers

This chapter contains NAT7210 address maps and detailed descriptions of the NAT7210 interface registers in 7210 mode. For 9914-mode register descriptions, see Chapter 4, *9914-Mode Interface Registers*.

7210 Register Map

Table 3-1 is the register bit map for the NAT7210 in 7210 mode.

Notice that bold-ruled cells distinguish seven registers that are accessible only when the Page-In state is true. Refer to *The Page-In State* section that immediately follows the register map for more information.

Table 3-1. 7210-Mode Register Map

Key

= 7210-Mode Paged Registers
R = Read Register
W = Write Register

		7	6	5	4	3	2	1	0	
DIR	+0	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	R
CDOR	+0	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	W
ISR1	+1	CPT	APT	DET	END RX	DEC	ERR	DO	DI	R
IMR1	+1	CPT IE	APT IE	DET IE	END IE	DEC IE	ERR IE	DO IE	DI IE	W
ISR2	+2	INT	SRQI	LOK	REM	CO	LOKC	REMC	ADSC	R
IMR2	+2	0	SRQI IE	DMAO	DMAI	CO IE	LOKC IE	REMC IE	ADSC IE	2
SPSR	+3	S8	PEND	S6	S5	S4	S3	S2	S1	R
VSR	+3	V3	V2	V1	V0	X	X	X	X	R
ICR2	+3	1	0	SLOW	0	0	0	0	MICR	W
SPMR	+3	S8	rsv/RQS	S6	S5	S4	S3	S2	S1	W
ADSR	+4	CIC	ATN*	SPMS	LPAS	TPAS	LA	TA	MJMN	R
ADMR	+4	ton	lon	TRM1	TRM0	0	0	ADM1	ADM0	W
CPTR	+5	CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0	R
SASR	+5	nba	AEHS	ANHS1	ANHS2	ADHS	ACRDY	SH1A	SH1B	R
AUXMR	+5	AUX7	AUX6	AUX5	AUX4	AUX3	AUX2	AUX1	AUX0	W
ADR0	+6	X	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0	R
ISR0	+6	nba	STBO	NL	EOS	IFCI	ATNI	X	SYNC	R
IMR0	+6	GLINT	STBO 1E	NLEN	BTO	IFCI IE	ATNI IE	0	SYNC IE	W
ADR	+6	ARS	DT	DL	AD5	AD4	AD3	AD2	AD1	W
ADR1	+7	EOI	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1	R
BSR	+7	ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN	R
BCR	+7	ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN	W
EOSR	+7	EOS7	EOS6	EOS5	EOS4	EOS3	EOS2	EOS1	EOS0	W

The Page-In State

At some offsets, Table 3-1 shows two readable or two writable registers. The shaded registers in Table 3-1 are accessible only when the Page-In state is true. For each shaded register, the corresponding unshaded register is accessible only when the Page-In state is false.

How to Page-In

The NAT7210 enters the Page-In state when the host interface writes the Page-In auxiliary command to the Auxiliary Mode Register (AUXMR). The NAT7210 registers appear at their Page-In state offset for the first register access after the Page-In command. The NAT7210 leaves the Page-In state at the end of the first register access after the Page-In command.

Hidden Registers

In addition to the registers shown in Table 3-1, the NAT7210 contains hidden registers. All hidden registers are write-only registers. Two or more hidden registers can appear at the same offset. When you write an 8-bit pattern to these offsets, some of the bits determine which hidden register will be written. The other bits represent the value written to the register.

Address Register Map

The NAT7210 has two address registers: ADR1 and ADR0. Table 3-1 shows the offsets for the readable portion of ADR1 and ADR0. The writable portion of ADR0 and ADR1 appears at the offset of the Address Register (ADR) shown in Table 3-1. Table 3-2 shows the bit map for the two writable address registers.

Table 3-2. Hidden Registers at Offset 6 (ADR)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADR0	0	DT0	DL0	AD5–0	AD4–0	AD3–0	AD2–0	AD1–0
ADR1	1	DT1	DL1	AD5–1	AD4–1	AD3–1	AD2–1	AD1–1

Auxiliary Mode Register Map

Several hidden registers appear at the AUXMR offset. Table 3-3 shows these hidden registers.

Table 3-3. Hidden Registers at Offset 5 (AUXMR)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PPR	0	1	1	U	S	P 3	P 2	P 1
AUXRA	1	0	0	BIN	XEOS	REOS	HLDE	HLDA
AUXRB	1	0	1	ISS	INV	TRI	SPEOI	CPT ENABLE
AUXRE	1	1	0	0	DHADT	DHADC	DHDT	DHDC
AUXRF	1	1	0	1	DHATA	DHALA	DHUNTL	DHALL
AUXRG	0	1	0	0	NTNL	RPP2	DISTCT	CHES
AUXRI	1	1	1	0	USTD	PP2	0	SISB
ICR	0	0	1	0	F 3	F 2	F 1	F 0

Register Bit Descriptions

Some 7210-mode registers and 9914-mode registers share identical names. The 9914-mode registers are described in Chapter 4, *9914-Mode Interface Registers*. If you are using the NAT7210 in 7210 mode, be sure to read the proper description for the 7210-mode registers.

All registers are listed in alphabetical order. The registers are alphabetized according to their mnemonics.

Address Mode Register (ADMR)

Attributes: Write only

7	6	5	4	3	2	1	0
ton	lon	TRM1	TRM0	0	0	ADM1	ADM0

The host interface can put the NAT7210 into one of six GPIB addressing modes by writing to the Address Mode Register (ADMR). The values of ADMR (7–6; 3–0) are undefined after a hardware reset. Before the host interface can clear pon, it must write a valid pattern to the ADMR.

Table 3-4. Valid ADMR Patterns

Hex Value of ADMR*	GPIB Addressing Mode
30	**No Addressing** The Controller cannot address the NAT7210 to become a Talker or Listener in no-addressing mode.
31	**Normal Dual Addressing** The NAT7210 can implement one or two logical devices by using normal dual addressing. See the *GPIB Addressing* section in Chapter 5, *Software Considerations*.
32	**Extended Single Addressing** Extended single addressing mode implements the Extended Listener and Extended Talker functions, as defined in the IEEE 488 standard, without intervention from the host interface. See the *GPIB Addressing* section in Chapter 5, *Software Considerations*.
33	**Extended Dual Addressing** Extended dual addressing mode implements the Extended Listener and Extended Talker functions, as defined in the IEEE 488 standard. This mode requires intervention from the host interface. See the *GPIB Addressing* section in Chapter 5, *Software Considerations*.

(continues)

ADMR (continued)

Table 3-4. Valid ADMR Patterns (Continued)

Hex Value of ADMR*	GPIB Addressing Mode
70	**Listen Only (lon)** The NAT7210 becomes a GPIB Listener and enters the Listener Active State (LACS). Do not use lon if a GPIB Controller is present in the GPIB system. The host interface should write a hex 30 (No Addressing) to the ADMR immediately after writing lon to the ADMR. To force the NAT7210 to exit LACS, issue the unlisten (lul) auxiliary command.
B0	**Talk Only (ton)** The NAT7210 becomes a GPIB Talker. Do not use ton if a GPIB Controller is present in the GPIB system. The host interface should write a hex 30 (No Addressing) to the ADMR immediately after writing ton to the ADMR. To force the NAT7210 to exit TACS, issue the local untalk (lut) auxiliary command.
* The hex values in Table 3-4 assume that TRM1 = 1 and TRM0 = 1.	

ADMR (continued)

Bit **Mnemonic** **Description**

5–4w TRM[1–0] Transmit/Receive Mode bit

TRM1 and TRM0 control the function of the NAT7210 T/R2 and T/R3 output pins in the following manner:

TRM1	TRM0	T/R2	T/R3
0	0	EOIOE	TRIG
0	1	CIC	TRIG
1	0	CIC	EOIOE
1	1	CIC	PE

Key:

EOIOE = GPIB EOI signal output enable TACS + SPAS + CIC & ~(CSBS + CSHS)
CIC = Controller-in-Charge (CIDS + CADS)
TRIG = Trigger (pulses when DTAS = 1 or a trigger auxiliary command is issued)
PE = Pull-up enable (CIC + -(PPAS))

A hardware reset clears TRM1 and TRM0.

Note: ***In many applications, the NAT7210 is interfaced to a 75160 and a 75162 GPIB transceiver. In these applications, TRM1 and TRM0 should always be set.***

Address Register (ADR)

Attributes: Write only

7	6	5	4	3	2	1	0
ARS	DT	DL	AD5	AD4	AD3	AD2	AD1

Writing to the Address Register (ADR) loads the internal registers ADR0 and ADR1. You must load both ADR0 and ADR1 for all addressing modes. See the *GPIB Addressing* section in Chapter 5, *Software Considerations*.

Bit	Mnemonic	Description
7w	ARS	Address Register Select bit If ARS = 1, writing to the ADR loads the seven low-order bits of ADR into internal register ADR1. If ARS = 0, writing to the ADR loads the seven low-order bits into ADR0.
6w	DT	Disable Talker bit DT = 1 disables recognition of the GPIB talk address formed from AD[5–1]. ADR0 and ADR1 have independent DT bits.
5w	DL	Disable Listener bit DL = 1 disables recognition of the GPIB listen address formed from AD[5–1]. ADR0 and ADR1 have independent DL bits.
4–0w	AD[5–1]	NAT7210 GPIB Address bits 5 through 1 These bits specify the GPIB address of the NAT7210. The corresponding GPIB talk address is formed by adding hex 40 to AD[5–1], while the corresponding GPIB listen address is formed by adding hex 20 to AD[5–1]. The value written to AD[5–1] should not be 11111 (binary), because the corresponding talk and listen addresses would conflict with the GPIB Untalk (UNT) and GPIB Unlisten (UNL) commands. ADR0 and ADR1 have independent AD[5–1] bits.

Address Register 0 (ADR0)

Attributes: Read only

7	6	5	4	3	2	1	0
X	DT0	DL0	AD5–0	AD4–0	AD3–0	AD2–0	AD1–0

Address Register 0 (ADR0) reflects the internal GPIB address status of the NAT7210. In extended single addressing mode, ADR0 indicates the address and enable bits for the primary GPIB address of the NAT7210. In the dual primary addressing modes, ADR0 indicates the NAT7210 major primary GPIB address. See the *GPIB Addressing* section in Chapter 5, *Software Considerations.*

Bit	Mnemonic	Description
7r	X	Reads back a 1 or 0.
6r	DT0	Disable Talker 0 bit If DT0 = 1, the primary (or major) Talker function is not enabled, and ADR0 is not compared with GPIB Talker addresses. If DT0 = 0, the NAT7210 responds to a GPIB talk address matching bits AD[5–0 through 1–0].
5r	DL0	Disable Listener 0 bit If DL0 = 1, the primary (or major) Listener function is not enabled, and ADR0 is not compared with GPIB Listener addresses. If DL0 = 0, the NAT7210 responds to a GPIB listen address matching bits AD[5–0 through 1–0].
4–0r	AD[5–0 – 1–0]	NAT7210 GPIB Address bits 5–0 through 1–0 These are the lower 5 bits of the NAT7210 GPIB primary (or major) address. The primary talk address is formed by adding hex 40 to AD[5–0 through 1–0], while the primary listen address is formed by adding hex 20.

Address Register 1 (ADR1)

Attributes: Read only

7	6	5	4	3	2	1	0
EOI	DT1	DL1	AD5–1	AD4–1	AD3–1	AD2–1	AD1–1

Address Register 1 (ADR1) indicates the status of the GPIB address and enable bits for the secondary address of the NAT7210 if extended single addressing is used. ADR1 indicates the minor primary address of the NAT7210 if dual primary addressing is used. See the *GPIB Addressing* section in Chapter 5, *Software Considerations.*

Bit	Mnemonic	Description
7r	EOI	End-or-Identify bit EOI indicates the value of the GPIB EOI line that is latched when a data byte is received by the NAT7210 GPIB Acceptor Handshake (AH) function. If EOI = 1, the EOI line was asserted with the received byte. EOI is cleared by issuing the Chip Reset auxiliary command. EOI is updated after each byte is received.
6r	DT1	Disable Talker 1 bit If DT1 = 1, the secondary (or minor) Talker function is not enabled—that is, the GPIB secondary address (or minor primary talk address) is not compared with this register.
5r	DL1	Disable Listener 1 bit If DL1 = 1, the secondary (or minor) Listener function is not enabled—that is, the GPIB secondary address (or minor primary listen address) is not compared with this register.
4–0r	AD[5–1 – 1–1]	NAT7210 GPIB Address bits 5–1 through 1–1 These bits indicate the NAT7210 secondary or minor address. Form the secondary address by adding hex 60 to bits AD[5–1 through 1–1]. Form the minor talk address by adding hex 40 to AD[5–1 through 1–1]. Form the listen address by adding a hex 20.

Address Status Register (ADSR)

Attributes: Read only

7	6	5	4	3	2	1	0
CIC	ATN*	SPMS	LPAS	TPAS	LA	TA	MJMN

The Address Status Register (ADSR) contains information that you can use to monitor the NAT7210 GPIB address status.

Bit	Mnemonic	Description
7r	CIC	Controller-In-Charge bit CIC = ~(CIDS + CADS) CIC indicates that the NAT7210 GPIB Controller function is either in an active state with ATN* asserted or a standby state with ATN* unasserted. The Controller function is in an idle state (CIDS or CADS) if CIC = 0.
6r	ATN*	Attention* bit ATN* is a status bit that indicates the current level of the GPIB ATN* signal. If ATN* = 0, the GPIB ATN* signal is asserted.
5r	SPMS	Serial Poll Mode State bit If SPMS = 1, the NAT7210 GPIB Talker (T) or Talker Extended (TE) function is enabled to participate in a serial poll. SPMS is set by SPE & ACDS SPMS is cleared by (SPD & ACDS) + pon + IFC

ADSR (continued)

Bit	Mnemonic	Description
4r	LPAS	Listener Primary Addressed State bit LPAS indicates that the NAT7210 has received its primary listen address. See the *Address Mode Register (ADMR)* section in this chapter. LPAS is cleared by (PCG & ~MLA & ACDS) + pon
3r	TPAS	Talker Primary Addressed State bit TPAS indicates that the NAT7210 has received its primary GPIB talk address. See the *Address Mode Register (ADMR)* section in this chapter. TPAS is cleared by (PCG & ~MTA & ACDS) + pon
2r	LA	Listener Active bit LA = 1 when the NAT7210 has been addressed or programmed as a GPIB Listener—that is, the NAT7210 is in the Listener Active State (LACS) or the Listener Addressed State (LADS). The NAT7210 is addressed to listen when it receives its listen address from the CIC. The NAT7210 can also be programmed to listen by using the Listen-Only (lon) bit in the ADMR. If the NAT7210 is addressed to talk, it is automatically unaddressed to listen. LA is also cleared by (UNL & ACDS) + IFC + pon + (lun & CACS) + lul
1r	TA	Talker Active bit TA = 1 when the NAT7210 has been addressed or programmed as the GPIB Talker—that is, the NAT7210 is in the Talker Active State (TACS), the Talker Addressed

ADSR (continued)

Bit	Mnemonic	Description
		State (TADS), or the Serial Poll Active State (SPAS). The NAT7210 can be addressed to talk when it receives its talk address from the CIC. It can also be programmed to talk by using the Talk-Only (ton) bit in the ADMR. If the NAT7210 is addressed to listen, it is automatically unaddressed to talk. TA is also cleared by (OTA & ACDS) + IFC + pon + lut
0r	MJMN	Major-Minor bit MJMN indicates whether the information in the other ADSR bits applies to the NAT7210 major or minor Talker and Listener functions. MJMN = 1 when the NAT7210 receives its GPIB minor talk address or minor listen address. MJMN clears when the NAT7210 receives its major talk or major listen address. The pon message also clears MJMN. **Note:** ***Only one Talker or Listener can be active at a time. The MJMN bit indicates which, if either, of the NAT7210 Talker and Listener functions is addressed or active***. MJMN is always 0 unless the normal or extended dual primary addressing mode is enabled. See the *Address Mode Register (ADMR)* section in this chapter.

Auxiliary Mode Register (AUXMR)

Attributes: Write only
Permits access to hidden registers

7	6	5	4	3	2	1	0
AUX7	AUX6	AUX5	AUX4	AUX3	AUX2	AUX1	AUX0

Use the AUXMR to issue auxiliary commands and to write the following eight hidden registers:

- Parallel Poll Register (PPR)
- Auxiliary Register A (AUXRA)
- Auxiliary Register B (AUXRB)
- Auxiliary Register E (AUXRE)
- Auxiliary Register F (AUXRF)
- Auxiliary Register G (AUXRG)
- Auxiliary Register I (AUXRI)
- Internal Counter Register (ICR)

Note: ***You should issue commands at intervals of at least 4 clock periods.***

For more information, see the *Hidden Registers* section, which is located earlier in this chapter.

AUXMR (continued)

Table 3-5 summarizes the AUXMR auxiliary commands and Table 3-6 describes the AUXMR auxiliary commands.

Table 3-5. Auxiliary Command Summary

Hex Code*	Auxiliary Command
00	Immediate Execute Power-On (pon)
01	Clear Parallel Poll Flag (~ist)
02	Chip Reset (chip_reset)
03	Finish Handshake (rhdf)
04	Trigger (trig)
05	Clear Or Pulse Return To Local (rtl)
06	Send EOI (seoi)
07	Nonvalid Secondary Command Or Address (nonvalid)
08†	Request Control Command (rqc)
09	Set Parallel Poll Flag (ist)
0A†	Release Control Command (rlc)
0B†	Untalk Command (lut)
0C†	Unlisten Command (lul)
0D	Set Return To Local
0E†	New Byte Available False (nbaf)
0F	Valid Secondary Command or Address (valid)
10	Go To Standby (gts)

(continues)

AUXMR (continued)

Table 3-5. Auxiliary Command Summary (Continued)

Hex Code*	Auxiliary Command
11 12 1A	Take Control Asynchronously (tca) Take Control Synchronously (tcs) Take Control Synchronously On End (tcse)
13 1B 1C	Listen (ltn) Listen In Continuous Mode (ltn and cont) Local Unlisten (lun)
14	Disable System Control (~rsc)
15†	Switch To 9914 Mode Command (sw9914)
16 1E	Clear IFC (~sic & rsc) Set IFC (sic & rsc)
17 1F	Clear REN (~sre & rsc) Set REN (sre & rsc)
18† 19†	Request rsv True (reqt) Request rsv False (reqf)
1D	Execute Parallel Poll (rppl)
50†	Page-In Additional Registers (page-in)
51†	Holdoff Handshake Immediately (hldi)
54†	Clear DET (ISR1[5]r) Command
55†	Clear END (ISR1[4]r) Command
56†	Clear DEC (ISR1[3]r) Command
57†	Clear ERR (ISR1[2]r) Command
58†	Clear SRQI (ISR2[6]r) Command
59†	Clear LOKC (ISR2[2]r) Command
5A†	Clear REMC (ISR2[1]r) Command

(continues)

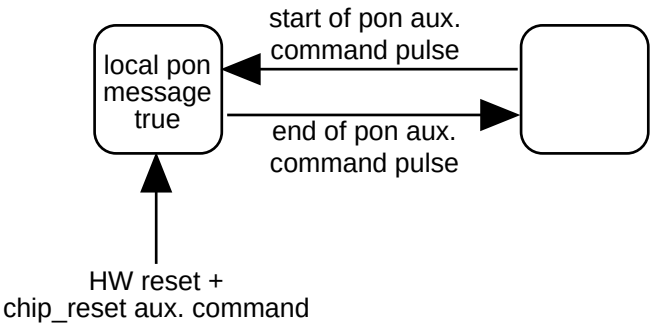
AUXMR (continued)

Table 3-5. Auxiliary Command Summary (Continued)

Hex Code*	Auxiliary Command
5B†	Clear ADSC (ISR2[0]r) Command
5C†	Clear IFCI (ISR0[3]r) Command
5D†	Clear ATNI (ISR0[2]r) Command
5E† 5F†	Clear SYNC (ISR0[0]r) Command Set SYNC (ISR0[0]r) Command
* Represents all eight bits of the AUXMR. † Denotes an auxiliary command not available in the NEC μPD7210.	

AUXMR (continued)

Table 3-6. Auxiliary Command Description

Data Pattern (Hex)	Description
00	**Immediate Execute Power-On (pon)** The Immediate Execute Power-On auxiliary command sets the local pon message true, then clears it. If the local pon message is already asserted, the pon auxiliary command simply clears the local pon message. The following figure illustrates the behavior of the local pon message:  When the local pon message is true, the NAT7210 holds all GPIB interface functions in their idle states.
01 09	**Clear Parallel Poll Flag (~ist)** **Set Parallel Poll Flag (ist)** These commands set and clear the Parallel Poll Flag. The value of the Parallel Poll Flag is used as the local message ist when AUXRB[4]w = 0. The value of SRQS is used as ist when ISS = 1. A hardware reset or the Chip Reset auxiliary command clears ist.

(continues)

AUXMR (continued)

Table 3-6. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
02	**Chip Reset** The Chip Reset auxiliary command resets the NAT7210 to the following conditions: • The local pon message is set and the interface functions are placed in their idle states. • The SPMR bits are cleared. • The TRM[1–0] bits are cleared. • The EOI bit is cleared. • The AUXRA, AUXRB, AUXRE, AUXRF, AUXRG, and AUXRI registers are cleared. • The Parallel Poll Flag is cleared. • The BCR is cleared. The interface functions remain in their idle states until they are released by an Immediate Execute pon command. While the interface functions are in their idle states, the host interface can program the NAT7210 writable bits to their desired states.
03	**Finish Handshake (rhdf)** The Finish Handshake command finishes a GPIB handshake that was stopped because of a Holdoff On RFD condition. See *The GPIB rdy Message and RFD Holdoffs* section in Chapter 5, *Software Considerations*.

(continues)

AUXMR (continued)

Table 3-6. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
04	**Trigger (trig)** The Trigger command generates a high pulse on the T/R 3 pin when TRM1 = 0. The DET bit is not set by issuing the Trigger command.
05 0D	**Clear Or Pulse Return To Local (rtl)** **Set Return To Local (rtl)** The two Return To Local commands implement the rtl message as defined by the IEEE 488 standard. If the host interface writes 05 hex, the rtl message is generated in the form of a pulse. If rtl is already set, the rtl command clears it. If the host interface writes 0D hex, the rtl command is set and remains set until either the 05 hex rtl command is issued or a Chip Reset auxiliary command is issued.
06	**Send EOI (seoi)** The seoi command causes the GPIB End-or-Identify (EOI) line to go true with the next data byte transmitted. The EOI line is cleared upon completion of the handshake for that byte. When NTNL = 0, the NAT7210 recognizes the seoi command only if TACS = 1—that is, the NAT7210 is in the Talker Active State.
07	**Nonvalid Secondary Command Or Address (nonvalid)** The nonvalid command releases a DAC (Data Accepted) holdoff. If APT = 1, the NAT7210 operates as if an Other Secondary Address (OSA) message had been received.
08*	**Request Control Command (rqc)** If the NAT7210 is in the Idle Controller State, the rqc command forces the NAT7210 to become the Active Controller when it detects that the ATN signal is unasserted.
0A*	**Release Control Command (rlc)** The rlc command forces the NAT7210 to become an Idle Controller and to unassert ATN.

(continues)

AUXMR (continued)

Table 3-6. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
0B*	**Untalk (lut)** The lut command issues the local unt message, forcing the Talker function to enter TIDS.
0C*	**Unlisten (lul)** The lul command issues the local unl message, forcing the Listener function to enter LIDS.
0E*	**New Byte Available False (nbaf)** nbaf forces the local message, nba, to become false. This action prohibits the NAT7210 from sending the last byte written to the Command/Data Our Register (CDOR). See the *Using nbaf* section in Chapter 5, *Software Considerations*.
0F	**Valid Secondary Command Or Address (valid)** The valid command releases a DAC holdoff. If APT = 1, the NAT7210 operates as if a My Secondary Address (MSA) message had been received.
10	**Go To Standby (gts)** The gts command pulses the local gts message. If the NAT7210 is the Active Controller, gts causes the NAT7210 to become the Standby Controller and to unassert the GPIB ATN signal. See the *Three Basic Controller States* section in Chapter 6, *Controller Software Considerations.*
11	**Take Control Asynchronously (tca)** The tca command pulses the local tca message. If the NAT7210 is the Standby Controller, tca causes the NAT7210 to become the Active Controller and to assert the GPIB ATN signal. See the *Standby State to Active State* section in Chapter 6, *Controller Software Considerations*.

(continues)

AUXMR (continued)

Table 3-6. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
12	**Take Control Synchronously (tcs)** The tcs command sets the local tcs message. If the NAT7210 is the Standby Controller and an Active Listener, the tcs message causes the NAT7210 to become the Active Controller when the NAT7210 performs an RFD holdoff—that is, the AH function enters ANRS. The local tcs message clears when the NAT7210 becomes the Active Controller by this method or if the NAT7210 becomes an Idle Controller. See the *Standby State to Active State* section in Chapter 6, *Controller Software Considerations*.
13	**Listen (ltn)** The ltn command pulses the local ltn message. If the NAT7210 is the Active Controller, the local ltn message causes the NAT7210 to become an Addressed Listener. The ltn command can also take the NAT7210 out of the continuous data-receiving mode (see ltn & cont command).
14	**Disable System Control (~rsc)** The ~rsc command, a hardware reset, or the Chip Reset auxiliary command clears the local rsc message. See the *System Controller Considerations* section in Chapter 6, *Controller Software Considerations*.
15*	**Switch To 9914A Mode (sw9914)** This command places the NAT7210 in 9914 compatibility mode.
16	**Clear IFC (~sic & rsc)** The ~sic & rsc command clears the local sic message and sets the local rsc messages. This action causes the NAT7210 to become the System Controller and to unassert the GPIB IFC signal. See the *System Controller Considerations* section in Chapter 6, *Controller Software Considerations*.

(continues)

AUXMR (continued)

Table 3-6. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
17	**Clear REN (~sre & rsc)** The ~sre & rsc command clears the local sre message and sets the local rsc messages. This action causes the NAT7210 to become the System Controller and to unassert the GPIB REN signal. See the *System Controller Considerations* section in Chapter 6, *Controller Software Considerations*.
18* 19*	**Request rsv True (reqt)** **Request rsv False (reqf)** The reqt and reqf commands are inputs to the IEEE 488.2 Service Request Synchronization Circuitry. These commands set and clear the local rsv message. If STBO IE = 1, the reqt and reqf commands are issued immediately. If STBO IE = 0, the reqt and reqf commands are not issued immediately: they are issued on the write of the SPMR that follows the issuing of the reqt or reqf auxiliary command.
1A	**Take Control Synchronously On END (tcse)** The tcse command causes the local tcs message to set when the NAT7210 accepts a byte satisfying the END condition (see the END RX bit ISR1[4]). If the NAT7210 is the Standby Controller and an Active Listener, the tcs message causes the NAT7210 to become the Active Controller when the NAT7210 performs an RFD holdoff—that is, when the AH function enters ANRS. The local tcs message (and the END detection circuitry) clears when the NAT7210 becomes the Active Controller by this method or if the NAT7210 becomes an Idle Controller.

(continues)

AUXMR (continued)

Table 3-6. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
1B	**Listen In Continuous Mode (ltn & cont)** The ltn & cont command pulses the local ltn message. If the NAT7210 is the Active Controller, the local ltn message causes the NAT7210 to become an Addressed Listener. The ltn & cont command also places the NAT7210 in continuous mode regardless of the settings of the AUXRA[1–0] bits (see *The GPIB rdy Message and RFD Holdoffs* section in Chapter 5). If the NAT7210 enters continuous mode because of the ltn & cont command, it remains in continuous mode until the NAT7210 becomes unaddressed to Listen—that is, the L or LE function enters LIDS—or until the control program issues the ltn command.
1C	**Local Unlisten (lun)** The lun command pulses the local lun message. If the NAT7210 is the Active Controller, the local lun message causes the NAT7210 to become an Unaddressed Listener—that is, the L or LE function enters LIDS.
1D	**Execute Parallel Poll (rpp1)** The rpp1 command sets the local rpp message. If the NAT7210 is the Active Controller, the rpp message causes the NAT7210 to send the IDY message to all GPIB devices in the system and to conduct a parallel poll. (See the *Conducting Parallel Polls* section in Chapter 6.) The rpp message clears when the NAT7210 completes a parallel poll or becomes an Idle Controller.
1E	**Set IFC (sic & rsc)** The sic & rsc command sets the local sic and rsc messages. The local message pon or the ~rsc auxiliary command also clears sic. This action causes the NAT7210 to become the System Controller and to assert the GPIB IFC signal. See the *System Controller Considerations* section in Chapter 6, *Controller Software Considerations*.

(continues)

AUXMR (continued)

Table 3-6. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
1F	**Set REN (sre & rsc)** The sre & rsc command sets the local sre and rsc messages. The local message pon or the ~rsc auxiliary command also clears sre. This action causes the NAT7210 to become the System Controller and to assert the GPIB REN signal. See the *System Controller Considerations* section in Chapter 6, *Controller Software Considerations*.
50*	**Page-In Additional Registers (page-in)** The Page-In command causes the NAT7210 to enter the Page-In state. The Page-In state makes several registers accessible. See *The Page-In State* section located at the beginning of this chapter.
51*	**Holdoff Handshake Immediately (hldi)** This command forces the Acceptor Handshake function to immediately perform an RFD holdoff. Issuing this command forces a transition into ANRS, where the handshake is held off until a finish handshake auxiliary command is issued.
54*	**Clear DET** This command clears the DET bit (ISR1[5]r). Use this command to clear the DET bit when SISB = 1.
55*	**Clear END** This command clears the END bit (ISR1[4]r). Use this command to clear the END bit when SISB = 1.
56*	**Clear DEC** This command clears the DEC bit (ISR1[3]r). Use this command to clear the DEC bit when SISB = 1.
57*	**Clear ERR** This command clears the ERR bit (ISR1[2]r). Use this command to clear the ERR bit when SISB = 1.

(continues)

AUXMR (continued)

Table 3-6. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
58*	**Clear SRQI Command** This command clears the SRQI bit if SISB = 1. See the SRQI bit description that is in the *Interrupt Status Register 2 (ISR2)* section in this chapter.
59*	**Clear LOKC** This command clears the LOKC bit (ISR2[2]r). Use this command to clear the LOKC bit when SISB = 1.
5A*	**Clear REMC** This command clears the REMC bit (ISR2[1]r). Use this command to clear the REMC bit when SISB = 1.
5B*	**Clear ADSC** This command clears the ADSC bit (ISR2[0]r). Use this command to clear the ADCS bit when SISB = 1.
5C*	**Clear IFCI** This command clears the IFCI bit (ISR0[3]r). Use this command to clear the IFCI bit when SISB = 1.
5D*	**Clear ATNI** This command clears the ATNI bit (ISR0[2]r). Use this command to clear the ATNI bit when SISB = 1.
5E* 5F*	**Clear SYNC** **Set SYNC** These commands start or reset the SYNC function.
* Denotes an auxiliary command not available in the μPD7210.	

Auxiliary Register A (AUXRA)

Attributes: Write only
Accessed at the same offset as AUXMR

7	6	5	4	3	2	1	0
1	0	0	BIN	XEOS	REOS	HLDE	HLDA

AUXRA controls the End-of-String (EOS) and END messages and specifies the RFD holdoff mode. The Chip Reset auxiliary command or a hardware reset clears AUXRA. You write to AUXRA at the same offset as the AUXMR.

Bit	Mnemonic	Description
4w	BIN	Binary bit The BIN bit selects the length of the EOS message. If BIN = 1, the End-of-String Register (EOSR) is treated as an 8-bit byte. When BIN = 0, the EOSR is treated as a 7-bit register (for ASCII characters), and only a 7-bit comparison is done with the data on the GPIB.
3w	XEOS	Transmit END With EOS bit XEOS permits or prohibits automatic transmission of the GPIB END message at the same time as the EOS message when the NAT7210 is in TACS. If XEOS = 1 and the byte in the CDOR matches the contents of the EOSR, the EOI line is sent true along with the data.
2w	REOS	END On EOS Received bit The REOS bit permits or prohibits setting the END bit (ISR1[4]r) when the NAT7210 receives the EOS message as a Listener. If REOS = 1 and the byte in the DIR matches the byte in the EOSR, the END RX bit (ISR1[4]r) is set and the acceptor function treats the EOS character just as if it were received with EOI asserted.

AUXRA (continued)

Bit	Mnemonic	Description
1w, 0w	HLDE HLDA	Holdoff On End bit Holdoff On All Data bit

HLDE and HLDA together determine the GPIB data-receiving mode.

HLDE	HLDA	Data-Receiving Mode
0	0	Normal Handshake Mode
0	1	RFD Holdoff on All Data Mode
1	0	RFD Holdoff on END Mode
1	1	Continuous Mode

For more information, see *The GPIB rdy Message and RFD Holdoffs* section in Chapter 5, *Software Considerations*.

Issuing the ltn & cont auxiliary command can also place the NAT7210 in the continuous data-receiving mode. The NAT7210 enters continuous mode regardless of the value of HLDE and HLDA. In this situation, the NAT7210 remains in continuous mode until you issue the ltn auxiliary command or the NAT7210 becomes unaddressed to listen (by entering the LIDS).

Auxiliary Register B (AUXRB)

Attributes: Write only
Accessed at the same offset as AUXMR

7	6	5	4	3	2	1	0
1	0	1	ISS	INV	TRI	SPEOI	CPT ENABLE

AUXRB affects several interface functions. The Chip Reset auxiliary command or a hardware reset clears AUXRB. You write to AUXRB at the same offset as the AUXMR.

Bit **Mnemonic** **Description**

4w ISS Individual Status Select bit

ISS determines the value of the NAT7210 ist message. When ISS = 1, ist takes on the value of the NAT7210 Service Request State (SRQS). (The NAT7210 asserts the GPIB SRQ message when it is in SRQS.) If ISS = 0, ist takes on the value of the NAT7210 Parallel Poll Flag. You set and clear the Parallel Poll Flag by using the Set Parallel Poll Flag and Clear Parallel Poll Flag auxiliary commands. See the *Parallel Polling* section in Appendix B, *Introduction to the GPIB*.

3w INV Invert bit

INV determines the polarity of the INT pin.

INV Bit	INT Pin Polarity
0	Active High
1	Active Low

2w TRI Three-State Timing bit

TRI determines the NAT7210 GPIB Source Handshake Timing (T1). Clearing TRI sets the low-speed timing (T1 ≥ 2 μs). Setting TRI enables the NAT7210 to use a shorter T1 delay. See the *T1 Delay Generation* section in Chapter 5, *Software Considerations*.

AUXRB (continued)

Bit	Mnemonic	Description
1w	SPEOI	Send Serial Poll EOI bit SPEOI determines whether the NAT7210 sends EOI when a Controller serial polls the NAT7210.

SPEOI	EOI During Serial Polls
0	Sent False
1	Sent True

0w	CPT ENABLE	Command Pass Through Enable bit The CPT ENABLE bit permits or prohibits detecting undefined GPIB commands and permits or prohibits setting the CPT bit (ISR1[7]r).

Auxiliary Register E (AUXRE)

Attributes: Write only
Accessed at the same offset as AUXMR

7	6	5	4	3	2	1	0
1	1	0	0	DHADT	DHADC	DHDT	DHDC

AUXRE determines when the NAT7210 performs a DAC holdoff. The Chip Reset auxiliary command or a hardware reset clears AUXRE.

Each bit of AUXRE enables DAC holdoffs on a GPIB command or group of commands. When a GPIB Controller sends the specified command to the NAT7210, the CPT bit sets and the NAT7210 performs a DAC holdoff. See the *DAC Holdoffs* section in Chapter 5, *Software Considerations.*

Bit	**Mnemonic**	**Description**
3w	DHADT	DAC Holdoff On GET Command bit
2w	DHADC	DAC Holdoff On DCL Or SDC Command bit
1w	DHDT	DAC Holdoff On DTAS Command bit
0w	DHDC	DAC Holdoff On DCAS Command bit

Auxiliary Register F (AUXRF)

Attributes: Write only
Accessed at the same offset as AUXMR

7	6	5	4	3	2	1	0
1	1	0	1	DHATA	DHALA	DHUNTL	DHALL

AUXRF determines how the NAT7210 uses a DAC holdoff. The Chip Reset auxiliary command or a hardware reset clears AUXRF.

Each bit of AUXRF enables DAC holdoffs on a GPIB command or group of commands. When a GPIB Controller sends the specified command to the NAT7210, the CPT bit sets and the NAT7210 performs a DAC holdoff. See the *DAC Holdoffs* section in Chapter 5, *Software Considerations.*

Bit	Mnemonic	Description
3w	DHATA	DAC Holdoff On All Talker Addresses Command bit
2w	DHALA	DAC Holdoff On All Listener Addresses Command bit
1w	DHUNTL	DAC Holdoff On The UNT Or UNL Command bit
0w	DHALL	DAC Holdoff On All UCG, ACG, And SCG Commands bit

Auxiliary Register G (AUXRG)

Attributes: Write only
Accessed at the same offset as AUXMR

7	6	5	4	3	2	1	0
0	1	0	0	NTNL	RPP2	DISTCT	CHES

Bit	Mnemonic	Description
3w	NTNL	No Talking When No Listener bit

Set NTNL to prevent the NAT7210 from sourcing data (talking) when there is no external Listener, to modify the setting of the ERR bit, to modify the way the nba local message is cleared, and to change the EOI generation function. If the NAT7210 is used in an IEEE 488.2 device, you should set NTNL.

If NTNL = 0, the following actions occur:

- The NAT7210 handshake function enters STRS after the T1 delay has elapsed and NRFD is unasserted.
- The ERR bit is set on TACS & SDYS & DAC & RFD or SIDS & (write CDOR) or the transition from SDYS to SIDS.
- The local nba message is cleared upon entering SIDS or STRS.
- The Send EOI auxiliary command is ignored or forgotten upon exiting TACS.

If NTNL = 1, the following actions occur:

- The NAT7210 handshake function does not make the transition from SDYS to STRS unless an external Listener exists—that is, a device on the GPIB is asserting NDAC.
- The ERR bit is set when the T1 delay has elapsed and TACS & SDYS & EXTDAC & RFD (where EXTDAC refers to some device on the GPIB asserting NDAC).

AUXRG (continued)

Bit	Mnemonic	Description
		• The local nba message is cleared upon entering STRS and ~SPAS. • The Send EOI auxiliary command is cleared upon entering SDYS or STRS.
2w	RPP2	Request Parallel Poll 2 bit If RPP2 = 1, the local rpp message is true. Clearing RPP2 clears rpp. If the NAT7210 is the Active Controller, setting rpp causes the NAT7210 to conduct a parallel poll. See the *Conducting Parallel Polls* section in Chapter 6, *Controller Software Considerations*.
1w	DISTCT	Disable Automatic Take Control bit If DISTCT = 1, the NAT7210 considers the GPIB TCT message undefined. If the GPIB Controller tries to pass control to the NAT7210, the NAT7210 will not become a Controller. You usually set DISTCT if the NAT7210 is a talk-only or listen-only device. If DISTCT = 0, the NAT7210 recognizes TCT. Another Controller may pass control to the NAT7210 without software intervention.
0w	CHES	Clear Holdoff On End Select bit CHES determines how long the NAT7210 remembers that it detected an END condition. If CHES = 0, the NAT7210 remembers the detection of the END condition until the host interface issues the Release Handshake Holdoff auxiliary command. If CHES = 1, the NAT7210 remembers the detection of the END condition until the Release Handshake Holdoff auxiliary command is issued or the DIR is read when the NAT7210 is in the normal handshake holdoff mode—that is, HLDE and HLDA = 0.

Auxiliary Register I (AUXRI)

Attributes: Write only
Accessed at the same offset as AUXMR

7	6	5	4	3	2	1	0
1	1	1	0	USTD	PP2	0	SISB

Bit	Mnemonic	Description
3w	USTD	Ultra Short T1 Delay bit USTD sets the value of the T1 delay (used by the Source Handshake function for data setup) to 350 ns for the second and subsequent data bytes sent after ATN unasserts. If USTD = 0, the TRI bit (AUXRB[2]w) determines the value of T1. See the *T1 Delay Generation* section in Chapter 5, *Software Considerations*.
2w	PP2	Parallel Poll bit 2 If PP2 = 0, the NAT7210 responds to parallel polls in the same manner as the μPD7210—that is, it supports Parallel Poll functions PP1 and PP2 simultaneously. However, a contradiction arises because PP1 requires the interface to be configured by remote GPIB commands, and PP2 requires the interface to be configured locally and ignore remote GPIB commands. When PP2 = 1, the chip ignores remote GPIB commands—that is, PPC and PPU are treated as undefined commands, allowing a true implementation of PP2. In addition, setting PP2 and U (PPR[4]w) lets the NAT7210 support PP0 (no Parallel Poll response). See the *Parallel Polling* section in Appendix B.
0w	SISB	Static Interrupt Status Bits bit If SISB = 0, reading ISR0, ISR1, or ISR2 clears the bits of the register that is read (ISR0, ISR1, or ISR2).

AUXRI (continued)

If SISB = 1, the bits remain set until a certain condition is met. Table 3-7 lists the condition that clears each interrupt status bit when SISB = 1.

Table 3-7. Clear Conditions for SISB Bit

Bit	Clear Condition when SISB = 1
ADSC	pon + clearADSC + ton + lon
APT	pon + valid + nonvalid
ATNI	pon + clearATNI
CO	pon + ~CACS + ~SGNS + nba
CPT	pon + read CPTR
DEC	pon + clearDEC
DET	pon + clearDET
DI	pon + (finish handshake) * (Holdoff mode) + read DIR
DO	pon + ~TACS + ~SGNS + nba
END	pon + clearEND
ERR	pon + clearERR
IFCI	pon + clearIFCI
LOKC	pon + clearLOKC
REMC	pon + clearREMC
SRQI	pon + clearSRQI

Note: ***Interrupt Status bits STBO and SYNC are not affected by the SISB bit.***

Bus Control Register (BCR)/Bus Status Register (BSR)

Attributes: Write only (BCR)
Read only (BSR)

7	6	5	4	3	2	1	0
ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN
ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN

Bit	Mnemonic	Description
7r	ATN	GPIB Attention Status bit
7w	ATN	GPIB Attention Control bit
6r	DAV	GPIB Data Valid Status bit
6w	DAV	GPIB Data Valid Control bit
5r	NDAC	GPIB Not Data Accepted Status bit
5w	NDAC	GPIB Not Data Accepted Control bit
4r	NRFD	GPIB Not Ready For Data Status bit
4w	NRFD	GPIB Not Ready For Data Control bit
3r	EOI	GPIB End-or-Identify Status bit
3w	EOI	GPIB End-or-Identify Control bit
2r	SRQ	GPIB Service Request Status bit
2w	SRQ	GPIB Service Request Control bit
1r	IFC	GPIB Interface Clear Status bit
1w	IFC	GPIB Interface Clear Control bit
0r	REN	GPIB Remote Enable Status bit
0w	REN	GPIB Remote Enable Control bit

Reads of the Bus Status Register (BSR) return the status of the GPIB control lines at the time of the read. Write ones to bits in the Bus Control Register (BCR) to assert the corresponding GPIB control lines.

BCR/BSR (continued)

Because the NAT7210 is either transmitting or receiving a GPIB control line at any particular time and is not performing both actions simultaneously, setting a bit in the BCR may not automatically assert the corresponding line on the GPIB. If the NAT7210 is transmitting a GPIB line when the corresponding bit in the BCR is set, the NAT7210 asserts the GPIB line. If the NAT7210 is receiving a GPIB line when the corresponding bit in the BCR is set, the GPIB line is not asserted. However, in both these cases, the GPIB signal internal to the NAT7210 is logically ORed with the value of the BCR bit. Figure 3-1 illustrates the GPIB input/output hardware configuration.

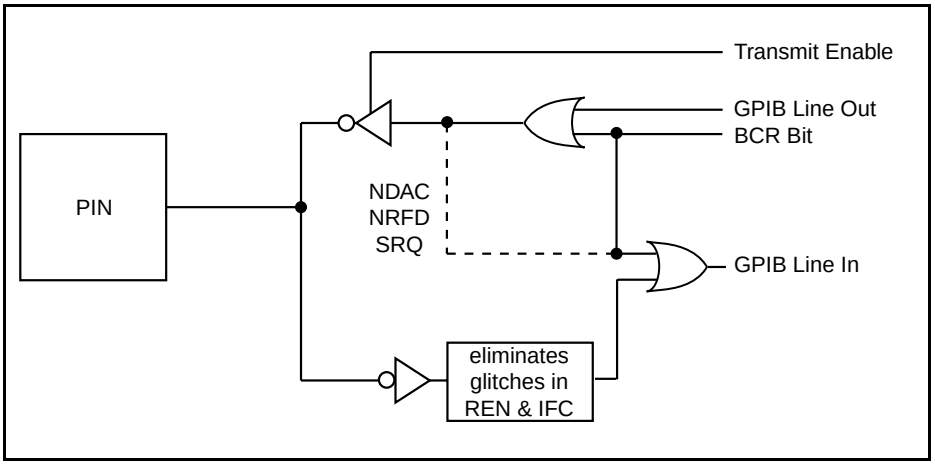


Figure 3-1. GPIB I/O Hardware Configuration

In Figure 3-1, *Transmit Enable* represents the internal signal that is true when the chip is driving a particular GPIB control line. *GPIB Line Out* represents the internal signal that is true when an interface function within the chip is attempting to assert a GPIB control signal. *BCR Bit* corresponds to the bit in the BCR. *GPIB Line In* represents the internal GPIB lines that are inputs to the GPIB interface functions and the BSR. The internal signals *SRQ*, *NDAC*, and *NRFD* are monitored by the interface functions even when they are not driven onto the pin. For this reason, the internal value of these signals is ORed with the external value.

Because the BSR samples the GPIB control lines from the GPIB transceiver—not the actual GPIB bus—the direction of each line determines the validity of each bit. Generally, when a signal is an input, the BSR reflects its true bus status, while an output signal reflects only the NAT7210 value of that particular line. Under normal GPIB operation, this restriction on the validity of the BSR should not be too limiting, because the lines that are typically monitored are valid when they are monitored. For example, the SRQ line is valid in the BSR when the NAT7210 is CIC, which is also when the SRQ line will be monitored.

Command/Data Out Register (CDOR)

Attributes: Write only

7	6	5	4	3	2	1	0
DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

Bit	Mnemonic	Description
7–0w	DIO[8–1]	GPIB data lines DIO[8–1]

The CDOR moves data from the CPU to the GPIB when the interface is the GPIB Talker or Controller. Writing to the CDOR sets the local message, nba. When nba is true, the Source Handshake (SH) function can transfer the data or command in the CDOR to other GPIB devices. Writing to the CDOR also

- Clears the Data Out (DO) bit.
- Clears the DRQ signal (unless DMAO = 0).

The host interface can write to the CDOR at offset 0 or by performing a DMA write operation.

The CDOR and the DIR use separate latches. A read of the DIR does not change data in the CDOR. The CDOR is a transparent latch; thus, the GPIB data bus (DIO(8–1)) reflects changes on the CPU data bus during write cycles to the CDOR.

Command Pass Through Register (CPTR)

Attributes: Read only

7	6	5	4	3	2	1	0
CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0

The host interface can examine the GPIB DIO lines by reading the Command Pass Through Register (CPTR). The CPTR has no storage; the host interface should read the CPTR only during a DAC holdoff. See the *DAC Holdoffs* section in Chapter 5, *Software Considerations.*

Bit	**Mnemonic**	**Description**
7–0r	CPT[7–0]	Command Pass Through bits 7 through 0

Data In Register (DIR)

Attributes: Read only

7	6	5	4	3	2	1	0
DIO8	DIO7	DIO6	DIO5	DI04	DIO3	DIO2	DIO1

Bit	Mnemonic	Description
7–0r	DIO[8–1]	GPIB data lines DIO[8–1]

The Data In Register (DIR) holds data that the NAT7210 receives when the NAT7210 is a Listener. The NAT7210 latches GPIB data into the DIR when LACS & ACDS is true.

Latching data into the DIR causes the DI bit (ISR1[0]) to set, unless the NAT7210 is in continuous mode (see AUXMR[1:0]). Usually, latching data into the DIR causes an RFD holdoff (see *The GPIB rdy Message and RFD Holdoffs* section in Chapter 5, *Software Considerations*).

The host interface can read the DIR at offset 0 or by asserting the DACK* and RD* pins. Reading the DIR also

- Clears DI (ISR1[0]).
- Clears the DRQ signal if DMAI (IMR2[4]) is set.
- Clears an RFD holdoff (depending on several other conditions).

The DIR and the CDOR use separate latches. When the host interface writes to the CDOR, data in the DIR is not changed.

End-of-String Register (EOSR)

Attributes: Write only

7	6	5	4	3	2	1	0
EOS7	EOS6	EOS5	EOS4	EOS3	EOS2	EOS1	EOS0

EOSR holds the byte that the NAT7210 uses to detect the end of a GPIB data block transfer. The NAT7210 compares data it receives to a 7- or 8-bit byte (ASCII or binary—depending on the BIN bit) in the EOSR in order to detect the end of a block of data.

If the NAT7210 is a Listener and REOS = 1, the END bit is set in ISR1 whenever the received data byte matches the EOSR. If the NAT7210 is a Talker and XEOS = 1, the END message (GPIB EOI* line asserted low) is sent along with a data byte whenever the data byte matches the EOSR.

Bit	**Mnemonic**	**Description**
7–0w	EOS[7–0]	End-of-String bits 7 through 0

Internal Count Register (ICR)

Attributes: Write only

7	6	5	4	3	2	1	0
0	0	1	0	F3	F2	F1	F0

ICR determines the internal clock frequency of the NAT7210. You write to the ICR at the same offset as the AUXMR.

Note: ***The ICR resets to 00101000 (8 MHz).***

Bit **Mnemonic** **Description**

3–0 F(3–0) Clock Frequency

These bits, in addition to MICR (ICR2[0]), determine the length of certain delays that are required by the IEEE 488 standard. You should set these bits according to the frequency of the signal driving the CLK pin. For proper operation, set F(3–0) and MICR as follows:

Clock Frequency	MICR	F(3–0)
1	0	0001
2	0	0010
3	0	0011
4	0	0100
5	0	0101
6	0	0110
7	0	0111
8	0	1000
10	1	0101
16	1	1000
20	1	1010

Internal Count Register 2 (ICR2)

Attributes: Write only

7	6	5	4	3	2	1	0
1	0	SLOW	0	0	0	0	MICR

Bit	Mnemonic	Description
5w	SLOW	Slow Handshake bit Setting this bit enables circuitry that increases the time NRFD* or NDAC* must be unasserted before the NAT7210 responds to the unassertion. If SLOW = 1, NRFD* and NDAC* must be unasserted for at least 400 ns before the NAT7210 responds to the unassertion.
0w	MICR	Modify Internal Count Register Setting this bit modifies the meaning of the bits in the Internal Count Register. For more details, see the *Internal Count Register (ICR)* section in this chapter.

Interrupt Mask Register 0 (IMR0)

Attributes: Write only

7	6	5	4	3	2	1	0
GLINT	STBO IE	NLEN	0	IFCI IE	ATNI IE	0	SYNC IE

Interrupt Status Register 0 (ISR0)

Attributes: Read only

7	6	5	4	3	2	1	0
nba	STBO	NL	EOS	IFCI	ATNI	X	SYNC

Interrupt Status Register 0 (ISR0) contains Interrupt Status bits and Internal Status bits. Interrupt Mask Register 0 (IMR0) contains Interrupt Enable bits and Internal Control bits. If an Interrupt Enable is true when the corresponding status condition or event occurs, the NAT7210 can generate a hardware interrupt request. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

Bits in ISR0 are set and cleared regardless of the status of bits in IMR0. If an interrupt condition occurs at the same time the host interface is reading ISR0, the NAT7210 does not set the corresponding Interrupt Status bit until the read is finished. A hardware reset clears all bits in IMR0 except GLINT, which is set.

Bit	Mnemonic	Description
7r	nba	New Byte Available local message bit nba reflects the status of the local New Byte Available message. nba is set on writes to the CDOR. nba is cleared by pon + nbaf + (NTNL & SIDS) + STRS
7w	GLINT	Global Interrupt Enable bit GLINT enables the NAT7210 to assert the INT pin. If GLINT = 0, the INT pin does not assert. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

IMR0/ISR0 (continued)

Bit	Mnemonic	Description
6r	STBO	Status Byte Out bit
6w	STBO IE	Status Byte Out Interrupt Enable bit
		STBO IE determines how the NAT7210 requests service and responds to serial polls. See the *Serial Polling* section in Appendix B, *Introduction to the GPIB.*
		If STBO IE = 0, the rsv bit in SPMR can be used to request service. When the GPIB Controller serial polls the NAT7210, the NAT7210 transmits the current value of SPMR.
		If STBO IE = 1, the rsv bit in the SPMR has no effect on the Service Request (SR1) function and rsv must be generated through the reqt auxiliary command. STBO sets when the GPIB Controller serial polls the NAT7210. In response to STBO, the host interface writes a byte to SPMR, then the NAT7210 transmits this byte as the Serial Poll response.
		STBO is set by STBO IE & SPAS
		STBO is cleared by pon + (write SPMR) + ~SPAS
5r	NL	New Line Receive bit
		NL is set when the NAT7210 accepts the ASCII new line character from the GPIB data bus.
		NL is set by LACS & NL & ACDS
		NL is cleared by pon + (LACS & ~NL & ACDS)

IMR0/ISR0 (continued)

Bit	Mnemonic	Description
5w	NLEN	New Line End Enable bit If NLEN = 1, the NAT7210 treats the 7-bit ASCII new line character (0A hex) as an EOS character. The Acceptor Handshake function responds to the acceptance of a new line character in the same manner as if EOI were sent.
4r	EOS	End-of-String bit The EOS bit indicates that the END bit in ISR1 was set by the acceptance of the End-of-String character. EOS is set by LACS & EOS & REOS & ACDS EOS is cleared by pon + (LACS & ~EOS & ACDS) + ~REOS
3r 3w	IFCI IFCI IE	IFC Interrupt bit IFC Interrupt Enable bit IFCI is set on the assertion of the GPIB IFC* line. IFCI is cleared by pon + (read ISR0) & ~SISB + clearIFCI
2r 2w	ATNI ATNI IE	ATN Interrupt bit ATN Interrupt Enable bit ATNI is set on the assertion of the ATN* line. ATNI is cleared by pon + (read ISR0) & ~SISB + clearATNI
1r	X	Don’t care bit

IMR0/ISR0 (continued)

Bit	Mnemonic	Description
0r	SYNC	GPIB Synchronization bit
0w	SYNC IE	GPIB Synchronization Interrupt Enable bit
		SYNC reflects the status of GPIB handshake lines after a transfer. SYNC is set at the completion of a transfer when the GPIB handshake is complete. An interrupt is generated when SYNC IE and SYNC are set.

Interrupt Mask Register 1 (IMR1)

Attributes: Write only

7	6	5	4	3	2	1	0
CPT IE	APT IE	DET IE	END IE	DEC IE	ERR IE	DO IE	DI IE

Interrupt Status Register 1 (ISR1)

Attributes: Read only
Bits are cleared when read if SISB = 0

7	6	5	4	3	2	1	0
CPT	APT	DET	END RX	DEC	ERR	DO	DI

Interrupt Status Register 1 (ISR1) contains eight Interrupt Status bits. Interrupt Mask Register 1 (IMR1) contains eight Interrupt Enable bits that directly correspond to the Interrupt Status bits in ISR1. As a result, ISR1 and IMR1 service eight possible interrupt conditions; each condition has an associated Interrupt Status bit and an Interrupt Enable bit. If an Interrupt Enable bit is true when the corresponding status condition or event occurs, the NAT7210 can generate a hardware interrupt request. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

Bits in ISR1 are set and cleared regardless of the status of the Interrupt bits in IMR1. If an interrupt condition occurs at the same time the host interface is reading ISR1, the NAT7210 does not set the corresponding Interrupt Status bit until the read is finished. A hardware reset clears all bits in IMR1.

Bit	Mnemonic	Description
7r	CPT	Command Pass Through bit
7w	CPT IE	Command Pass Through Interrupt Enable bit
		The CPT bit can flag the occurrence of two types of GPIB commands: undefined commands and user-specified commands.
		When CPT ENAB = 1, the CPT bit flags the occurrence of undefined commands and all following secondary commands. The CPT bit flags undefined Address Command Group (ACG) commands only when the NAT7210 is an addressed Talker or Listener. The host interface can read the CPTR to determine the command the NAT7210 received.

IMR1/ISR1 (continued)

Bit	Mnemonic	Description
		The CPT bit also flags the occurrence of commands that you specify when you set the AUXRE[3–0] or AUXRF[3–0] bits. When the CPT bit flags a command, the NAT7210 remains in a DAC Holdoff state until the host interface writes the valid or invalid auxiliary command to the AUXMR. CPT is set by [UCG + ACG & (TADS + LADS)] & undefined & ACDS & CPT ENABLE + UDPCF & SCG & ACDS & CPT ENABLE + DHADT & GET & ACDS + DHADC & (SDC + DCL) & ACDS + DHATA & TAG & ~UNT & ACDS + DHALA & LAG & ~UNL & ACDS + DHUNTL & (UNT + UNL) & ACDS + DHALL & ATN & ACDS CPT is cleared by pon + (read ISR1) & ~SISB + (read CPTR) & SISB UDPCF is set by [UCG + ACG & (TADS + LADS)] & undefined & ACDS & CPT ENAB UDPCF is cleared by [(UCG + ACG) & defined + TAG + LAG] & ACDS + ~(CPT ENAB) + pon
6r 6w	APT APT IE	Address Pass Through bit Address Pass Through Interrupt Enable bit APT indicates that the NAT7210 has received a secondary GPIB address. The host interface can read the secondary GPIB address in the CPTR. **Note:** ***If the application program uses extended dual addressing, it must check this bit.***

IMR1/ISR1 (continued)

Bit	Mnemonic	Description
		When APT sets, the NAT7210 enters the DAC Holdoff state. When the host interface writes the valid or invalid auxiliary command to the AUXMR, the NAT7210 exits the DAC Holdoff state. APT is set by ADM1 & ADM0 & (TPAS + LPAS) & SCG & ACDS APT is cleared by pon + (read ISR1) & ~SISB + (valid + nonvalid) & SISB
5r 5w	DET DET IE	Device Execute Trigger bit Device Execute Trigger Interrupt Enable bit DET indicates that the NAT7210 received the GPIB Group Execute Trigger (GET) command while the NAT7210 was a GPIB Listener. DET is set by DTAS = GET & LADS & ACDS DET is cleared by pon + (read ISR1) & ~SISB + clearDET
4r 4w	END RX END IE	End Received bit End Received Interrupt Enable bit END RX sets when the NAT7210, as a Listener, receives a data byte satisfying the END condition. A data byte satisfies the END condition if one of the following conditions is true: • REOS = 1 and the data byte matches the contents of the EOSR. • NLEN = 1 and the data byte matches the ASCII new line character (hex 0A). • The GPIB EOI signal is asserted when the byte is received.

IMR1/ISR1 (continued)

Bit	Mnemonic	Description
		END RX is set by (EOI + EOS & REOS + NL & NLEN) & ACDS & LACS END RX is cleared by pon + (read ISR1) & ~SISB + clearEND
3r 3w	DEC DEC IE	Device Clear bit Device Clear Interrupt Enable bit DEC indicates that either the NAT7210 received the GPIB Device Clear (DCL) command or that the NAT7210 was a GPIB Listener and received the GPIB Selected Device Clear (SDC) command. DEC is set by DCAS = (SDC & LADS + DCL) & ACDS DEC is cleared by pon + (read ISR1) & ~SISB + clearDEC
2r 2w	ERR ERR IE	Error bit Error Interrupt Enable bit The definition of ERR depends on NTNL. When NTNL = 0, ERR indicates that the contents of the CDOR have been lost. ERR sets when the NAT7210 sends data over the GPIB while no Listener exists on the GPIB. ERR also sets when a byte is written to the CDOR during SIDS or when a transition from SDYS to SIDS occurs. When NTNL = 1, ERR indicates that the source handshake has attempted to send data or commands across the bus but has found no Listeners (that is, NDAC and NRFD were unasserted). Data is not lost. The SH function does not source the data or command until a Listener appears (that is, NDAC asserts).

IMR1/ISR1 (continued)

Bit	Mnemonic	Description
		ERR is set by ~NTNL & TACS & SDYS & DAC & RFD + ~NTNL & SIDS & (write CDOR) + ~NTNL & (SDYS to SIDS) + NTNL & SDYS & EXTDAC & RFD ERR is cleared by pon + (read ISR1) & ~SISB + clearERR
1r 1w	DO DO IE	Data Out bit Data Out Interrupt Enable bit DO indicates that the NAT7210, as GPIB Talker, is ready to accept another data byte into the CDOR. This data byte will be transmitted to the GPIB. DO clears when a byte is written to the CDOR or when the NAT7210 ceases to be the Active Talker. DO is set by TACS & SGNS & ~nba DO is cleared by ~TACS + ~SGNS + nba + (read ISR1) & ~SISB
0r 0w	DI DI IE	Data In bit Data In Interrupt Enable Bit DI indicates that the NAT7210, as a GPIB Listener, has accepted a data byte from the GPIB Talker. DI is set by LACS & ACDS & ~(continuous mode) DI is cleared by pon + (read ISR1 & ~SISB) + (Finish Handshake & Holdoff mode) + (read DIR)

Interrupt Mask Register 2 (IMR2)

Attributes: Write only

7	6	5	4	3	2	1	0
0	SRQI	DMAO	DMAI	CO IE	LOKC IE	REMC IE	ADSC IE

Interrupt Status Register 2 (ISR2)

Type: 7210 mode

Attributes: Read only
Bits clear when read if SISB = 0

7	6	5	4	3	2	1	0
INT	SRQI	LOK	REM	CO	LOKC	REMC	ADSC

Interrupt Status Register 2 (ISR2) contains Interrupt Status bits and Internal Status bits. Interrupt Mask Register 2 (IMR2) contains Interrupt Enable bits and Internal Control bits. If an Interrupt Enable is true when the corresponding status condition or event occurs, the NAT7210 can generate a hardware interrupt request. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

Bits in ISR2 are set and cleared regardless of the status of the bits in IMR2. If an interrupt condition occurs at the same time the host interface is reading ISR2, the NAT7210 does not set the corresponding Interrupt Status bit until the read is finished. A hardware reset clears all bits in IMR2.

Bit	**Mnemonic**	**Description**
7r	INT	Interrupt bit This bit is the logical OR of the Enabled Interrupt Status bits in ISR0, ISR1, and ISR2.

IMR2/ISR2 (continued)

Bit	Mnemonic	Description
		INT is set by GLINT & [(CPT & CPT IE) + (APT & APT IE) + (DET & DET IE) + (ERR & ERR IE) + (END RX & END IE) + (DEC & DEC IE) + (DO & DO IE) + (DI & DI IE) + (REMC & REMC IE) + (SRQI IE & SRQI) + (LOKC & LOKC IE) + (CO IE & CO) + (ADSC & ADSC IE) + (STBO IE & STBO) + (IFCI IE & IFCI) + (ATNI IE & ATNI) + (SYNC IE & SYNC)]
6r 6w	SRQI SRQI IE	Service Request bit Service Request Interrupt Enable bit SRQI indicates that the NAT7210, as CIC, received a GPIB Service Request (SRQ) message. SRQI is normally edge sensitive; however, consider the following sequence of events: 1. SRQ asserts, which asserts SRQI. 2. The control program clears SRQI, but SRQ remains asserted. 3. The NAT7210 serial polls a device. 4. The device sends the RQS message to the NAT7210 in response to the serial poll. 5. SRQ remains asserted because another device (not the one being serial polled) is asserting SRQ. In the situation outlined above, SRQI sets at the end of the serial poll, even if SRQ never unasserts. In addition, if the control program issues the clear SRQI auxiliary command while SRQ is asserted, the NAT7210 clears SRQI for one clock pulse and then sets SRQI again. SRQI is set by (CIC & SRQ & -(RQS & DAV)) becoming true where RQS = DIO7 & ~ATN & SPMS SRQI is cleared by pon + (read ISR2) & ~SISB + clearSRQI

IMR2/ISR2 (continued)

Bit	Mnemonic	Description
5r	LOK	Lockout bit
4r	REM	Remote bit

LOK and REM indicate the status of the GPIB Remote/Local (RL1) function of the NAT7210.

LOK	REM	RL1 State
0	0	LOCS
0	1	REMS
1	0	LWLS
1	1	RWLS

See the *Remote/Local State Considerations* section in Chapter 5, *Software Considerations*.

5w	DMAO	DMA Output Enable bit

If DMAO = 1, the DRQ pin asserts when the NAT7210, as a GPIB Talker, is ready to accept another data byte into the CDOR.

4w	DMAI	DMA Input Enable bit

If DMAI = 1, the DRQ pin asserts to indicate that the NAT7210, as a GPIB Listener, has accepted a data byte from the GPIB Talker.

3r	CO	Command Out bit
3w	CO IE	Command Out Interrupt Enable bit

CO = 1 indicates that the CDOR is empty and that another command can be written to it for transmission over the GPIB without overwriting a previous command.

CO is set by

CACS & SGNS & ~nba

IMR2/ISR2 (continued)

Bit	Mnemonic	Description
		CO is cleared by (read ISR2) & ~SISB + ~CACS + ~SGNS + cdba
2r 2w	LOKC LOKC IE	Lockout Change bit Lockout Change Interrupt Enable bit
		LOKC sets when no change occurs in the LOK bit, ISR2[5]r.
		LOKC is set by any change in LOK
		LOKC is cleared by pon + (read ISR2) & ~SISB + clearLOKC
1r 1w	REMC REMC IE	Remote Change bit Remote Change Interrupt Enable bit
		REMC sets when no change occurs in the REM bit, ISR2[4]r.
		REMC is set by any change in REM
		REMC is cleared by pon + (read ISR2) & ~SISB + clearREMC
0r 0w	ADSC ADSC IE	Addressed Status Change bit Addressed Status Change Interrupt Enable bit
		ADSC sets when one of the following ADSR bits changes: TA, LA, CIC, or MJMN.
		ADSC is set by [(any change in TA) + (any change in LA) + (any change in CIC) + (any change in MJMN)] & ~(lon + ton)
		ADSC is cleared by pon + (read ISR2) & ~SISB + clearADSC + lon + ton

Parallel Poll Register (PPR)

Attributes: Write only
Accessed at the same offset as AUXMR

7	6	5	4	3	2	1	0
0	1	1	U	S	P3	P2	P1

You use the Parallel Poll Register (PPR) to locally configure the manner in which the NAT7210 responds to a parallel poll. You write to the PPR at the same offset as the AUXMR. See the *Parallel Polling* section in Appendix B, *Introduction to the GPIB*.

When you use remote Parallel Poll Configuration (IEEE 488 capability code PP1), do not write to the PPR: writing to the PPR after it is remotely configured corrupts the configuration. The NAT7210 implements remote configuration fully and automatically without software assistance. However, you must still set or clear the individual status (ist) message (by using Set/Clear Parallel Poll Flag auxiliary commands) according to pre-established system protocol convention.

When you use the local Parallel Poll Configuration (capability code PP2), write to the PPR in advance of a poll. If PP2 (AUXRI[2]w) = 0, the contents written to the PPR are overwritten if the Controller sends a Parallel Poll command (such as PPE or PPD while in PACS or PPU) that causes the remote configuration to override the local configuration. If PP2 = 1, the reception of parallel poll commands does not affect the contents of the PPR and the local configuration determines the response during parallel polls.

Bit	Mnemonic	Description
4w	U	Unconfigure bit The U bit determines whether the NAT7210 is locally configured to participate in parallel polls. If U = 1, the NAT7210 does not participate in parallel polls unless it is remotely configured to do so. If the host interface sets U, it should clear S and P[3–1] simultaneously. If U = 0, the NAT7210 participates in parallel polls and responds in the manner defined by PPR[3] through PPR[0] and by ist. S and P[3–1] are the same as the bits of the same name in the PPE message, and the I/O write operation to the PPR is the same as the receipt of the PPE message from the GPIB Controller.

PPR (continued)

Bit **Mnemonic** **Description**

3w S Status Bit Polarity (Sense) bit

S indicates the polarity, or sense, of the NAT7210 local ist message. The following table describes the function of S.

S	ist	State of DIO Line (Selected by P[3–1] During a Parallel Poll)
0	0	Low Voltage—Logic 1
0	1	Unasserted—Logic 0
1	0	Unasserted—Logic 0
1	1	Low Voltage—Logic 1

Note: ***The DIO lines must be driven with open-collector drivers during parallel polls.***

2–0w P[3–1] Parallel Poll Response bits 3 through 1

P[3–1] indicate which of the eight DIO lines is asserted during a parallel poll. The following table shows the signal on which the NAT7210 responds to parallel polls.

P[3–1]	Signals on which NAT7210 Responds to Parallel Polls
000	DIO1
001	DIO2
010	DIO3
011	DIO4
100	DIO5
101	DIO6
110	DIO7
111	DIO8

For examples of PPR configuration, see Chapter 6, *Controller Software Considerations*.

Source/Acceptor Status Register (SASR)

Attributes: Read only

7	6	5	4	3	2	1	0
nba	AEHS	ANHS1	ANHS2	ADHS	ACRDY	SH1A	SH1B

The Source/Acceptor Status Register (SASR) contains status bits that you can use to determine the state of the Source and Acceptor functions.

Bit	Mnemonic	Description
7r	nba	New Byte Available local message
6r	AEHS	Acceptor End Holdoff State bit
5r	ANHS1	Acceptor Not Ready Holdoff bit
4r	ANHS2	Acceptor Not Ready Holdoff Immediately bit
3r	ADHS	Acceptor Data Holdoff State bit
2r	ACRDY	Acceptor Ready State bit Use this bit to determine the state of the Acceptor Handshake. By monitoring the LA and ATN bits in the ADSR, the DAV bit in the BSR, and the ADHS and ACRDY bits, you can determine the state of the Acceptor Handshake function as described below: AIDS = ~ATN & ~LA ANRS = ~AIDS & ~ACRDY & ~DAV ACRS = ~AIDS & ACRDY & ~DAV ACDS = ~AIDS & ACRDY & DAV + ~AIDS & ~ACRDY & DAV & ATN & ADHS AWNS = ~AIDS & ~ACRDY & DAV & ~(ATN & ADHS)

SASR (continued)

Bit	Mnemonic	Description
1–0r	SH1A, SH1B	Source Handshake State bits

Use these bits to determine the state of the Source Handshake interface function. By monitoring the TA, Serial Poll Mode State (SPMS), ATN bits in the ADSR, and the SH1A and SH1B bits, you can determine the state of the Source Handshake function as described below:

SIDS = ~(TACS & ~ATN + CIC & ATN)
SGNS = ~SIDS & ~SH1A & ~SH1B
SDYS = ~SIDS & SH1A
STRS = ~SIDS & ~SH1A & SH1B

Serial Poll Mode Register (SPMR)

Attributes: Write only

7	6	5	4	3	2	1	0
S8	rsv/RQS	S6	S5	S4	S3	S2	S1

Serial Poll Status Register (SPSR)

Attributes: Read only

7	6	5	4	3	2	1	0
S8	PEND	S6	S5	S4	S3	S2	S1

Bit	Mnemonic	Description
7r, 7w	S8	Serial Poll Status bit 8
5–0r, 5–0w	S[6–1]	Serial Poll Status bits 6 through 1 These bits send device- or system-dependent status information over the GPIB when the Controller serial polls the NAT7210. When STBO IE = 0, the NAT7210 transmits a byte of status information, SPMR[7–0], to the CIC if the CIC serial polls the NAT7210. The SPMR bits S[8, 6–1] are double buffered. If the host interface writes to the SPMR during a serial poll when SPAS is active, the NAT7210 saves the value. The NAT7210 updates the SPMR when the NAT7210 exits SPAS. When STBO IE = 1 and the Controller serial polls the NAT7210, the STBO interrupt condition sets. The host interface should write the STB and the RQS bit to the SPMR in response to an STBO interrupt. Issuing the Chip Reset auxiliary command clears these bits.

SPMR/SPSR (continued)

Bit	Mnemonic	Description
6r	PEND	Pending bit PEND sets when rsv = 1. PEND clears when the NAT7210 is in the Negative Poll Response State (NPRS) and the local Request Service (rsv) message is false. By reading the PEND status bit, you can confirm that a request was accepted and that the Status Byte (STB) was transmitted (PEND = 0).
6w	rsv/RQS	Request Service/ RQS bit When STBO IE = 0, bit 6 is the rsv bit. The rsv bit generates the GPIB local rsv message. When rsv = 1 and the GPIB Controller is not serial polling the NAT7210, the NAT7210 enters the Service Request State (SRQS) and asserts the GPIB SRQ signal. When the Controller reads the STB during the poll, the NAT7210 clears rsv. The rsv bit is also cleared by a hardware reset or by writing 0 to it. Issuing the Chip Reset auxiliary command also clears rsv. When STBO IE = 1, bit 6 is the RQS bit. When the Controller serial polls the NAT7210, the STBO interrupt condition sets. The host interface should write the STB and the RQS bit to the SPMR in response to an STBO interrupt. The NAT7210 transfers the STB and RQS to the Controller during that particular serial poll. A hardware reset clears RQS. Issuing the Chip Reset auxiliary command also clears RQS.

Version Status Register (VSR)

Attributes: Read only

7	6	5	4	3	2	1	0
V3	V2	V1	V0	X	X	X	X

The Version Status Register (VSR) contains a value that is unique to each NAT7210 revision. You can use the VSR to distinguish a NAT7210 from a µPD7210. Future versions of the NAT7210 may read 10XX.

Bit	Mnemonic	Description
7–4r	V[3–0]	The version number of the NAT7210APD is 1000.
3–0r	X	Don't care bits

Chapter 4
9914-Mode Interface Registers

This chapter contains NAT7210 address maps and detailed descriptions of the NAT7210 interface registers in 9914 mode. For 7210-mode register descriptions, see Chapter 3, *7210-Mode Interface Registers*.

9914 Register Map

Table 4-1 is the register bit map for the NAT7210 in 9914 mode.

Notice that bold-ruled cells distinguish six registers that are accessible only when the Page-In state is true. Refer to *The Page-In Condition* section that immediately follows the register map for more information.

9914 Mode Only

Table 4-1. 9914-Mode Interface Registers

Key

= 9914-Mode Paged Registers
R = Read Register
W = Write Register

		7	6	5	4	3	2	1	0	
ISR0	+0	INT0	INT1	BI	BO	END	SPAS	RLC	MAC	R
IMR0	+0	DMAO	DMAI	BI IE	BO IE	END IE	SPAS IE	RLC IE	MAC IE	W
ISR1	+1	GET	ERR	UNC	APT	DCAS	MA	SRQ	IFC	R
IMR1	+1	GET IE	ERR IE	UNC IE	APT IE	DCAS IE	MA IE	SRQ IE	IFC IE	W
ADSR	+2	REM	LLO	ATN	LPAS	TPAS	LA	TA	ulpa	R
IMR2	+2	GLINT	STBO IE	NLEN	0	LLOC IE	ATNI IE	0	CIC IE	W
EOSR	+2	EOS7	EOS6	EOS5	EOS4	EOS3	EOS2	EOS1	EOS0	W
BCR	+2	ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN	W
ACCR	+2	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0	W
BSR	+3	ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN	R
AUXCR	+3	C/S	0	0	F4	F3	F2	F1	F0	W
ISR2	+4	nba	STBO	NL	EOS	LLOC	ATNI	X	CIC	R
ADR	+4	edpa	dal	dat	A5	A4	A3	A2	A1	W
SPSR	+5	S8	PEND	S6	S5	S4	S3	S2	S1	R
SPMR	+5	S8	rsv/RQS	S6	S5	S4	S3	S2	S1	W
CPTR	+6	CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0	R
PPR	+6	PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1	W
DIR	+7	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	R
CDOR	+7	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	W

9914 Mode Only

The Page-In Condition

Four writable registers can appear at the same offset as the Address Status Register (ADSR) (offset 4). After a hardware or software reset, no writable register appears at the ADSR offset: the NAT7210 ignores writes to that offset.

One Page-In auxiliary command exists for each of the four registers. The host interface can make one of the four registers accessible by issuing the appropriate Page-In command to the Auxiliary Command Register (AUXCR). The paged-in register remains accessible at the ADSR offset until the host interface pages-in another register or issues the Clear Page-In Register auxiliary command.

When any one of the four writable registers is accessible at the ADSR offset, Interrupt Status Register 2 (ISR2) is accessible at the same offset as ADR, and the Serial Poll Status Register (SPSR) is accessible at the same offset as the Serial Poll Mode Register (SPMR).

Hidden Registers

In addition to the registers shown in Table 4-1, the NAT7210 contains hidden registers. All hidden registers are write-only registers. Two or more hidden registers can appear at the same offset. When you write an 8-bit pattern to these offsets, some of the bits determine which hidden register will be written. The other bits represent the value written to the register.

Accessory Read Register Map

Several hidden registers appear at the ACCR offset. Table 4-2 shows these hidden registers.

Table 4-2. Hidden Registers at the ACCR Offset

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ICR	0	0	1	0	F3	F2	F1	F0
ACCA	1	0	0	BIN	XEOS	REOS	0	0
ACCB	1	0	1	ISS	INV	LWC	SPEOI	ATCT
ACCE	1	1	0	0	DHADT	DHADC	0	0
ACCF	1	1	0	1	DHATA	DHALA	DHUNTL	DHALL
ACCI	1	1	1	0	USTD	PP1	0	DMAE

9914 Mode Only

Register Bit Descriptions

Some 7210-mode registers and 9914-mode registers share identical names. The 7210-mode registers are described in Chapter 3, *7210-Mode Interface Registers*. If you are using the NAT7210 in 9914 mode, be sure to read the proper description for the 9914-mode registers.

All registers are listed in alphabetical order. The registers are alphabetized according to their mnemonics.

9914 Mode Only

Accessory Register A (ACCRA)

Attributes: Write only
Accessed at the same offset as ACCR

7	6	5	4	3	2	1	0
1	0	0	BIN	XEOS	REOS	0	0

Accessory Register A (ACCRA) controls the EOS and END messages. A hardware reset or the ch_rst auxiliary command clears ACCRA.

Bit	Mnemonic	Description
4w	BIN	Binary bit The BIN bit selects the length of the EOS message. If BIN = 1, the EOSR is treated as an 8-bit byte. When BIN = 0, the EOSR is treated as a 7-bit register (for ASCII characters), and only a 7-bit comparison is done with the data on the GPIB.
3w	XEOS	Transmit END With EOS bit The XEOS bit permits or prohibits automatic transmission of the GPIB END message at the same time as the EOS message when the NAT7210 is in Talker Active State (TACS). If XEOS = 1 and the byte in the CDOR matches the contents of the EOSR, the EOI line is sent true along with the data.
2w	REOS	END On EOS Received bit The REOS bit permits or prohibits setting the END bit (ISR0[3]r) when the NAT7210 receives the EOS message as a Listener. If REOS = 1 and the byte in the DIR matches the byte in the EOSR, the END bit (ISR1[4]r) is set and the acceptor function treats the EOS character just as if it were received with EOI asserted.

9914 Mode Only

Accessory Register B (ACCRB)

Attributes: Write only
Accessed at the same offset as ACCR

7	6	5	4	3	2	1	0
1	0	1	ISS	INV	LWC	SPEOI	ATCT

Bit **Mnemonic** **Description**

4w ISS Individual Status Select bit

ISS determines the value of the NAT7210 ist message. When ISS = 1, ist takes on the value of the NAT7210 SRQS. (The NAT7210 is asserting the GPIB SRQ message when it is in SRQS.) If ISS = 0, ist takes on the value of the NAT7210 Parallel Poll Flag. You set and clear the Parallel Poll Flag by using the Set Parallel Poll Flag and Clear Parallel Poll Flag auxiliary commands.

3w INV Invert bit

INV determines the polarity of the INT pin.

INV Bit	INT Pin Polarity
0	Active High
1	Active Low

See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations.*

2w LWC Listen When Controller bit

LWC enables the NAT7210 to accept command bytes that the NAT7210 sources when it is CIC. If LWC = 0, the NAT7210 does not accept command bytes sent by the NAT7210.

9914 Mode Only

ACCRB (continued)

Bit	Mnemonic	Description
1w	SPEOI	Send Serial Poll EOI bit SPEOI determines whether the NAT7210 sends EOI when a Controller serial polls the NAT7210.

SPEOI	EOI During Serial Polls
0	Sent False
1	Sent True

0w	ATCT	Automatic Take Control bit If ATCT = 1, the NAT7210 can—without software intervention—take control when another CIC passes control to the NAT7210. Use the CIC bit (ISR2[0] to determine when the NAT7210 receives control. See the *GPIB Controller Considerations* section in Chapter 6, *Controller Software Considerations*.

9914 Mode Only

Accessory Register E (ACCRE)

Attributes: Write only
Accessed at the same offset as ACCR

7	6	5	4	3	2	1	0
1	1	0	0	DHADT	DHADC	0	0

Accessory Register E (ACCRE) determines how the NAT7210 uses a Data Accepted (DAC) holdoff. A hardware reset or the ch_rst auxiliary command clears ACCRE.

Each bit of ACCRE enables DAC holdoffs on a GPIB command or group of commands. When a GPIB Controller sends the specified command to the NAT7210, the Unrecognized Command (UNC) bit sets and the NAT7210 performs a DAC holdoff. See the *DAC Holdoffs* section in Chapter 5, *Software Considerations.*

Bit	**Mnemonic**	**Description**
3w	DHADT	DAC Holdoff On GET bit
2w	DHADC	DAC Holdoff On DCL Or SDC bit

9914 Mode Only

Accessory Register F (ACCRF)

Attributes: Write only
Accessed at the same offset as ACCR

7	6	5	4	3	2	1	0
1	1	0	1	DHATA	DHALA	DHUNTL	DHALL

Accessory Register F (ACCRF) determines how the NAT7210 uses a DAC holdoff. A hardware reset or the ch_rst auxiliary command clears ACCRF.

Each bit of ACCRF enables DAC holdoffs on a GPIB command or group of commands. When a GPIB Controller sends the specified command to the NAT7210, the UNC bit sets and the NAT7210 performs a DAC holdoff. See the *DAC Holdoffs* section in Chapter 5, *Software Considerations.*

Bit	Mnemonic	Description
3w	DHATA	DAC Holdoff On All Talker Addresses bit
2w	DHALA	DAC Holdoff On All Listener Addresses bit
1w	DHUNTL	DAC Holdoff On The UNT Or UNL Command bit
0w	DHALL	DAC Holdoff On All UCG, ACG, And SCG Commands bit

9914 Mode Only

Accessory Register I (ACCRI)

Attributes: Write only
Accessed at the same offset as ACCR

7	6	5	4	3	2	1	0
1	1	1	0	USTD	PP1	0	DMAE

Bit	Mnemonic	Description
3w	USTD	Ultra Short T1 Delay bit If USTD = 1, the T1 delay can be as short as 350 ns. See the *T1 Delay Generation* section in Chapter 5, *Software Considerations*.
2w	PP1	Parallel Poll bit 1 The PP1 bit permits or prohibits the NAT7210's ability to automatically respond to remote parallel poll configuration. If PP1 = 1, the NAT7210 can be configured remotely for parallel polls without software intervention. The Acceptor Handshake does not perform a DAC holdoff or set the UNC bit when it receives a Parallel Poll Command (PPC or PPU). If PP1 = 0, parallel polls must be configured through the PPR, and Parallel Poll commands must be monitored by UNC.
0w	DMAE	DMA Enable bit DMAE lets you use DMAO (IMR0[7]) and DMAI (IMR0[6]) to enable the DRQ signal. If DMAE = 0, DRQ always asserts when the NAT7210 receives a data byte as a Listener or when the NAT7210 is a Talker and the CDOR is empty. DRQ is cleared by pon + (read DIR) + (write CDOR)

9914 Mode Only

Address Register (ADR)

Attributes: Write only

7	6	5	4	3	2	1	0
edpa	dal	dat	A5	A4	A3	A2	A1

ADR is used to load the primary GPIB address of the interface.

Bit	Mnemonic	Description
7w	edpa	Enable Dual Primary Addressing Mode bit Setting edpa enables the dual primary addressing mode of the NAT7210. If edpa = 1, the NAT7210 ignores the least significant bit (A1) of its GPIB address. The NAT7210 then has two consecutive primary addresses. The ulpa bit in the Address Status Register indicates which address is active.
6w	dal	Disable Listener bit Setting dal returns the NAT7210 Listener function to LIDS and forces the NAT7210 Listener function to remain in LIDS even if the chip receives its GPIB listen address or a lon auxiliary command.
5w	dat	Disable Talker bit Setting dat returns the NAT7210 Talker function to TIDS and forces the Talker function to remain in TIDS even if the chip receives its GPIB talk address or a ton auxiliary command.
4–0w	A[5–1]	NAT7210 GPIB Address bits 5 through 1 A[5–1] specify the primary GPIB address of the NAT7210. The corresponding GPIB talk address is formed by adding hex 40 to A[5–1], while the corresponding GPIB listen address is formed by adding hex 20. A[5–1] should not be 11111 (binary) to prevent the corresponding talk and listen addresses from conflicting with the GPIB UNT and GPIB UNL commands.

9914 Mode Only

Address Status Register (ADSR)

Attributes: Read only

7	6	5	4	3	2	1	0
REM	LLO	ATN	LPAS	TPAS	LA	TA	ulpa

The Address Status Register (ADSR) contains information that you can use to monitor the NAT7210 GPIB address status.

Bit **Mnemonic** **Description**

7r REM Remote bit
6r LLO Local Lockout bit

REM and LLO indicate the status of the GPIB Remote/Local (RL1) function of the NAT7210.

LLO	REM	RL1 State
0	0	LOCS
0	1	REMS
1	0	LWLS
1	1	RWLS

5r ATN Attention bit

ATN indicates the current level of the NAT7210 ATN pin. If ATN = 1, the ATN pin is asserted (active low).

4r LPAS Listener Primary Addressed State bit

LPAS indicates that the NAT7210 has accepted its primary listen address.

LPAS is cleared by
(PCG & ~MLA & ACDS) + pon

9914 Mode Only

ADSR (continued)

Bit	Mnemonic	Description
3r	TPAS	Talker Primary Addressed State bit TPAS indicates that the NAT7210 has accepted its primary talk address. TPAS is cleared by (PCG & ~MTA & ACDS) + pon
2r	LA	Listener Active bit LA = 1 when the NAT7210 has been addressed or programmed as a GPIB Listener—that is, the NAT7210 is in LACS or LADS. The NAT7210 is addressed to listen when it receives its listen address from the CIC. You can also program the NAT7210 to listen by using the Listen-Only auxiliary command. If the NAT7210 is addressed to listen, it is automatically unaddressed to talk. LA is cleared by pon + IFC + (UNL & ACDS)
1r	TA	Talker Active bit TA = 1 when the NAT7210 has been addressed or programmed as the GPIB Talker—that is, the NAT7210 is in TACS, TADS, or SPAS. The NAT7210 can be addressed to talk when it receives its talk address from the CIC. You can also program the NAT7210 to talk by using the Talk-Only auxiliary command. If the NAT7210 is addressed to talk, it is automatically unaddressed to listen. TA is cleared by pon + IFC + (OTA & ACDS)

9914 Mode Only

ADSR (continued)

Bit	Mnemonic	Description
0r	ulpa	Upper/Lower Primary Address bit ulpa indicates the least significant bit of the last primary address that the NAT7210 received. **Note:** ***Only one Talker or Listener is active at a time. ulpa indicates which, if either, NAT7210 Talker or Listener function is addressed or active.*** The ch_rst auxiliary command clears the ulpa bit in the ADSR.

9914 Mode Only

Auxiliary Command Register (AUXCR)

Attributes: Write only

7	6	5	4	3	2	1	0
C/S	0	0	F4	F3	F2	F1	F0

Use the AUXCR to issue auxiliary commands. Two basic types of commands are implemented in the AUXCR: pulsed and static. Use static commands to enable (set) or disable (clear) various features of the NAT7210. The pulsed commands stay active for one clock pulse after the AUXCR has been written.

Note: ***Writes to the AUXCR should be separated by at least 4 clock cycles.***

Table 4-3 summarizes the AUXCR auxiliary commands and Table 4-4 describes the AUXCR auxiliary commands.

Table 4-3. Auxiliary Command Summary

Hex Code	Type	Mnemonic	Auxiliary Command
00 80	static static	~swrst swrst	Clear Software Reset Set Software Reset
01 81	pulsed pulsed	nonvalid valid	Nonvalid Release DAC Holdoff Valid Release DAC Holdoff
02	pulsed	rhdf	Release RFD Holdoff
03 83	static static	~hdfa hdfa	Clear Holdoff On All Data Set Holdoff On All Data
04 84	static static	~hdfe hdfe	Clear Holdoff On END Only Set Holdoff On END Only
05	pulsed	nbaf	New Byte Available False
06 86	static static	~fget fget	Clear Force Group Execute Trigger Set Force Group Execute Trigger
07 87	static static	~rtl rtl	Clear Return To Local Set Return To Local

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-3. Auxiliary Command Summary (Continued)

Hex Code	Type	Mnemonic	Auxiliary Command
08	pulsed	feoi	Send EOI With The Next Byte
09 89	static static	~lon lon	Clear Listen Only Set Listen Only
0A 8A	static static	~ton ton	Clear Talk Only Set Talk Only
0B	pulsed	gts	Go To Standby
0C	pulsed	tca	Take Control Asynchronously
0D	pulsed	tcs	Take Control Synchronously
0E 8E	static static	~rpp rpp	Clear Request Parallel Poll Set Request Parallel Poll
0F 8F	static static	~sic sic	Clear Send Interface Clear Set Send Interface Clear
10 90	static static	~sre sre	Clear Send Remote Enable Set Send Remote Enable
11	pulsed	rqc	Request Control
12	pulsed	rlc	Release Control
13 93	static static	~dai dai	Clear Disable IMR2, IMR1, And IMR0 Interrupts Set Disable IMR2, IMR1, And IMR0 Interrupts
14	pulsed	pts	Pass Through Next Secondary
15 95	static static	~stdl stdl	Clear Short T1 Delay Set Short T1 Delay
16 96	static static	~shdw shdw	Clear Shadow Handshaking Set Shadow Handshaking

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-3. Auxiliary Command Summary (Continued)

Hex Code	Type	Mnemonic	Auxiliary Command
17 97	static static	~vstdl vstdl	Clear Very Short T1 Delay Set Very Short T1 Delay
18 98	static static	~rsv2 rsv2	Clear Request Service bit 2 Set Request Service bit 2
99	pulsed	sw7210	Switch To 7210 Mode
1A 9A	pulsed pulsed	reqf reqt	Request rsv False (reqf) Request rsv True (reqt)
1C	pulsed	ch_rst	Chip Reset
1D 9D	static static	~ist ist	Clear Parallel Poll Flag Set Parallel Poll Flag
1E	pulsed	piimr2	Page-In Interrupt Mask Register 2
1F	pulsed	pibcr	Page-In Bus Control Register
9C	pulsed	clrpi	Clear Page-In Registers
9E	pulsed	pieosr	Page-In End-of-String Register
9F	pulsed	piaccr	Page-In Accessory Register

Values not specified are reserved.

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description

Data Pattern (Hex)	Description
00 80	**Clear Software Reset (~swrst)** **Set Software Reset (swrst)** The local swrst message places all GPIB interface functions into their idle states. swrst is equivalent to the GPIB local Power On (pon) message. swrst is set by a hardware reset, the ch_rst auxiliary command, or the swrst auxiliary command. You should configure the NAT7210 while swrst is set. Configuration includes writing the address of the device into the Address Register, writing mask values into the Interrupt Mask Registers, and selecting the desired features in the Auxiliary Command, Accessory, and Address Registers. When swrst is cleared, the device becomes logically existent on the GPIB.
01 81	**Release DAC Holdoff (nonvalid)** **Release DAC Holdoff (valid)** These commands clear a DAC holdoff condition. When APT = 1, *nonvalid* indicates that the last GPIB command byte received from the Controller was an invalid secondary address. *Valid* indicates a valid secondary address. A DAC holdoff caused by any other GPIB command byte should be released with the nonvalid command. See the *DAC Holdoffs* section in Chapter 5, *Software Considerations.*
02	**Release RFD Holdoff (rhdf)** This command releases any RFD holdoffs that hdfa or hlde have caused.
03 83	**Clear Holdoff On All Data (~hdfa)** **Set Holdoff On All Data (hdfa)** If hdfa is true, the NAT7210 performs an RFD holdoff after it receives a data byte. To complete the handshake, you must issue the rhdf command after the NAT7210 receives each byte. A hardware reset or the ch_rst auxiliary command clears hdfa. See *The GPIB rdy Message and RFD Holdoffs* section in Chapter 5, *Software Considerations*.

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
04 84	**Clear Holdoff On END Only (~hdfe)** **Set Holdoff On END Only (hdfe)** If hdfe is true, the NAT7210 performs an RFD holdoff after it receives a data byte that satisfies the END condition. A hardware reset or the ch_rst auxiliary command clears hdfe. *The GPIB rdy Message and RFD Holdoffs* section in Chapter 5, *Software Considerations*.
05	**New Byte Available False (nbaf)** nbaf forces the local message, nba, to become false. This action prohibits the NAT7210 from sending the last byte written to the CDOR. See the *Using nbaf* section in Chapter 5, *Software Considerations*.
06 86	**Clear Force Group Execute Trigger (~fget)** **Set Force Group Execute Trigger (fget)** These commands generate a trigger condition. If the host interface issues ~fget, the TRIG pin pulses asserted for one clock cycle. If the host interface issues fget, the TRIG pin asserts and remains asserted until the host interface issues ~fget. These commands do not set or clear the GET bit. **Note:** ***TRIG pin refers to the T/R3 pin when TRM1 = 0. If TRM1 does not equal 1, the TRIG signal is not visible at a pin. See the*** *Address Mode Register (ADMR)* ***section in Chapter 3,*** *7210-Mode Interface Registers.*

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
07 87	**Clear Return To Local (~rtl)** **Set Return To Local (rtl)** These commands set and clear the IEEE 488 rtl local message. A hardware reset or the ch_rst auxiliary command also clears rtl. If the host interface issues the ~rtl command, the IEEE 488 rtl message pulses true. If the host interface issues the rtl command, the IEEE 488 rtl message becomes true and remains true until the host interface issues ~rtl.
08	**Send EOI With The Next Byte (feoi)** The Send EOI command causes the GPIB EOI line to go true with the next data byte transmitted.
09 89	**Clear Listen Only (~lon)** **Set Listen Only (lon)** lon forces the Listener function into the Listener Active state. ~lon forces the Listener function to leave the Listener Active state. The local message pon clears lon.
0A 8A	**Clear Talk Only (~ton)** **Set Talk Only (ton)** ton forces the Talker function into the Talker Active state. ~ton forces the Talker function to leave the Talker Active state. The local message pon clears ton.
0B	**Go To Standby (gts)** The gts command pulses the local gts message. If the NAT7210 is the Active Controller, gts causes the NAT7210 to become the Standby Controller and to unassert the GPIB ATN signal. See the *Three Basic Controller States* section in Chapter 6, *Controller Software Considerations*.

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
0C	**Take Control Asynchronously (tca)** The tca command pulses the local tca message. If the NAT7210 is the Standby Controller, tca causes the NAT7210 to become the Active Controller and to assert the GPIB ATN signal.
0D	**Take Control Synchronously (tcs)** The tcs command pulses the local tcs message. If the NAT7210 is the Standby Controller and an Active Listener, the tcs message causes the NAT7210 to become the Active Controller when the NAT7210 performs an RFD holdoff (that is, the AH function enters ANRS).
0E 8E	**Clear Request Parallel Poll (~rpp)** **Set Request Parallel Poll (rpp)** The ~rpp and rpp commands set and clear the local rpp message. A hardware reset or the ch_rst auxiliary command also clears rpp. If the NAT7210 is the Active Controller, the rpp message causes the NAT7210 to send the IDY message to all GPIB devices in the system and to conduct a parallel poll. After the NAT9914 has been conducting a parallel poll for at least 2 µs, the control program can read the Command Pass Through Register (CPTR) to obtain the parallel poll result, then the control program can end the parallel poll by issuing the ~rpp command.
0F 8F	**Clear Send Interface Clear (~sic)** **Set Send Interface Clear (sic)** The ~sic and sic commands clear and set the sic and rsc local messages. A hardware reset or the ch_rst auxiliary command also clears sic. Setting sic and rsc causes the NAT7210 to become the System Controller and to assert the GPIB Interface Clear (IFC) signal. The control program must not issue the ~sic command until after IFC has been asserted at least 100 µs. See the *System Controller Considerations* section in Chapter 6, *Controller Software Considerations*. **Note:** ***Before it issues the sic command, the control program must ensure—by some means external to the NAT7210—that the GPIB transceivers are enabled to drive the GPIB IFC* signal.***

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
10 90	**Clear Send Remote Enable (~sre)** **Set Send Remote Enable (sre)** The ~sre and sre commands clear and set the sre and rsc local messages. A hardware reset or the ch_rst auxiliary command also clears sre. Setting sre and rsc causes the NAT7210 to become the System Controller and to assert the GPIB REN signal. The control program must not issue the sre command until after REN has been unasserted at least 100 µs. Remote Enable (REN) signal. The control program must not issue the sre command until after REN has been unasserted at least 100 µs. See the *System Controller Considerations* section in Chapter 6, *Controller Software Considerations.* **Note:** ***Before it issues the sre command, the control program must ensure—by some means external to the NAT7210—that the GPIB transceivers are enabled to drive the GPIB REN* signal.***
11	**Request Control (rqc)** If the NAT7210 is in the Idle Controller State, the rqc command forces the NAT7210 to become the Active Controller when it detects that the ATN signal is unasserted.
12	**Release Control (rlc)** The rlc command forces the NAT7210 to become an Idle Controller and to unassert ATN.
13 93	**Clear Disable IMR2, IMR1, And IMR0 Interrupts (~dai)** **Set Disable IMR2, IMR1, And IMR0 Interrupts (dai)** Issuing dai disables the interrupt pin. The Interrupt Status Registers and any holdoffs selected in the Interrupt Mask Register are not affected by the dai command. A hardware reset or the ch_rst auxiliary command clears dai. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
14	**Pass Through Next Secondary (pts)** After you issue the pts command, UNC (ISR1[5]) sets when the NAT7210 receives a secondary command from the Controller. If PP1 = 0, you can use the pts command to implement remote parallel poll configuration. **Note:** ***It is simpler to set the PP1 bit to implement remote parallel poll configuration. When PP1 = 1, the NAT7210 interprets remote parallel poll configuration commands without software intervention.*** If the NAT7210 receives the PPC command, UNC sets. When the control program detects UNC, the control program issues pts. UNC sets again when the Controller sends the PPE command. The control program reads the CPTR to obtain the PPE command, then the control program writes the appropriate value to the PPR.
15 95	**Clear Short T1 Delay (~stdl)** **Set Short T1 Delay (stdl)** Issuing stdl makes the T1 delay time 1.1 µs. A hardware reset or the ch_rst auxiliary command clears stdl. See the *T1 Delay Generation* section in Chapter 5, *Software Considerations*.
16 96	**Clear Shadow Handshaking (~shdw)** **Set Shadow Handshaking (shdw)** The shdw command places the NAT7210 in continuous mode. A hardware reset or the ch_rst auxiliary command clears shdw. See *The GPIB rdy Message and RFD Holdoffs* section in Chapter 5, *Software Considerations*.
17 97	**Clear Very Short T1 Delay (~vstdl)** **Set Very Short T1 Delay (vstdl)** Issuing vstdl reduces the T1 delay time to 500 ns. A hardware reset or the ch_rst auxiliary command clears vstdl. See the *T1 Delay Generation* section in Chapter 5, *Software Considerations*.

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
18 98	**Clear Request Service bit 2 (~rsv2)** **Set Request Service bit 2 (rsv2)** The rsv2 bit performs the same function as the rsv bit in the SPMR, but it provides a means of requesting service that is independent of the SPMR. With rsv2, you can make minor updates to the SPMR without affecting the state of service request. rsv2 is cleared when the serial poll status byte is sent to the Controller during a serial poll (SPAS & APRS & STRS). A hardware reset or the ch_rst auxiliary command also clears rsv2.
99	**Switch To 7210 Mode (sw7210)** Issuing sw7210 places the NAT7210 into 7210 compatibility mode.
1A 9A	**Request rsv False (reqf)** **Request rsv True (reqt)** The reqt and reqf commands are inputs to the IEEE 488.2 Service Request Synchronization Circuit. Use these commands to set and clear the local rsv message. The local message pon clears reqf and reqt. If STBO IE = 0, reqt and reqf are not issued immediately; they are issued on the write of the SPMR that follows the issuing of the reqt or reqf auxiliary command. If STBO IE = 1, reqt and reqf are issued immediately. The local message pon clears reqf and reqt. See the *IEEE 488.2 Service Requesting* section in Chapter 5, *Software Considerations*.

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
1C	**Chip Reset (ch_rst)** The Chip Reset command resets the NAT7210 to the following conditions: • The local swrst message is set and the interface functions are placed in their idle states. • SPMR bits are cleared. • EOS and NL bits are cleared. • ACCRA, ACCRB, ACCRE, ACCRF, and ACCRI registers are cleared. • The Parallel Poll Flag local message is cleared. • The ulpa bit is cleared.
1D 9D	**Clear Parallel Poll Flag (~ist)** **Set Parallel Poll Flag (ist)** The ~ist and ist commands set and clear the Parallel Poll Flag. The value of the Parallel Poll Flag is used as the local ist message when bit four of Accessory Register B (ISS) = 0. The value of SRQS is used as the local ist message when ISS = 1. A hardware reset or the ch_rst auxiliary command clears ist.
1E	**Page-In Interrupt Mask Register 2 (piimr2)** Issuing piimr2 maps Interrupt Mask Register 2 (IMR2) to the ADSR offset. After this command is issued, you can access IMR2 at the ADSR offset until one of the following events occurs: • A hardware reset occurs. • The ch_rst auxiliary command is issued. • Another register is paged into the ADSR offset. • The Clear Page-In auxiliary command is issued.

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

Data Pattern (Hex)	Description
1F	**Page-In Bus Control Register (pibcr)** Issuing pibcr maps the Bus Control Register (BCR) to the ADSR offset. After this command is issued, you can access BCR at the ADSR offset until one of the following events occurs: • A hardware reset occurs. • The ch_rst auxiliary command is issued. • Another register is paged into the ADSR offset. • The Clear Page-In auxiliary command is issued.
9C	**Clear Page-In Registers (clrpi)** Issuing clrpi removes the previously paged-in Accessory Register from the ADSR offset. After this command is issued, writes to offset 2 have no effect until a Page-In auxiliary command is issued.
9E	**Page-In End-of-String Register (pieosr)** Issuing pieosr maps the EOSR to the ADSR offset. After this command is issued, you can access the EOSR at the ADSR offset until one of the following events occurs: • A hardware reset occurs. • The ch_rst auxiliary command is issued. • Another register is paged into the ADSR offset. • The Clear Page-In auxiliary command is issued.

(continues)

9914 Mode Only

AUXCR (continued)

Table 4-4. Auxiliary Command Description (Continued)

9F	**Page-In Accessory Register (piaccr)** Issuing piaccr maps the ACCR to the ADSR offset. After this command is issued, you can access the ACCR at the ADSR offset until one of the following events occurs: • A hardware reset occurs. • The ch_rst auxiliary command is issued. • Another register is paged into the ADSR offset. • The Clear Page-In auxiliary command is issued.

9914 Mode Only

Bus Control Register (BCR)/Bus Status Register (BSR)

Attributes: Write only (BCR)
Read only (BSR)

7	6	5	4	3	2	1	0
ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN
ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN

Bit	Mnemonic	Description
7r	ATN	GPIB Attention Status bit
7w	ATN	GPIB Attention Control bit
6r	DAV	GPIB Data Valid Status bit
6w	DAV	GPIB Data Valid Control bit
5r	NDAC	GPIB Not Data Accepted Status bit
5w	NDAC	GPIB Not Data Accepted Control bit
4r	NRFD	GPIB Not Ready For Data Status bit
4w	NRFD	GPIB Not Ready For Data Control bit
3r	EOI	GPIB End-or-Identify Status bit
3w	EOI	GPIB End-or-Identify Control bit
2r	SRQ	GPIB Service Request Status bit
2w	SRQ	GPIB Service Request Control bit
1r	IFC	GPIB Interface Clear Status bit
1w	IFC	GPIB Interface Clear Control bit
0r	REN	GPIB Remote Enable Status bit
0w	REN	GPIB Remote Enable Control bit

Reads of the Bus Status Register (BSR) return the status of the GPIB control lines at the time of the read. Write ones to bits in the Bus Control Register (BCR) to assert the corresponding GPIB control lines.

9914 Mode Only

BCR/BSR (continued)

Because the NAT7210 is either transmitting or receiving a GPIB control line at any particular time and is not performing both actions simultaneously, setting a bit in the BCR may not automatically assert the corresponding line on the GPIB. If the NAT7210 is transmitting a GPIB line when the corresponding bit in the BCR is set, the NAT7210 asserts the GPIB line. If the NAT7210 is receiving a GPIB line when the corresponding bit in the BCR is set, the GPIB line is not asserted. However, in both these cases, the GPIB signal internal to the NAT7210 is logically ORed with the value of the BCR bit. Figure 4-1 illustrates the GPIB input/output hardware configuration.

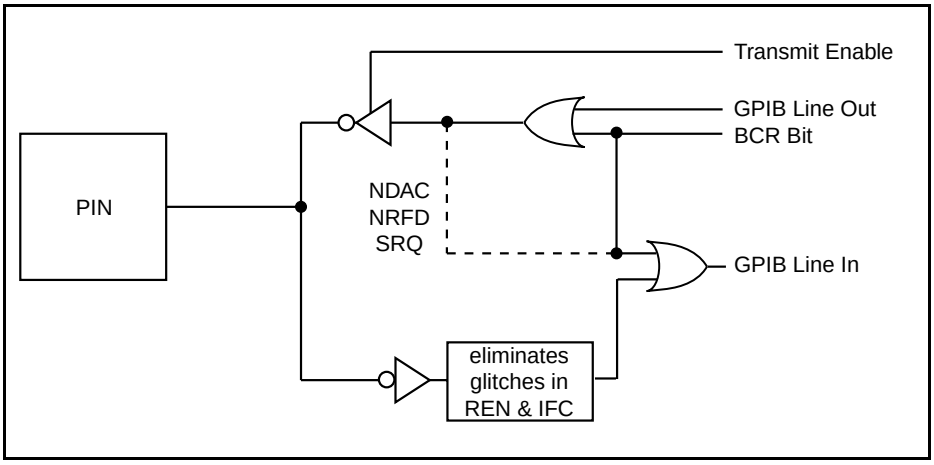


Figure 4-1. GPIB I/O Hardware Configuration

In Figure 4-1, *Transmit Enable* represents the internal signal that is true when the chip is driving a particular GPIB control line. *GPIB Line Out* represents the internal signal that is true when an interface function within the chip is attempting to assert a GPIB control signal. *BCR Bit* corresponds to the bit in the BCR. *GPIB Line In* represents the internal GPIB lines that are inputs to the GPIB interface functions and the BSR. The internal signals *SRQ*, *NDAC*, and *NRFD* are monitored by the interface functions even when they are not driven onto the pin. For this reason, the internal value of these signals is ORed with the external value.

Because the BSR samples the GPIB control lines from the GPIB transceiver—not the actual GPIB bus—the direction of each line determines the validity of each bit. Generally, when a signal is an input, the BSR reflects its true bus status, while an output signal reflects only the NAT7210 value of that particular line. Under normal GPIB operation, this restriction on the validity of the BSR should not be too limiting, because the lines that are typically monitored are valid when they are monitored. For example, the SRQ line is valid in the BSR when the NAT7210 is CIC, which is also when the SRQ line will be monitored.

9914 Mode Only

Command/Data Out Register (CDOR)

Attributes: Write only

7	6	5	4	3	2	1	0
DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

Bit	Mnemonic	Description
7–0w	DIO[8–1]	GPIB data lines DIO[8–1]

The CDOR moves data from the CPU to the GPIB when the interface is the GPIB Talker or Controller. Writing to the CDOR sets the local message, nba. When nba is true, the Source Handshake (SH) function can transfer the data or command in the CDOR to other GPIB devices. Writing to the CDOR also

- Clears the Byte Out (BO) bit.
- Clears the DRQ signal (unless DMAE = 1 and DMAO = 0).

The host interface can write to the CDOR at offset 7 or by performing a DMA write operation.

The CDOR and the DIR use separate latches. A read of the DIR does not change data in the CDOR. The CDOR is a transparent latch; thus, the GPIB data bus (DIO(8–1)) reflects changes on the CPU data bus during write cycles to the CDOR.

9914 Mode Only

Command Pass Through Register (CPTR)

Attributes: Read only

7	6	5	4	3	2	1	0
CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0

The host interface can examine the GPIB DIO lines by reading the Command Pass Through Register (CPTR). The CPTR has no storage; the host interface should read the CPTR only during a DAC holdoff. See the *DAC Holdoffs* section in Chapter 5, *Software Considerations.*

Bit	Mnemonic	Description
7–0r	CPT[7–0]	Command Pass Through bits 7 through 0

9914 Mode Only

Data In Register (DIR)

Attributes: Read only

7	6	5	4	3	2	1	0
DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

Bit	Mnemonic	Description
7–0r	DIO[8–1]	GPIB data lines DIO[8–1]

The DIR holds data that the NAT7210 receives when the NAT7210 is a Listener. The NAT7210 latches GPIB data into the DIR when LACS & ACDS is true.

Latching data into the DIR causes the DI bit to set. Usually, latching data into the DIR causes an RFD (Ready For Data) holdoff. (See *The GPIB rdy Message and RFD Holdoffs* section in Chapter 5, *Software Considerations*.)

The host interface can read the DIR at offset 7 or by performing a DMA read operation. Reading the DIR also

- Clears the BI bit.
- Can clear an RFD holdoff (depending on several other conditions).
- Clears the DRQ signal (unless DMAE = 1 and DMAI = 0).

The DIR and the CDOR use separate latches. When the host interface writes to the CDOR, data in the DIR is not changed.

9914 Mode Only

End-of-String Register (EOSR)

Attributes: Write only

7	6	5	4	3	2	1	0
EOS7	EOS6	EOS5	EOS4	EOS3	EOS2	EOS1	EOS0

The End-of-String Register (EOSR) holds the byte that the NAT7210 uses to detect the end of a GPIB data block transfer. The NAT7210 compares data it receives to a 7- or 8-bit byte (ASCII or binary—depending on the BIN bit) in the EOSR in order to detect the end of a block of data.

If the NAT7210 is a Listener and REOS = 1, the END bit is set in ISR0 whenever the received data byte matches the EOSR. If the NAT7210 is a Talker and XEOS = 1, the END message (GPIB EOI* line asserted low) is sent along with a data byte whenever the data byte matches the EOSR.

Bit	**Mnemonic**	**Description**
7–0w	EOS[7–0]	End-of-String bits 7 through 0

9914 Mode Only

Internal Count Register (ICR)

Attributes: Write only

7	6	5	4	3	2	1	0
0	0	1	0	F3	F2	F1	F0

The Internal Count Register (ICR) determines the internal clock frequency of the NAT7210. You write to the ICR at the same offset as the ACCR.

Note: ***The ICR resets to 00101000 (8 MHz).***

Bit **Mnemonic** **Description**

3–0 F(3–0) Clock Frequency

These bits, in addition to MICR (ICR2[0]), determine the length of certain delays that are required by the IEEE 488 standard. You should set these bits according to the frequency of the signal driving the CLK pin. For proper operation, set F(3–0) and MICR as follows:

Clock Frequency	MICR	F(3–0)
1	0	0001
2	0	0010
3	0	0011
4	0	0100
5	0	0101
6	0	0110
7	0	0111
8	0	1000
10	1	0101
16	1	1000
20	1	1010

9914 Mode Only

Interrupt Mask Register 0 (IMR0)

Attributes: Write only

7	6	5	4	3	2	1	0
DMAO	DMAI	BI IE	BO IE	END IE	SPAS IE	RLC IE	MAC IE

Interrupt Status Register 0 (ISR0)

Attributes: Read only
Bits are cleared when read

7	6	5	4	3	2	1	0
INT0	INT1	BI	BO	END	SPAS	RLC	MAC

Interrupt Status Register 0 (ISR0) contains Interrupt Status bits. Interrupt Mask Register 0 (IMR0) contains Interrupt Enable bits that directly correspond to the Interrupt Status bits in ISR0. As a result, ISR0 and IMR0 service six possible interrupt conditions; each condition has an associated Interrupt Status bit and an Interrupt Enable bit. If an Interrupt Enable bit is true when the corresponding status condition or event occurs, the NAT7210 can generate a hardware interrupt request. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

Bits in ISR0 are set and cleared regardless of the status of the Interrupt bits in IMR0. If an interrupt condition occurs at the same time the host interface is reading ISR0, the NAT7210 does not set the corresponding Interrupt Status bit until the read is finished. A hardware reset clears all bits in IMR0.

Bit	Mnemonic	Description
7r	INT0	Interrupt Register 0 Interrupt bit INT0 is set when an unmasked status bit in ISR0 is set.
7w	DMAO	DMA Output Enable bit If DMAE = 1 (ACCRI[0]), DMAO enables the NAT7210 to assert the DRQ pin as a GPIB Talker. The NAT7210 asserts DRQ when it is ready to accept another byte in the CDOR. DRQ does not assert if the NAT7210 is not a Talker. If DMAE = 0, write 0 to DMAO.

9914 Mode Only

IMR0/ISR0 (continued)

Bit	Mnemonic	Description
6r	INT1	Interrupt Register 1 Interrupt bit INT1 is set when an unmasked status bit in Interrupt Status Register 1 (ISR1) is set.
6w	DMAI	DMA Input Enable bit If DMAE = 1 (ACCRI[0]), DMAI enables the NAT7210 to assert the DRQ pin as a GPIB Listener. The NAT7210 asserts DRQ when the DIR contains a byte for the host interface to read. If DMAE = 0, write 0 to DMAI.
5r	BI	Byte In bit
5w	BI IE	Byte In Interrupt Enable bit BI indicates that a data byte has been received in the Data In Register. An RFD holdoff must be cleared before the NAT7210 accepts the next data byte. BI is set by LACS & ACDS & ~(continuous mode) BI is cleared by swrst + (read ISR0) + (read DIR)
4r	BO	Byte Out bit
4w	BO IE	Byte Out Interrupt Enable bit BO indicates that the NAT7210 is the Active Controller or Talker and that the CDOR does not contain a byte to send over the GPIB. BO sets again after each byte has been sent and the source handshake has returned to SGNS. BO is set by (CACS + TACS) & SGNS & ~nba BO is cleared by swrst + (read ISR0) + (write CDOR)

9914 Mode Only

IMR0/ISR0 (continued)

Bit	Mnemonic	Description
3r	END	End Received bit
3w	END IE	End Received Interrupt Enable bit
		END sets when the NAT7210, as a Listener, receives a data byte satisfying the END condition. A data byte satisfies the END condition if one of the following conditions is true: • REOS = 1 and the data byte matches the contents of the EOSR. • NLEN = 1 and the data byte matches the ASCII new line character (hex 0A). • The GPIB EOI signal is asserted when the byte is received. That is, END is set by (EOI + EOS & REOS + NL & NLEN) & LACS & ACDS END is cleared by swrst + (read ISR0)
2r	SPAS	Serial Poll Active State bit
2w	SPAS IE	Serial Poll Active State Interrupt Enable bit
		SPAS indicates that the Controller has serial polled the NAT7210 in response to the NAT7210 requesting service. SPAS is set by [STRS & SPAS & APRS] becoming false SPAS is cleared by swrst + (read ISR0)
1r	RLC	Remote/Local Change bit
1w	RLC IE	Remote/Local Change Interrupt Enable bit
		RLC is set when a change occurs in the REM bit, ADSR[7]r.

9914 Mode Only

IMR0/ISR0 (continued)

Bit	Mnemonic	Description
		RLC is cleared by swrst + (read ISR0)
0r	MAC	My Address Change bit
0w	MAC IE	My Address Change Interrupt Enable bit
		MAC indicates that the NAT7210 has received a command from the Controller and that this command has changed the addressed state of the NAT7210. If the NAT7210 is using secondary addressing, MAC sets only when the NAT7210 becomes unaddressed. If edpa = 1, MAC does not set when the Controller readdresses the NAT7210 at the NAT7210's other primary address. MAC is set by ACDS & (MTA & ~TADS & ~APT IE + OTA & TADS + MLA & ~LADS & ~APT IE + UNL & LADS) MAC is cleared by swrst + (read ISR0)

9914 Mode Only

Interrupt Mask Register 1 (IMR1)

Attributes: Write only

7	6	5	4	3	2	1	0
GET IE	ERR IE	UNC IE	APT IE	DCAS IE	MA IE	SRQ IE	IFC IE

Interrupt Status Register 1 (ISR1)

Attributes: Read only
Bits are cleared when read

7	6	5	4	3	2	1	0
GET	ERR	UNC	APT	DCAS	MA	SRQ	IFC

Interrupt Status Register 1 (ISR1) contains Interrupt Status bits. Interrupt Mask Register 1 (IMR1) contains Interrupt Enable bits that directly correspond to the Interrupt Status bits in ISR1. As a result, ISR1 and IMR1 service interrupt conditions; each condition has an associated Interrupt Status bit and an Interrupt Enable bit. If an Interrupt Enable bit is true when the corresponding status condition or event occurs, the NAT7210 can generate a hardware interrupt request. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

Bits in ISR1 are set and cleared regardless of the status of the Interrupt bits in IMR1. If an interrupt condition occurs at the same time the host interface is reading ISR1, the NAT7210 does not set the corresponding Interrupt Status bit until the read is finished. A hardware reset clears all bits in IMR1.

The interrupts GET, UNC, APT, DCAS, and MA are set in response to commands received over the bus. If the corresponding Interrupt Enable bit is set, a DAC holdoff occurs when the interrupt sets.

Bit	Mnemonic	Description
7r	GET	Group Execute Trigger bit
7w	GET IE	Group Execute Trigger Interrupt Enable bit
		GET indicates that the NAT7210 received the GPIB GET command while the NAT7210 was a GPIB Listener.

9914 Mode Only

IMR1/ISR1 (continued)

Bit	Mnemonic	Description
		If GET IE = 1, a DAC holdoff occurs when the interrupt condition occurs. The TRIG pin goes high when the interrupt condition occurs and remains high until the DAC holdoff is released. If GET IE = 0, the TRIG pin pulses high. GET is set by GET & LADS & ACDS GET is cleared by swrst + (read ISR1) **Note:** ***TRIG pin refers to the T/R3 pin when TRM1 = 0. If TRM1 does not equal 1, the TRIG signal is not visible at a pin. See the*** *Address Mode Register (ADMR)* ***section in Chapter 3,*** *7210-Mode Interface Registers.*
6r 6w	ERR ERR IE	Error bit Error Interrupt Enable bit ERR sets when the Source Handshake becomes active (enters SDYS) and finds that the NDAC and NRFD lines are both unasserted on the GPIB. This condition indicates that there are no acceptors on the GPIB. ERR is set by SDYS & EXTDAC & RFD ERR is cleared by swrst + (read ISR1)
5r 5w	UNC UNC IE	Unrecognized Command bit Unrecognized Command Interrupt Enable bit UNC flags the occurrence of several types of GPIB commands. UNC sets when the NAT7210 accepts any unrecognized Universal Command Group (UCG) command.

9914 Mode Only

IMR1/ISR1 (continued)

Bit	Mnemonic	Description
		If the NAT7210 is an addressed Listener, UNC sets when the NAT7210 accepts any unrecognized Addressed Command Group (ACG) command. UNC flags the first secondary command that the NAT7210 accepts after the host interface issues the Pass Through Next secondary auxiliary command. UNC can also flag the occurrence of commands that you specify when you set the AUXRE[3–2]w or AUXRF[3–0]w bits. If UNC IE = 1, the NAT7210 performs a DAC holdoff when UNC sets. The host interface releases the DAC holdoff by issuing the Release DAC Holdoff auxiliary command. Read undefined commands by using the Command Pass Through Register (CPTR). UNC is set by ACDS & UCG & ~(LLO + SPE + SPD + DCL + PPU & PP1) + ACDS & ACG & ~(GET + GTL + SDC + TCT + PPC & PP1) & LADS + SCG & PTS & ACDS + DHADT & GET & ACDS + DHADC & (SDC + DCL) & ACDS + DHATA & TAG & ~UNT & ACDS + DHALA & LAG & ~UNL & ACDS + DHUNTL & (UNT + UNL) & ACDS + DHALL & ATN & ACDS UNC is cleared by swrst + (read ISR1)
4r 4w	APT APT IE	Address Pass Through bit Address Pass Through Interrupt Enable bit Setting APT IE enables secondary addressing. If the last primary command accepted was a primary talk or listen address of the NAT7210, APT sets when the NAT7210 accepts a secondary command. The secondary command is a secondary GPIB address that can be read in the CPTR. **Note:** ***When the host interface uses secondary addressing, it must check APT.***

9914 Mode Only

IMR1/ISR1 (continued)

Bit	Mnemonic	Description
		If APT IE = 1, the NAT7210 performs a DAC holdoff when APT sets. The host interface releases the DAC holdoff by issuing the Release DAC Holdoff auxiliary command. APT is set by (TPAS + LPAS) & SCG & ACDS APT is cleared by swrst + (read ISR1)
3r 3w	DCAS DCAS IE	Device Clear Active State bit Device Clear Active State Interrupt Enable bit DCAS indicates that either the NAT7210 received the GPIB Device Clear (DCL) command or that the NAT7210 was a Listener and received the GPIB Selected Device Clear (SDC) command. If DCAS IE = 1, the NAT7210 performs a DAC holdoff when DCAS sets. The host interface releases the DAC holdoff by issuing the Release DAC Holdoff auxiliary command. DCAS is set by ACDS & (DCL + SDC & LADS) DCAS is cleared by swrst + (read ISR1)
2r 2w	MA MA IE	My Address bit My Address Interrupt Enable bit MA sets when the NAT7210 accepts its primary talk or listen address. If MA IE = 1, the NAT7210 performs a DAC holdoff when MA sets. The host interface releases the DAC holdoff by issuing the Release DAC Holdoff auxiliary command.

9914 Mode Only

IMR1/ISR1 (continued)

Bit	Mnemonic	Description
		MA is set by (MLA + MTA) & ACDS & ~SPMS & ~APT IE MA is cleared by swrst + (read ISR1)
1r 1w	SRQ SRQ IE	Service Request bit Service Request Interrupt Enable bit The SRQ bit indicates that the NAT7210 received a GPIB SRQ message while the NAT7210 was the CIC. The SRQ bit is cleared by swrst + (read ISR1)
0r 0w	IFC IFC IE	Interface Clear bit Interface Clear Interrupt Enable bit IFC sets on the assertion of the GPIB IFC signal. IFC is cleared by swrst + (read ISR1)

9914 Mode Only

Interrupt Mask Register 2 (IMR2)

Attributes: Write only

7	6	5	4	3	2	1	0
GLINT	STBO IE	NLEN	0	LLOC IE	ATNI IE	0	CIC IE

Interrupt Status Register 2 (ISR2)

Attributes: Read only

7	6	5	4	3	2	1	0
nba	STBO	NL	EOS	LLOC	ATNI	X	CIC

Interrupt Status Register 2 (ISR2) contains Interrupt Status bits and Internal Status bits. Interrupt Mask Register 2 (IMR2) contains Interrupt Enable bits and Internal Control bits. As a result, ISR2 and IMR2 service several possible interrupt conditions; each condition has an associated Interrupt Status bit and an Interrupt Enable bit. If an Interrupt Enable bit is true when the corresponding status condition or event occurs, the NAT7210 can generate a hardware interrupt request. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

Bits in ISR2 are set and cleared regardless of the status of the Interrupt bits in IMR2. If an interrupt condition occurs at the same time the host interface is reading ISR2, the NAT7210 does not set the corresponding Interrupt Status bit until the read is finished. A hardware reset clears all bits in IMR2 except GLINT.

Bit	Mnemonic	Description
7r	nba	New Byte Available local message bit nba is true when the local variable nba is true. nba is set on writes to the CDOR and cleared on entrance to STRS, pon, or nbaf.
7w	GLINT	Global Interrupt Enable bit GLINT enables the NAT7210 to assert the INT pin. If GLINT = 0, INT does not assert. See the *Generating Hardware Interrupts* section in Chapter 5, *Software Considerations*.

9914 Mode Only

IMR2/ISR2 (continued)

Bit	Mnemonic	Description
6r	STBO	Status Byte Out bit
6w	STBO IE	Status Byte Out Interrupt Enable bit

STBO is set when the NAT7210 enters SPAS when STBO IE = 1. After STBO sets, the control program should write the current STB to the SPMR. The current STB is then transmitted to the GPIB as the STB. Writing the SPMR clears STBO.

STBO IE determines how the NAT7210 requests service and responds to serial polls.

If STBO IE = 0, the rsv bit in the SPMR can be used to request service. When the GPIB Controller serial polls the NAT7210, the NAT7210 transmits the current value of the SPMR.

If STBO IE = 1, the rsv bit in the SPMR has no effect on the SR1 function and rsv must be generated through the reqt auxiliary command. When the GPIB Controller serial polls the NAT7210, STBO sets. In response to STBO, the host interface writes a byte to the SPMR, then the NAT7210 transmits this byte as the Serial Poll response.

STBO is set by
STBO IE & SPAS

STBO is cleared by
swrst + (write SPMR) + ~SPAS

5r	NL	New Line Receive bit

NL indicates that the last data byte that the NAT7210 received was an ASCII new line character.

NL is set by
LACS & NL & ACDS

NL is cleared by
swrst + (LACS & ~NL & ACDS)

9914 Mode Only

IMR2/ISR2 (continued)

Bit	Mnemonic	Description
5w	NLEN	New Line End Enable bit If NLEN = 1, the NAT7210 treats the 7-bit ASCII new line character (0A hex) as an EOS character. The Acceptor Handshake function responds to the acceptance of a new line character in the same manner as if EOI were sent.
4r	EOS	End-of-String bit EOS indicates that REOS = 1 and that the last data byte the NAT7210 received matched the contents of the EOSR. EOS is set by LACS & EOS & REOS & ACDS EOS is cleared by swrst + (LACS & ~EOS & ACDS) + ~REOS
3r 3w	LLOC LLOC IE	Local Lockout Change bit Local Lockout Change Interrupt Enable bit LLOC is set by any change in the LLO bit LLOC is cleared by ch_rst + (read ISR0)
2r 2w	ATNI ATNI IE	ATN Interrupt bit ATN Interrupt Enable bit ATN is set by (ATN) becoming true ATN is cleared by ch_rst + read ISR0
1r	X	Don’t care bit

9914 Mode Only

IMR2/ISR2 (continued)

Bit	Mnemonic	Description
0r	CIC	Controller-In-Charge bit
0w	CIC IE	Controller-In-Charge Interrupt Enable bit
		CIC indicates whether the NAT7210 is the Controller-in-Charge.
		CIC = ~(CIDS + CADS)

9914 Mode Only

Parallel Poll Register (PPR)

Attributes: Write only

7	6	5	4	3	2	1	0
PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1

Bit	Mnemonic	Description
7–0w	PP8–PP1	When a Controller initiates a parallel poll, the NAT7210 drives the contents of the PPR on the GPIB DIO lines using open-collector drivers. If PP8–PP1 = 00 (hex), none of the lines (DIO(8–1)) are asserted during a parallel poll. The PPR is double buffered. If the PPR is written during a parallel poll, the new value is held until the parallel poll ends. When the parallel poll ends, the register is updated, so the control program can update the parallel poll response asynchronously to the GPIB. A hardware reset or the ch_rst auxiliary command clears PPR. The host interface can load PPR while swrst = 1.

9914 Mode Only

Serial Poll Mode Register (SPMR)

Attributes: Write only

7	6	5	4	3	2	1	0
S8	rsv/RQS	S6	S5	S4	S3	S2	S1

Serial Poll Status Register (SPSR)

Attributes: Read only

7	6	5	4	3	2	1	0
S8	PEND	S6	S5	S4	S3	S2	S1

Bit	Mnemonic	Description
7r, 7w	S8	Serial Poll Status bit 8
5–0r, 5–0w	S[6–1]	Serial Poll Status bits 6 through 1 These bits send device- or system-dependent status information over the GPIB when the Controller serial polls the NAT7210. When STBO IE = 0, the NAT7210 transmits a byte of status information, SPMR[7–0], to the CIC if the CIC serial polls the NAT7210. The SPMR bits S[8, 6–1] are double buffered. If the host interface writes to the SPMR during a serial poll when SPAS is active, the NAT7210 saves the value. The NAT7210 updates the SPMR when the NAT7210 exits SPAS. When STBO IE = 1 and the Controller serial polls the NAT7210, the STBO interrupt condition sets. The host interface should write the STB and the RQS bit to the SPMR in response to an STBO interrupt. Issuing the ch_rst auxiliary command clears these bits.

9914 Mode Only

SPMR/SPSR (continued)

Bit	Mnemonic	Description
6r	PEND	Pending bit PEND sets when rsv = 1. PEND clears when the NAT7210 is in the Negative Poll Response State (NPRS) and the local Request Service (rsv) message is false. By reading the PEND status bit, you can confirm that a request was accepted and that the Status Byte (STB) was transmitted (PEND = 0).
6w	rsv/RQS	Request Service/ RQS bit When STBO IE = 0, bit 6 is the rsv bit. The rsv bit generates the GPIB local rsv message. When rsv = 1 and the GPIB Controller is not serial polling the NAT7210, the NAT7210 enters the Service Request State (SRQS) and asserts the GPIB SRQ signal. When the Controller reads the STB during the poll, the NAT7210 clears rsv. The rsv bit is also cleared by a hardware reset or by writing 0 to it. Issuing the ch_rst auxiliary command also clears rsv. When STBO IE = 1, bit 6 is the RQS bit. When the Controller serial polls the NAT7210, the STBO interrupt condition sets. The host interface should write the STB and the RQS bit to the SPMR in response to an STBO interrupt. The NAT7210 transfers the STB and RQS to the Controller during that particular serial poll. A hardware reset clears RQS. Issuing the ch_rst auxiliary command also clears RQS.

Chapter 5
Software Considerations

The chapter explains important NAT7210 programming considerations, including chip initialization, Talkers and Listeners, message reception, and holdoffs. You should also refer to Chapter 6, *Controller Software Considerations*, for applications that use the NAT7210 GPIB Controller functions.

This chapter, except where explicitly noted, assumes that the NAT7210 uses 7210-mode architecture and that the reader is familiar with the GPIB. For more information about GPIB, refer to Appendix B, *Introduction to the GPIB*, and the IEEE 488.1 and IEEE 488.2 standard.

Chip Initialization Sequence

1. Place the NAT7210 in 7210 Mode

Place the NAT7210 in 7210 mode by performing one of the following actions:

- Assert the RESET* pin.
- Write the sw7210 command to the AUXCR if the NAT7210 is in 9914 mode.
- Write 99 (hex) to offset 3 if you do not know whether the NAT7210 is in 7210 mode or 9914 mode. If the chip is in 9914 mode, this write places it in 7120 mode. If the chip is already in 7210 mode, this write access writes to the 7210-mode SPMR.

2. Make Sure the Local *pon* Message Is Asserted

Write the Chip Reset auxiliary command (2 hex) to the AUXMR to assert the local pon message. When pon is asserted, the chip is logically disconnected from the GPIB and the GPIB interface functions of the NAT7210 are idle (and ignore GPIB signals).

3. Set the Clock Frequency

Set the clock frequency of the NAT7210 by writing to the ICR and the ICR2.

4. Configure the Chip for GPIB Operation

A. Set the GPIB Address(es)

Write to the ADMR to configure the GPIB addressing mode of the NAT7210. In most applications, you write 31 (hex) to the ADMR to set the dual primary addressing mode. Load the primary GPIB address of the NAT7210 into internal Address Register 0 (ADR0) by writing to the Address register. If the device has only one GPIB address, disable ADR1 by writing E0 (hex) to the Address Register.

Note: ***If the device has only one GPIB address, disable ADR1 by writing E0 (hex) to the Address Register.***

B. Write the Initial Serial Poll Response

If another device in your system has Controller capabilities, write the initial serial poll response of the NAT7210 to the SPMR. See the *Serial Polling* section in Appendix B, *Introduction to the GPIB.*

C. Configure Initial Parallel Response

If another device in your system has Controller capabilities, configure the initial parallel poll response of the NAT7210 by writing to AUXRI and PPR. (See the *Parallel Polling* section in Appendix B, *Introduction to the GPIB*.)

D. Set GPIB Handshake Parameters

1. Set the Source Handshake T1 delay by setting or clearing the TRI and USTD bits. (See the *T1 Delay Generation* section, which is located later in this chapter.)

2. Set CHES to enable the clearing of the END detection circuitry on the reception of a data byte without END.

3. Set NTNL to prevent the NAT7210 from sourcing data or commands when no Listeners are on the GPIB.

4. Issue the hldi auxiliary command to make the NAT7210 perform an RFD holdoff if it is addressed to listen.

5. Enable Interrupts

Set or clear the appropriate Interrupt Enable bits in IMR0, IMR1, and IMR2. Set the SISB bit, if desired.

6. Clear the Local *pon* Message

Issue the pon auxiliary command to clear the local pon message and to begin GPIB operation.

GPIB Talker or Listener Considerations

GPIB Addressing

Logical and Physical Devices

When you connect the NAT7210 to the GPIB by using standard GPIB transceivers, the NAT7210 is one physical GPIB device. The IEEE 488.1 transceiver places a single physical device load on the GPIB. The IEEE 488.1 standard specifies that a GPIB system contain no more than 15 physical devices.

In most applications, the NAT7210 is also one logical device. However, a single physical GPIB device can implement more than one logical GPIB device. Each logical device must have a unique GPIB address. The NAT7210 can implement up to 961 logical GPIB devices.

Normal and Extended Addressing

Logical GPIB devices use either normal or extended addressing. With normal addressing, a GPIB device has a single address; valid addresses are 0 through 30 (decimal), inclusive. To address a device to become a Talker or Listener, a Controller sends the talk or listen address of the device. If a device's address is 6, for example, a Controller sends the My Talk Address 6 (MTA6) message to address that device to become a Talker.

With extended addressing, a GPIB device has two addresses: a primary address and a secondary address. Valid primary addresses are 0 through 30 (decimal), inclusive; valid secondary addresses are also 0 through 30 (decimal), inclusive. With extended addressing, 961 (decimal) unique GPIB addresses exist. To address a device to become a Talker or Listener, a Controller sends the primary talk or listen address of the device, then sends the secondary address of the device.

Implementing One Logical Device: Normal Addressing

The NAT7210 can implement one logical device that uses normal addressing. The NAT7210 can become an Addressed Listener or Talker without the intervention of the host interface. The TA bit in ADSR sets when the NAT7210 is an Addressed Talker, and the Listener Active (LA) bit in ADSR sets when the NAT7210 is an Addressed Listener.

Complete the following steps to implement one logical device that uses normal addressing:

1. Choose the normal dual addressing mode by writing a 31 (hex) to the ADMR.

2. Write the logical address to ADR0.

3. Disable ADR1 by setting the Disable Talker (DT) and Disable Listener (DL) bits in ADR1—that is, write a hex E0 to offset hex 6. Notice that ADR1 and ADR0 both appear at offset 6.

Implementing One Logical Device: Extended Addressing

The NAT7210 can implement one logical device that uses extended addressing. The NAT7210 can become an Addressed Listener or Talker without the intervention of the host interface. When the Controller sends the primary talk or listen address of the NAT7210, the Talker Primary Addressed State (TPAS) bit or the Listener Primary Addressed State (LPAS) bit in the ADSR sets. When the Controller sends the secondary address of the NAT7210, the TA bit and the LA bit in the ADSR set.

Complete the following steps to implement one logical device that uses extended addressing:

1. Choose the extended single addressing mode by writing a 32 (hex) to the ADMR.

2. Write the primary and secondary addresses to ADR0 and ADR1, respectively. Notice that ADR1 and ADR0 both appear at offset 6.

Implementing Two Logical Devices: Normal Addressing

The NAT7210 can implement two logical devices that use normal addressing. The NAT7210 can become an Addressed Listener or Talker for either of these devices without the intervention of the host interface. The TA bit in ADSR sets when the NAT7210 is an Addressed Talker, and the LA bit in ADSR sets when the NAT7210 is an Addressed Listener. The Major-Minor (MJMN) bit in ADSR indicates which of the two devices is addressed.

This mode requires one logical address for each device: the major device address and the minor device address. *Major* and *minor* distinguish between the two devices and do not denote the priority of one device over the other.

Complete the following steps to implement two logical devices that use normal addressing:

1. Choose the normal dual addressing mode by writing a 31 (hex) to the ADMR.

2. Write the major address to ADR0 and write the minor address to ADR1. Notice that ADR1 and ADR0 both appear at offset 6.

Implementing Two Logical Devices: Extended Addressing

The NAT7210 can implement two logical devices that use extended addressing. The NAT7210 can become an Addressed Talker or Listener only after the Controller sends the primary and secondary addresses of one of the two logical devices. The two logical devices are the major logical device and the minor logical device.

This mode requires intervention from the host interface. Complete the following steps to implement two logical devices that use extended addressing:

1. Choose the extended dual addressing mode by writing a 33 (hex) to the ADMR.

2. Write the primary address of the major device to the ADR0 and write the primary address of the minor device to the ADR1. The host interface stores the secondary addresses of the NAT7210 external to the NAT7210, then the following sequence of events occurs:

    - The Controller sends the primary talk or listen address of the NAT7210 to the GPIB.

    - The NAT7210 enters the TPAS or LPAS state. The MJMN bit sets or clears to indicate the reception of the minor or major primary address.

    - The Controller sends a secondary address to the GPIB.

    - The Address Pass Through (APT) bit sets (see ISR1).

    - The NAT7210 performs a Data Accepted (DAC) holdoff.

    - The host interface reads the Command Pass Through Register (CPTR) to determine whether the Controller sent the secondary address of the NAT7210.

    - If the host interface determines that the Controller sent the secondary address of the NAT7210, it issues the valid auxiliary command, and the NAT7210 becomes addressed.

    - If the host interface determines that the Controller sent the secondary address of another device, it issues the nonvalid auxiliary command.

Implementing Three or More Logical Devices: Normal Addressing

The NAT7210 can implement three or more logical devices that use normal addressing. This mode requires intervention from the host interface. Refer to the Talker function in the IEEE 488.1 standard.

Complete the following steps to implement three logical devices that use normal addressing:

1. Choose the no-addressing mode by writing a 30 (hex) to the ADMR. The host interface stores the addresses of the NAT7210 external to the NAT7210.

2. Set the following bits in Auxiliary Register F (AUXRF): DAC Holdoff On All Talker Addresses (DHATA) bit and DAC Holdoff On All Listener Addresses (DHALA) bit. The following sequence of events then occurs:

   - The Controller sends a talk or listen address to the GPIB.
   - The Command Pass Through (CPT) bit sets (see ISR1).
   - The NAT7210 performs a DAC holdoff.

3. Wait for the CPT bit to set. CPT sets when the Controller sends any talk or listen address over the GPIB.

4. Read the CPTR to determine whether the Controller sent one of the talk or listen addresses of the NAT7210.

5. If the CPTR matches one of the talk addresses of the NAT7210, the following sequence programs the NAT7210 to be the Addressed Talker:

   - Write B0 to the ADMR.
   - Write 30 to the ADMR.

   If the CPTR matches one of the listen addresses of the NAT7210, the following sequence programs the NAT7210 to be the Addressed Listener:

   - Write 70 to the ADMR.
   - Write 30 to the ADMR.

6. Write the valid auxiliary command to the AUXMR. The NAT7210 performs a DAC holdoff on the command byte the Controller sends. The valid auxiliary command releases the DAC holdoff.

Implementing Three or More Logical Devices: Extended Addressing

The NAT7210 can implement three or more logical devices that use extended addressing. The required steps are similar to the steps for implementing three or more logical devices that use normal addressing. By using CPT and CPTR, the host interface monitors all commands. The host interface addresses and unaddresses the NAT7210 as needed. See the Extended Talker function in the IEEE 488.1 standard.

Complete the following steps to implement three logical devices that use extended addressing:

1. Choose the no-addressing mode by writing a 30 (hex) to the ADMR. The host interface stores the addresses of the NAT7210 external to the NAT7210.

2. Set the DAC Holdoff On All UCG, ACG, and SCG Commands (DHALL) bit in AUXRF.

Programmed Implementation of a Talker and Listener

When no Controller is in the GPIB system, you can use the ton and lon address modes to activate the NAT7210 GPIB Talker and Listener functions (refer to the *Address Mode Register* section in Chapter 3, *7210-Mode Interface Registers*). Set the ton or lon mode during NAT7210 initialization.

Detecting a GPIB Listener

When the NAT7210 is the Active Talker (and not an Active Listener), it can determine whether a Listener is in the GPIB system. Simply read the BCR. If either NDAC or NRFD is asserted in the BCR, an Active Listener is in the system.

Sending GPIB Data Messages

Basic Flow

Complete the following steps to send GPIB (device-dependent) data to another device in your system:

1. Be certain that NTNL = 1 (AUXRG[3]). NTNL prevents the NAT7210 from sourcing commands if there are no listening devices in the system.

2. Be certain that the you have enabled the desired T1 delay. (See the *T1 Delay Generation* section, which is located later in this chapter.)

3. Wait until the NAT7210 has been programmed or addressed to be the Active Talker. Poll or interrupt on the ADSC (ISR2[0]) condition, then read the ADSR to determine whether the NAT7210 is a Talker.

4. Wait for the DO bit to set.

5. Perform the following steps for each data byte that is sent:

   - Write the data byte to the CDOR.
   - Wait for either the DO bit or the ERR (ISR1[2]) bit to set. DO indicates that all devices in the system have accepted the command. ERR indicates that no properly operating GPIB devices are in the system.

   The host interface can either poll the DO and ERR bits or configure the NAT7210 to interrupt on DO and ERR by setting the DO IE and ERR IE bits.

   If the ERR bit sets, issue the nbaf auxiliary command to clear any byte that may be in the CDOR, then issue the ClearERR auxiliary command to clear the ERR bit.

Sending EOI or EOS

To send the GPIB END message and a data byte, issue the SendEOI auxiliary command just before you write the byte to the CDOR. To send the EOS message, simply make the last byte that you write to the CDOR the End-of-String (EOS) code. You can also use the XEOS bit (AUXRA[3]) to send the END message whenever the byte in the CDOR matches the End-of-String Register (EOSR).

Using DMA

To use DMA to send data bytes, set the DMAO bit after the NAT7210 has been addressed to talk. The DRQ pin of the NAT7210 asserts whenever the CDOR is empty. The DMA Controller should respond to DRQ by asserting the DACK* and WR* pins to place a data byte into the CDOR.

T1 Delay Generation

The T1 Delay

When the NAT7210, as a GPIB Talker, transfers data bytes to GPIB Listeners, it drives the data byte on the GPIB DIO[8–1] signals. After waiting for a certain delay (known as the T1 delay), the NAT7210 asserts DAV to indicate to the Listeners that the data byte has settled on the DIO[8–1] signals.

HSTS Definition

The length of the T1 delay depends on several factors. One factor is the internal HSTS signal of the NAT7210. HSTS clears when the GPIB Controller asserts the GPIB ATN signal. HSTS sets after the NAT7210, as a GPIB Talker, transfers a byte. Usually, the T1 delay is longer for the first data byte of a transfer (HSTS = 0). The T1 delay is shorter for the second byte of a transfer (and for subsequent bytes).

IEEE 488.1 Standard Requirements

The IEEE 488.1 standard describes minimum T1 delay requirements for three cases:

- **Case 1**: The Talking device uses three-state drivers on the DIO, DAV, and EOI signals. Usually, the NAT7210 is connected to the GPIB with 75160 and 75162 transceivers. In the typical configuration, the 75160 and 75162 transceivers use three-state drivers on the DIO, DAV, and EOI signals.

- **Case 2**: The Talking device satisfies the requirements of case 1 and uses 48-mA three-state drivers (again, the 75160 and 75162 satisfy this requirement). In addition, the following requirements are placed on the GPIB system (and not just on the Talking device):

  – All devices in the system must be powered on.

  – The maximum cable length for the system is either 15 m or 1 m times the number of equivalent device loads in the system—whichever is less.

  – The device capacitance on each signal should be less that 50 pF per device (again, the 75160 and 75162 satisfy this requirement).

- **Case 3**: All other configurations.

Table 5-1 shows the IEEE 488.1 minimum T1 delay requirements for the three cases.

Table 5-1. IEEE 488.1 Minimum T1 Delay Requirements

Case Number	First Byte (HSTS = 0, or ATN asserted)	Other Bytes (HSTS = 1)
Case 1	1100 ns	500 ns
Case 2	1100 ns	350 ns
Case 3	2000 ns	2000 ns

T1 Delay: 7210 Mode

In 7210 mode, the USTD bit (AUXRI[3]), the B2 bit (AUXRB[2]), and the HSTS signal determine the NAT7210 T1 delay. Table 5-2 shows the T1 delay for various settings.

Table 5-2. T1 Delay Settings in 7210 Mode

HSTS	USTD	TRI	T1 Delay (ns)
0	0	X	2000
0	1	X	1100
1	0	0	2000
1	1	0	1100
1	0	1	500
1	1	1	350

T1 Delay: 9914 Mode

In 9914 mode, the USTD bit (ACCRI[3]), the vstdl bit (AUXCR), the stdl bit (AUXCR), and the HSTS signal determine the NAT7210 T1 delay (see the *HSTS Definition* section in this chapter). Table 5-3 shows the T1 delay for various settings.

Table 5-3. T1 Delay Settings in 9914 Mode

HSTS	USTD	vstdl	stdl	T1 Delay (ns)
X	0	0	0	2000
0	X	X	0	2000
0	X	X	1	1100
1	0	0	1	1100
1	0	1	X	500
1	1	X	X	350

Using nbaf

You can use the nbaf auxiliary command in the following situation: Suppose the NAT7210 is a Talker and a byte is written to the CDOR. Also suppose that the Controller asserts ATN before the NAT7210 transfers the byte in the CDOR. The Controller unasserts ATN and the NAT7210 is still a Talker. If NTNL is set, the Talker transmits the byte stored in the CDOR. The nbaf command suppresses the transmission of this byte.

Receiving GPIB Data Messages

Basic Flow

Complete the following steps to receive GPIB (device-dependent) data from another device in your system:

1. Wait until the NAT7210 has been programmed or addressed to be an Active Listener.

2. Issue the rhdf auxiliary command to release holdoffs caused by the hldi auxiliary command.

3. Perform the following steps for each data byte that is received:

   - Wait for the DI bit to set.

   - Read the DIR to obtain the received byte.

   The host interface can either poll the DI bit or configure the NAT7210 to interrupt on the DI condition by setting DI IE.

Receiving END or EOS

The END RX bit (ISR1[4]) informs the control program that the NAT7210 has received an END or EOS message. The EOS message can be the new line character, 0A hex (if the NLEN bit is set in IMR0), the value written to the EOSR (if the REOS bit in AUXRA is set), or both.

When the END bit is set, read the EOI bit in ADR1 and the NL and EOS bits in ISR0 to determine which terminating event caused END to set.

Performing an RFD Holdoff on the Last Data Byte

You usually want to perform an RFD holdoff on the last byte of a string of data that was read in from the GPIB during a GPIB read operation. If the last byte of the data read in was sent with EOI asserted or was the EOS character, an RFD holdoff is performed automatically if the NAT7210 has been programmed for the RFD Holdoff On END Mode (AUXRA(1–0) = 10 (bin)).

If you do not know whether the last byte that the Talker sent will be transmitted with EOI or whether it will match the EOS character, program the NAT7210 to RFD Holdoff On All Data Mode (AUXRA(1–0) = 01 (bin)). By programming the NAT7210 to RFD Holdoff On All Data Mode, you force the NAT7210 to perform an RFD holdoff on the next byte of data that is received in the DIR until a Release Handshake Holdoff auxiliary command is issued.

Also see the *Acceptor Handshake Holdoffs* section, which is located later in this chapter.

Using DMA

To use DMA to receive data bytes, set the DMAI bit after the NAT7210 has been addressed to listen. The DRQ pin of the NAT7210 asserts whenever the DIR contains a byte to read. The DMA Controller should respond to DRQ by asserting the DACK* and RD* pins in order to read a data byte from the DIR.

Acceptor Handshake Holdoffs

The GPIB rdy Message and RFD Holdoffs

When it is a Listener, the NAT7210 must let the Talker know whether the NAT7210 is ready to receive another data byte. The NAT7210 unasserts the GPIB Not Ready For Data (NRFD) signal to indicate that it is ready to receive another byte. The NAT7210 generates the Ready For Next (rdy) message internally. When rdy = 1, the NAT7210 is ready to receive a data byte. When rdy = 0, the NAT7210 is not ready to receive a data byte and it asserts the GPIB NRFD signal. When the NAT7210 asserts the GPIB NRFD signal to prevent the transmission of a data byte, the NAT7210 is performing a *Ready For Data (RFD) holdoff*. (For more information, refer to Figure B-5, *Three-Wire Handshake Process*, in Appendix B, *Introduction to the GPIB*.)

The NAT7210 performs RFD holdoffs only on data bytes—that is, bytes sent with ATN unasserted. The NAT7210 can holdoff command bytes by using DAC holdoffs.

Generating the rdy Message

The local rdy message becomes true if ATN is asserted or if the following conditions are both true:

- The NAT7210 is not performing an immediate RFD holdoff.
- The NAT7210 is not performing a data byte RFD holdoff.

Immediate RFD Holdoff

Write the Holdoff Handshake Immediately (hldi) auxiliary command to the AUXMR in order to start an immediate RFD holdoff. You can clear the immediate RFD holdoff by writing the Release RFD Holdoff (rhdf) or Chip Reset auxiliary command to the AUXMR. The pon message does not clear an immediate RFD holdoff condition, so the host interface can issue hldi while pon is set and the NAT7210 is being configured.

Read the Acceptor Not Ready Holdoff Immediately (ANHS2) bit (SASR[4]) to determine the state of the Immediate Holdoff function.

Data-Receiving Modes

The data byte RFD holdoff condition is set and cleared depending on the data-receiving mode. When the NAT7210 is a Listener, it receives data in one of four modes. One (and only one) data-receiving mode is active at any time.

1. Normal

In normal mode, the NAT7210 performs a data byte RFD holdoff after every byte it accepts. After receiving a data byte, the NAT7210 is not ready to receive another byte until the data byte RFD holdoff condition is cleared. The NAT7210 releases the holdoff when the DIR is read.

2. RFD Holdoff On All Data (hlda)

In hlda mode, the NAT7210 performs a data byte RFD holdoff after every data byte it receives. Writing the rhdf auxiliary command to the AUXMR clears the RFD holdoff condition in hlda mode.

3. RFD Holdoff On END (hlde)

In hlde mode, the NAT7210 performs a data byte RFD holdoff after every data byte it receives. If the last data byte the NAT7210 received satisfies the END condition, writing the rhdf auxiliary command to the AUXMR clears the RFD holdoff. The END condition is defined by

END = EOI + (REOS & EOS) + (NLEE & newline)

If the last data byte the NAT7210 received does not satisfy the END condition, the NAT7210 releases the holdoff when the DIR is read.

4. Continuous (cont)

You generally use continuous mode when the NAT7210, as a Controller, wants to monitor a data transfer between two devices but does not need to store the data.

The NAT7210 performs an RFD holdoff in continuous mode if it receives a byte that satisfies the END condition. If the NAT7210 receives a byte that does not satisfy the END condition, it does not perform an RFD holdoff. Writing the rhdf auxiliary command to the AUXMR clears the RFD holdoff in continuous mode. Bytes are not stored in the DIR and the DI bit does not set in continuous mode.

Choosing a Data-Receiving Mode

The value of the AUXRA[1–0] bits usually determines the data-receiving mode. However, if the host interface writes the (ltn&cont) command to the AUXMR, the NAT7210 uses continuous mode until the host interface either writes the ltn command to the AUXMR or the NAT7210 becomes unaddressed as a Listener.

DAC Holdoffs

When a DAC holdoff condition is true, the NAT7210 is interpreting but has not yet accepted a command byte sent by the Controller. A DAC holdoff forces the Controller to keep the command byte on the GPIB valid and the GPIB Data Valid (DAV) signal asserted. (For more information, refer to Figure B-5, *Three-Wire Handshake Process*, in Appendix B, *Introduction to the GPIB*.) By using DAC holdoffs, a control program prevents another command from being sent until the current command has been completely processed. Once the host interface responds to the command byte, it releases the DAC holdoff by writing the valid or nonvalid auxiliary command to the AUXMR.

In most applications, you do not need to use DAC holdoffs: the NAT7210 automatically interprets command bytes and sets various interrupt bits when it receives certain command bytes.

DAC holdoffs can occur only on GPIB command bytes (ATN asserted). Data bytes (ATN unasserted) can be held off with RFD holdoffs, which are described earlier in this chapter.

Determining When DAC Holdoffs Occur

The NAT7210 can be configured to perform DAC holdoffs on many different types of command bytes. The SDHS signal determines which command bytes will cause a DAC holdoff. SDHS is defined by the following:

```
SDHS  = [UCG + ACG & (TADS + LADS)] & undefined & CPT ENAB
      + UDPCF & SCG & CPT ENAB
      + DHADT & GET
      + DHADC & (SDC + DCL)
      + DHATA & TAG & ~UNT
      + DHALA & LAG & ~UNL
      + DHUNTL & (UNT + UNL)
      + DHALL & (UCG + ACG + SCG)
      + DHDC & (DCL + SDC & LADS)
      + DHDT & GET
      + SCG & (TPAS + LPAS) & (dual extended address mode)
```

By issuing the valid or nonvalid auxiliary command, you clear the Acceptor Data Holdoff State (ADHS). By clearing ADHS, you clear the DAC holdoff.

Read the ADHS bit (SASR[3]) to determine the state of the DAC holdoff condition.

Device Status Reporting (Polling)

Requesting Service

Asserting the SRQ Signal

The NAT7210 requests service from the GPIB CIC by asserting the GPIB SRQ signal. However, the host interface cannot directly control the SRQ signal: the rsv signal determines when the NAT7210 asserts SRQ.

After rsv asserts, the NAT7210 asserts the SRQ signal. When the CIC serial polls the NAT7210, the NAT7210 unasserts SRQ. The NAT7210 does not assert SRQ again until rsv unasserts, then reasserts. Refer to the SR1 function in the IEEE 488.1 standard for more information.

IEEE 488.2 Service Requesting

To request service, issue the reqt auxiliary command, then write the status byte (STB) to the SPMR.

Note: ***If STBO IE = 1 after issuing reqt, do not write to the SPMR until the STBO interrupt condition becomes true.***

When you write to the SPMR, write 0 to bit 6 (the rsv bit). The NAT7210 asserts and unasserts the rsv signal according to the Set rsv State Machine that is described in the IEEE 488.2 standard.

After the CIC serial polls the NAT7210, you must issue the reqt auxiliary command and write to the SPMR again to request service. If you want to stop requesting service before a serial poll occurs, issue the reqf auxiliary command.

7210-Style Service Requesting

Most applications should use the IEEE 488.2 service-requesting method that is described in the section above. However, the NAT7210 also supports the 7210 style of requesting service. To request service, check the PEND bit of the SPSR to make sure that the NAT7210 is not currently responding to a serial poll. If PEND = 0, write the desired STB to the SPMR. When you write to the SPMR, set bit 6, which is the rsv bit. This write causes the PEND bit to set and the NAT7210 to assert the GPIB SRQ line. The PEND bit remains set until the serial poll completes.

Responding to Serial Polls

If STBO IE = 0 when the CIC serial polls the NAT7210, the NAT7210 sends the STB to the CIC without the host interface intervening.

If the contents of the STB are likely to change between the time you issue reqt and the time the CIC serial polls the NAT7210, you can use the STBO IE bit. When STBO IE = 1, the NAT7210 does not respond to a serial poll immediately. Instead, when the CIC serial polls the NAT7210, the NAT7210 generates an interrupt. In response to this interrupt, the host interface writes the STB to the SPMR (write 0 to bit 6). The NAT7210 responds to the serial poll by sending the STB to the CIC.

When the NAT7210 sends the STB to the CIC, it also sends the RQS message to the CIC by asserting the GPIB DIO7 signal if the NAT7210 is requesting service. The CIC normally reads the STB once, but if the CIC asserts ATN between each 1-byte read, the CIC can read the STB any number of times. The NAT7210 asserts the GPIB DIO7 signal, however, only during the first read. After the first read, rsv clears. PEND clears when the CIC asserts ATN in order to terminate the serial poll.

The NAT7210 asserts the GPIB EOI line during a serial poll if the SPEOI bit of AUXRB is set.

Responding to Parallel Polls

For more information, see the *Parallel Polling* section in Appendix B, *Introduction to the GPIB.*

The ist Message

When it responds to a Parallel Poll, the NAT7210 can transmit only one bit of information to the CIC. This one bit contains the status of the ist message. If ISS = 1 (AUXRB[4]), ist is true if the NAT7210 is asserting the SRQ signal—that is, the IEEE 488.1 Service Request function of the NAT7210 is in the SRQS state.

If ISS = 0, you set and clear the ist message by using the ist and ~ist auxiliary commands. If ISS = 0, the meaning of the ist message is device dependent.

Remote Configuration

Before the CIC can parallel poll the NAT7210, the NAT7210 must first be configured to respond to parallel polls. The control program can locally configure the NAT7210 (IEEE 488.1 capability code PP1) or the NAT7210 can let the CIC remotely configure the NAT7210 (IEEE 488.1 capability code PP2).

To let the CIC remotely configure the NAT7210, clear the PP2 bit in AUXRI. Do not write to the PPR. The CIC configures the NAT7210 without software intervention, and it enables or disables the NAT7210 to respond to parallel polls. The CIC configures the polarity of the NAT7210's response, and it also selects the PPR message that the NAT7210 uses to respond to parallel polls.

Local Configuration

To implement local configuration, disable remote configuration by setting the PP2 in AUXRI, then write to the PPR to configure the parallel poll response. The bits in the PPR determine which GPIB data line the NAT7210 uses to respond to parallel polls. The PPR also determines the polarity of the parallel poll response. See Table 5-4.

Table 5-4. Parallel Poll Register Example

Binary Value Written to the AUXMR	Result
0 1 1 1 0 0 0 0	Unconfigures PPR. U = 1.
0 1 1 0 0 0 0 0	0 0 0 0 0 is written to the PPR. The NAT7210 participates in parallel polls, asserting the DIO1 line if ist is 0.
0 1 1 0 1 0 0 1	0 1 0 0 1 is written to the PPR. The NAT7210 participates in parallel polls, asserting the DIO2 line if ist is 1.

Disabling the Parallel Poll Response

To completely disable the NAT7210 from responding to parallel polls (IEEE 488.1 capability code PP0), set the PP2 bit in AUXRI and set the U bit in the PPR.

Generating Hardware Interrupts

The Interrupt pin asserts when the INT bit (ISR2[7]) is true. If INV = 1 (AUXRB[3]), the INTR pin is active low. If INV = 0, the INTR pin is active high.

Remote/Local State Considerations

The NAT7210 implements the GPIB Remote/Local (RL1) function as described by the IEEE 488.1 standard. The host interface determines the state of the RL1 function by reading the Lockout (LOK) bit and the Remote (REM) bit in ISR2. The Lockout Change (LOKC) bit and the Remote Change (REMC) bit can be used to interrupt the host interface when the state of the RL1 function changes.

If the NAT7210 is not in a Lockout state (that is, LOK = 0), the host interface can force the NAT7210 to enter a Local state (REM = 0) by writing one of the Return To Local (rtl) auxiliary commands to the AUXMR.

See the IEEE 488.1 standard and the IEEE 488.2 standard for device requirements that depend on the RL1 function.

Device Triggering

The NAT7210 enters the Device Trigger Active State (DTAS) when the GPIB Controller sends the Group Execute Trigger (GET) command to the NAT7210. As the IEEE 488.1 standard requires, the NAT7210 enters DTAS only if the NAT7210 is an Addressed Listener. The DET bit in ISR1 sets when the NAT7210 enters DTAS. If the Data Accepted Holdoff On Device Trigger Active State (DHDT) bit (AUXRE[1]) is set, the NAT7210 performs a DAC holdoff when the DET bit sets.

If the DAC Holdoff On Get (DHADT) bit (AUXRE[3]) is set, the NAT7210 performs a DAC holdoff when the NAT7210 receives the GET command (whether or not the NAT7210 is a GPIB Listener).

The DET bit can cause an interrupt if the Enable Interrupt On Device Execute Trigger (DET IE) bit in IMR1 is set. DHADT and DHDT can cause an interrupt if the Enable Interrupt On Command Pass Through (CPT IE) bit (IMR1) is set.

Device Clearing

As the IEEE 488.1 standard requires, the NAT7210 enters the Device Clear Active State (DCAS) when the GPIB Controller sends the Device Clear (DCL) command or when the NAT7210 is a GPIB Listener and the Controller sent the Selected Device Clear (SDC) command. The Device Clear (DEC) bit in ISR1 detects when the NAT7210 enters DCAS. If the Data Accepted Holdoff On Device Clear Active State (DHDC) bit (AUXRE[0]) is set, the NAT7210 performs a DAC holdoff when the DEC bit sets.

If the DAC Holdoff On DCL Or SDC (DHADC) bit (AUXRE[2]) is set, the NAT7210 performs a DAC holdoff when the NAT7210 receives the DCL or SDC command (whether or not the NAT7210 is a GPIB Listener).

DEC can cause an interrupt if the Device Clear Interrupt Enable (DEC IE) bit in IMR1 is set. DHADC and DHDC can cause an interrupt if the CPT IE bit (IMR1) is set.

Chapter 6
Controller Software Considerations

This chapter explains important system and GPIB Controller considerations. The information in this chapter applies only to applications in which the NAT7210 has GPIB Controller capabilities. Because a GPIB Controller has Talker, Listener, and Controller capabilities, you should also refer to Chapter 5, *Software Considerations*.

This chapter, except where explicitly noted, assumes that the NAT7210 uses 7210-mode architecture and that the reader is familiar with the GPIB. For more information about GPIB, refer to Appendix B, *Introduction to the GPIB*, and the IEEE 488.1 and IEEE 488.2 standard.

System Controller Considerations

In a GPIB system, only one device may be the System Controller. The System Controller drives the GPIB REN and IFC signals true or false. The System Controller becomes the Active Controller by asserting IFC.

Becoming System Controller

Complete the following steps to make a device become the System Controller:

1. Enable the external GPIB transceivers to drive the GPIB IFC and REN signals. In a typical system, a 75162-type transceiver drives these signals. In such a system, you assert the SC pin of the 75162 to enable the GPIB transceiver to drive the GPIB IFC and REN signals.

   The NAT7210 does not directly drive the 75162 SC pin. The SC pin should be driven with circuitry that is external to the NAT7210. See the *Interfacing to Common GPIB Transceivers* section in Chapter 7, *Hardware Considerations*.

2. Write the sic auxiliary command to the AUXMR to assert the GPIB IFC signal. The IFC message initializes the GPIB interface functions of all devices on the bus. The sic command also causes the NAT7210 to become CIC and to assert the GPIB ATN signal.

3. Wait at least 100 µs. The IEEE 488.1 standard requires that IFC be asserted for at least 100 µs.

4. Write the ~sic auxiliary command to the AUXMR in order to unassert the GPIB IFC signal.

Other System Controller Capabilities: Setting and Clearing REN

The System Controller can assert and unassert the GPIB REN signal. The control program issues the sre and ~sre auxiliary commands to set and clear REN. The control program must guarantee that the NAT7210 never unasserts the REN signal for less than 100 µs.

Disabling System Controller Capabilities

Complete the following steps to disable the System Controller capabilities of the NAT7210:

1. Write the ~rsc auxiliary command to the AUXMR.

2. Disable the external GPIB transceivers from driving the GPIB IFC and REN signals. In a system that uses a 75162-type transceiver, you unassert the SC pin of the 75162 to enable the GPIB transceiver to receive the GPIB IFC and REN signals.

GPIB Controller Considerations

Three Basic Controller States

The IEEE 488.1 Controller function has many states, but for clarity, this manual groups them into three basic Controller function states: Idle Controller State, Active Controller State, and Standby Controller State. See Figure 6-1.

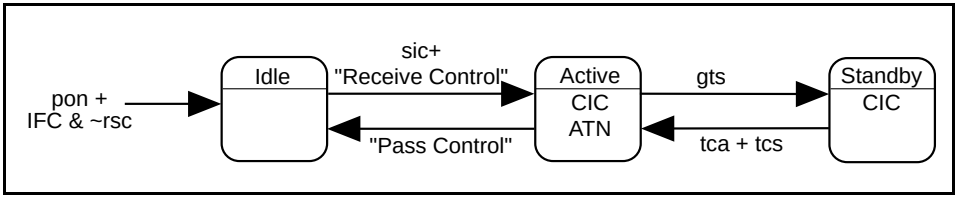


Figure 6-1. Basic Controller States

Idle Controller State

In the Idle Controller State, the NAT7210 is not CIC and does not drive the ATN signal. The NAT7210 can, however, send and receive data bytes in the Idle Controller State if the device that is CIC addresses the NAT7210 to talk or listen.

Only one device may be CIC at a time. Therefore, every GPIB device except the CIC must be in the Idle Controller State. In Talk-Only and Listen-Only applications, no device is CIC.

The NAT7210 enters the Idle Controller State if the pon message is true or if another device asserts the GPIB IFC message.

Active Controller State

In the Active Controller State, the NAT7210 is CIC and drives the ATN signal asserted.

As an Active Controller, the NAT7210 can send remote multiline messages (commands such as Untalk) and conduct serial and parallel polls. No devices can send or receive data bytes when the NAT7210 is the Active Controller.

Standby Controller State

In the Standby Controller State, the NAT7210 is CIC and drives the ATN signal unasserted. As a Standby Controller, the NAT7210 can send and receive data bytes if it addresses itself to talk or listen.

Determining the Basic Controller State

Read the ADSR and examine the CIC and ATN* bits (ADSR[7] and ADSR[6]). Read Table 6-1 to determine the Basic Controller State.

Table 6-1. Basic Controller State

CIC (ADSR[7])	ATN* (ADSR[6])	Basic Controller State
0	X	Idle Controller
1	0	Active Controller
1	1	Standby Controller

Changing Controller States

Idle State to Active State: Becoming CIC

When the NAT7210 becomes System Controller (by issuing the sic command), the NAT7210 exits the Idle Controller State and enters the Active Controller State. The NAT7210 also enters the Active Controller State if another CIC passes control to it.

The NAT7210 receives control from another CIC when the following events occur:

1. The current CIC sends the GPIB talk address (MTA) of the NAT7210. The NAT7210 enters the GPIB Talker Addressed State (TADS) and sets the TA bit in the ADSR.

2. The current CIC sends the GPIB Take Control (TCT) message to the NAT7210.

3. The current CIC sees that the GPIB handshake completes. The device that sent the TCT message becomes an Idle Controller and stops driving ATN.

4. When ATN unasserts, the NAT7210 becomes the Active Controller and asserts ATN. The NAT7210 sets the CIC bit in the ADSR and the CO bit in the ISR2.

Active State to Standby State

Complete the following steps to make the NAT7210 exit the Active Controller State and enter the Standby Controller State:

1. Complete any necessary addressing. In most cases, you address the NAT7210 to be a Talker or a Listener before the NAT7210 becomes the Standby Controller.

2. Wait until the CO bit in ISR2 is set in order to guarantee that the NAT7210 is the Active Controller and that no parallel poll or command transfer is in progress.

3. Write the Go To Standby (gts) auxiliary command to the AUXMR.

Standby State to Active State

Complete the following steps to make the NAT7210 exit the Standby Controller State and enter the Active Controller State:

1. Wait for the chip to finish transferring bytes as a Talker or a Listener. Use the SYNC bit to determine when the GPIB is synchronized.

2. Write the Take Control Asynchronously (tca) or Take Control Synchronously (tcs) auxiliary command to the AUXMR.

3. Wait for the CO bit to set. The CO bit sets when the NAT7210 becomes the Active Controller.

Active State to Idle State: Passing Control

The NAT7210 exits the Active Controller State and enters the Idle Controller State by passing control to another device. Complete the following steps to pass control to another device:

1. Send the device's talk address (its MTA) across the GPIB.
2. Send the TCT command.

The NAT7210 enters the Idle Controller State and unasserts ATN when the device accepts the TCT command.

Sending Remote Multiline Messages (Commands)

Complete the following steps to send commands (GPIB interface messages) to devices in your system:

1. Be sure that NTNL = 1 (AUXRG[3]). NTNL prevents the NAT7210 from sourcing commands if there are no listening devices in the system.
2. Make the NAT7210 become the Active Controller.
3. Wait for the CO bit to set.
4. Perform the following actions for each command that is sent:
   - Write the command byte to the CDOR.
   - Wait for either the CO bit (ISR2[3]) or the ERR (ISR1[2]) bit to set. CO indicates that all devices in the system have accepted the command. ERR indicates that no properly operating GPIB devices are in the system.

   The host interface can either poll the CO and ERR bits or configure the NAT7210 to interrupt on these conditions by setting CO IE and ERR IE.

   If the ERR bit sets, clear any byte that might be in the CDOR by issuing the nbaf auxiliary command, then clear the ERR bit by issuing the ClearERR auxiliary command.

   Normally, command strings are short and DMA is not used.

Polling: Obtaining Status from Devices

According to the IEEE 488.1 standard, the CIC obtains status from devices by using two methods: Parallel Polls and Serial Polls. A CIC can serial poll only one device at a time. The device responds by sending back a status byte to the CIC. All system devices can respond to a parallel poll simultaneously. However, each device can send only one bit of status to the CIC.

Any device can assert the GPIB SRQ signal to indicate that its status has changed. The CIC can poll devices at any time; however, the CIC does not generally poll devices unless SRQ is asserted.

Conducting Serial Polls

The NAT7210, as a CIC, can serial poll other devices as described in the IEEE 488.1 standard. Complete the following steps to conduct a serial poll:

1. Make the NAT7210 become the Active Controller.
2. Send the UNT and UNL commands to unaddress all devices.
3. Make the device that will be serial polled an Addressed Talker. To make the device a Talker, send the MTA command and—if the device uses secondary addressing—the MSA command of the device.
4. Configure the NAT7210 to be a Listener.
5. Send the SPE command.
6. Make the NAT7210 become a Standby Controller. (See the *Changing Controller States* section, which is located earlier in this chapter.)

The device then sources its status byte. The NAT7210 should read this byte from the DIR like it would read any other data transfer from the device. After it has read the status byte, make the NAT7210 become the Active Controller again, then send the SPD command.

Configuring Devices for Parallel Polls

If the NAT7210 is the Controller, complete the following steps to use the NAT7210 to remotely configure other devices to respond to parallel polls:

1. Place the NAT7210 in the Active Controller State.
2. Send the GPIB UNL (Unlisten) message to unaddress all GPIB Listeners.
3. Address a device to Listen by sending the MLA command and—if the device uses secondary addressing—the MSA command of the device.
4. Send the GPIB Parallel Poll Configure command (PPC).
5. Send the PPE message for that device. (See the *Determining the PPE Message* section in Appendix B, *Introduction to the GPIB*.)
6. Repeat steps 2 through 5 for each device that you will configure.

To disable a device from responding to parallel polls, repeat steps 1 through 3, then send the GPIB Parallel Poll Disable (PPD) command.

Conducting Parallel Polls

The NAT7210, as a CIC, can parallel poll other devices by using the rpp auxiliary command or the RPP2 bit.

The rpp Auxiliary Command

In most applications, this method is easier to implement than the RPP2 method. Complete the following steps to conduct a parallel poll that uses the rpp auxiliary command:

1. Make the NAT7210 become the Active Controller.
2. Issue the rpp auxiliary command.
3. Wait until the CO bit (ISR2[3]) sets.
4. Read the CPTR to obtain the parallel poll response.

The NAT7210 stops conducting the parallel poll 2 µs (or longer) after you issue the rpp command. The CO bit sets when the NAT7210 stops conducting the parallel poll after you issue the rpp command.

The RPP2 Bit

By using the RPP2 bit, you can control the duration of a parallel poll in software. Complete the following steps to conduct a parallel poll that uses the RPP2 bit:

1. Make the NAT7210 become the Active Controller.
2. Set the RPP2 bit.
3. Wait for at least 2 µs.
4. Read the CPTR to obtain the parallel poll response.
5. Clear the RPP2 bit.

Note: ***For more information on parallel polls, see the*** *Parallel Polling* ***section in Appendix B,*** *Introduction to the GPIB.*

Chapter 7
Hardware Considerations

This chapter explains important NAT7210 hardware-interfacing considerations, including a description of the pins.

Pin Descriptions

GPIB Transceiver Controls

T/R1

T/R1 asserts when the NAT7210 drives the GPIB DIO signals. The NAT7210 drives the DIO signals when it is responding to a parallel poll or when it is sourcing data bytes, command bytes, or status bytes. T/R1 can be connected directly to the TE inputs of 75160 and 75162 GPIB transceivers.

T/R1 = (PPAS & ~CIC) + ~SIDS

T/R2 and T/R3

Four internal NAT7210 signals—EOIOE, TRIG, CIC, and PE—can drive the T/R2 and T/R3 signals. (EOIOE, TRIG, CIC, and PE are described in the following sections.) The TRM[1–0] bits in the ADMR determine which internal signals drive T/R2 and T/R3. In applications that use 75160 and 75162 transceivers, the TRM[1–0] bits should both be set so that the CIC signal drives T/R2 and the PE signal drives T/R3.

CIC Internal Signal

CIC asserts when the NAT7210 is an Active or Standby Controller. CIC unasserts when the NAT7210 is an Idle Controller. The NAT7210 is Controller-in-Change when it is the Active or Standby Controller.

When the NAT7210 is the Active Controller, it asserts the IEEE 488 ATN* signal. As an Active Controller, the NAT7210 can send remote multiline messages (commands) and conduct serial and parallel polls.

When the NAT7210 is the Standby Controller, it does not assert the IEEE 488 ATN* signal. When the NAT7210 is the Standby Controller, the IEEE 488 Addressed Talker can send data to the IEEE 488 Addressed Listener.

The CIC signal can be read as bit 7 of the Address Status Register (ADSR[7]) in 7210 mode.

Referring to the IEEE 488.1 Controller function:
CIC = ~(CIDS + CADS)

PE Internal Signal

PE asserts when the NAT7210 is not responding to a parallel poll. When the NAT7210 is responding to parallel polls, the GPIB transceivers should drive the GPIB DIO signals with open-collector drivers.

PE = ~(PPAS & ~CIC)

EOIOE Internal Signal

EOIOE asserts when the NAT7210 drives the GPIB EOI signal. The NAT7210 drives EOI when it is sourcing data bytes, command bytes, or status bytes. The NAT7210 also drives EOI when it conducts a parallel poll.

EOIOE = TACS + SPAS + (CIC & ~CSBS)

TRIG Internal Signal

TRIG asserts when the NAT7210 is in the IEEE 488 Device Trigger Active State (DTAS). The NAT7210 enters the DTAS when it is an Addressed Listener and is receiving the Group Execute Trigger (GET) command from the Active Controller.

The TRIG internal signal also pulses when the trig auxiliary command is written to the Auxiliary Mode Register (AUXMR) in 7210 mode or when the fget auxiliary command is written to the Auxiliary Command Register (AUXCR) in 9914 mode.

GPIB Signal Pins

You can directly connect the NAT7210 GPIB signal pins (IFC, REN, SRQ, ATN, EOI, DAV, NDAC, and NRFD) to a GPIB transceiver. One popular transceiver is the 75162 IC. If you use a 75162, you must connect the GPIB *bus* pins that are on the 75162 (usually pins 3 through 10 in a DIP package) to the GPIB connector and connect the *terminal* pins (pins 13 through 20 in a DIP package) to the NAT7210. (See Figure 7-2.)

GPIB Data Bus Pins

You can directly connect the NAT7210 GPIB data bus pins (DIO[8–1]) to a GPIB transceiver. One popular transceiver is the 75160 IC. If you use a 75160, you must connect the GPIB *bus* pins that are on the 75160 (usually pins 2 through 9 in a DIP package) to the GPIB connector and connect the *terminal* pins (pins 12 through 19 in a DIP package) to the NAT7210.

CPU Register Control Pins

CS* and the CPU Address Bus

The address bus, RS(2–0), selects one NAT7210 internal register for reading and writing. The Page-in condition also affects which register RS(2–0) selects. See *The Page-In State* section in Chapter 3, *7210-Mode Interface Registers*.

The NAT7210 ignores RS(2–0) if the host interface asserts the DACK* pin or if the host interface does not assert CS*.

RD*/WR*

Asserting the RD* pin enables the CPU or DMA Controller to read a NAT7210 register. Some read accesses also have other effects on the NAT7210. For example, if the SISB = 0, reading ISR1 clears the bits in ISR1.

Asserting the WR* pin enables the CPU or DMA Controller to write to a NAT7210 register. Some writable registers are not storage registers. For example, writing certain values to the AUXMR forces internal signals to pulse true.

CPU Data Bus

The bidirectional CPU Data bus D(7–0) transfers data, commands, and status between the CPU and the NAT7210. D7 is the most significant bit of the NAT7210 8-bit CPU data bus. D0 is the least significant bit.

DMA Pins

DRQ

When the DMAI bit is set, DRQ asserts when the NAT7210, as a Listener, accepts a byte from the GPIB. DRQ does not assert if the NAT7210 is in continuous mode (see the *Acceptor Handshake Holdoffs* section in Chapter 5, *Software Considerations*.) DRQ unasserts when the host interface reads the GPIB data byte from the DIR.

When the DMAO bit is set, DRQ asserts when the NAT7210 is a Talker and the CDOR is empty. DRQ unasserts when the host interface writes a byte to the CDOR.

DRQ unasserts when the host interface asserts the DACK* pin. However, if the host interface does not read DIR (if DMAI = 1) or write to the CDOR (if DMAO=1), DRQ reasserts when DACK* unasserts.

DRQ is an output-only pin. If the application does not require DMA, you can leave DRQ unconnected.

DACK*

Asserting DACK* enables accesses to the CDOR (for DMA writes) and the DIR (for DMA reads). Asserting DACK* also causes the DRQ pin to unassert. The NAT7210 ignores the CS* and RS(2–0) pins when DACK* is asserted.

The DACK* pin is an active low input-only pin with an internal pull-up resistor. If the application does not require DMA, you can connect DACK* to Vcc or leave DACK* unconnected.

Other Pins

INT

The Interrupt pin asserts when the INT bit in ISR2 is true. If INV = 1 (AUXRB[3]), the INTR pin is active low. If INV = 0, the INTR pin is active high.

Several of the interrupting conditions depend on the CLK signal. For these interrupts, the delay between an event occurring and the related interrupt asserting depends on the CLK frequency.

RESET

Asserting RESET places the NAT7210 in 7210 mode. (See Figure 2-2, *Changing the NAT7210 Mode*, in Chapter 2, *NAT7210 Architecture*.) Asserting RESET initializes all GPIB interface functions. The NAT7210 asserts the local pon message until the control software issues the immediate execute pon auxiliary command. See the *Auxiliary Mode Register (AUXMR)* section in Chapter 3, *7210-Mode Interface Registers.*

The RESET signal is asynchronous to the CLK signal.

CLK

The following list summarizes the CLK signal requirements for the NAT7210:

- The CLK pin is a TTL input.
- The minimum CLK frequency is 1 MHz.
- The NAT7210 does not function properly without a CLK signal.
- The rising and falling edges of the CLK signal must be monotonic.
- The minimum pulse width of the CLK signal high or low (T_{pw} in Figure 7-1) is 15 ns.
- National Instruments does not specify the maximum rise or fall time of the CLK signal (T_{rf} in Figure 7-1). Using a signal with unusually slow rise and fall times can adversely affect power consumption.

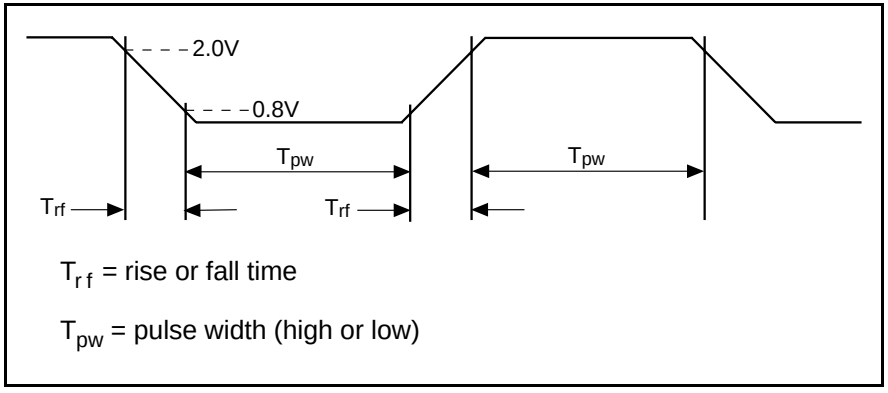


Figure 7-1. CLK Signal Timing Diagram

Interfacing to Common GPIB Transceivers

Figure 7-2 shows how to interface the NAT7210 to the 75160 and 75162 transceivers. The SC signal can be driven by a register external to the NAT7210.

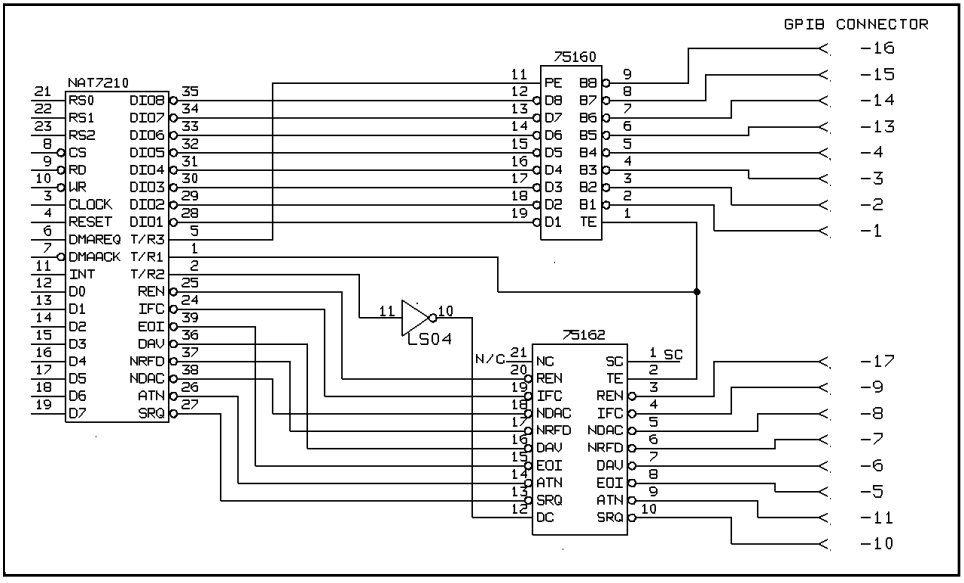


Figure 7-2. Interfacing the NAT7210 to the 75160 and 75162 Transceivers

Appendix A
Common Questions

This appendix lists common questions and answers.

If there are two devices, one using a NAT7210 with a 16 MHz clock frequency, and another using the NAT7210 with a 5 MHz clock frequency, will the one with the higher clock frequency communicate faster?

Yes, but not much faster. Depending on the application, the 16 MHz chip might increase throughput by up to 15%. It will not make your throughput 3 times faster.

When the NAT7210 sends a data byte, it delays for a time "T1." The longer the T1 delay is, the longer it takes to transfer a byte.

For example, suppose you program the NAT7210 to use a T1 delay of 500 ns. For a 5 MHz clock, T1 would have to be at least 600 ns, because it must be some multiple of the clock period. On average, the actual delay would be about 700 ns, because you must add about 0.5 of a clock period for synchronizing delays. For a 16 MHz clock, 500 ns is an exact multiple of the clock period, so the average T1 delay would be close to 500 ns + 0.5 clock periods—about 531 ns.

Therefore, the T1 delay for a 16 MHz clock might be 169 ns shorter (on average) than that of a 5 MHz clock. The faster clock saves 169 ns of time on every bite. If your throughput is 10 kBytes/s, this will not be a noticeable time difference. However, if your throughput is 800 kBytes/s, the difference will be noticeable.

Although the faster clock frequencies don't improve throughput very much, the NAT7210 is designed have high throughput at any clock frequency.

What should I do if I am using a clock that uses fractional output frequencies?

Simply round up to the next highest integer. Examples follow:

Fractional Output Frequency	ICR Value
1.2	2
3.35	4
3.6864	4

You can write any integer higher than the actual input frequency to the ICR. Even if you do not write any value to the ICR, the NAT7210 will still work with actual frequencies of 1.2, 3.35, 3.6864 MHz, because it defaults to 8 MHz.

The NAT7210 uses the input clock to generate certain delays, such as the T1 delay (see previous question). For example, suppose you program the NAT7210 to use a 2 µs nominal T1 delay. The NAT7210 examines the ICR register to determine how many clock periods to wait. If the ICR contains 2, the NAT7210 assumes that the input frequency is 2 MHz and the clock period is 500 ns. For a delay of 2 µs, the NAT7210 calculates that it will need to wait:

2 µs ÷ 500 ns = 4 clock periods

If the ICR contains 2, the NAT7210 waits for 4 clock periods regardless of the actual clock frequency. If the actual frequency is 1.2 MHz (with a clock period of 833 ns), the actual delay will be:

4 clock periods x 833 ns = 3.33 µs

3.33 µs is longer than the nominal 2 µs, but this is acceptable. The T1 delay can be longer than 2 µs.

Appendix B
Introduction to the GPIB

This appendix discusses the history of the GPIB, GPIB hardware configurations, and serial polling.

History of the GPIB

Hewlett-Packard developed the original GPIB (and called it the HP-IB) in the late 1960s. Hewlett-Packard developed its HB-IB to connect and control programmable instruments that Hewlett-Packard had manufactured. The introduction of digital controllers and programmable test equipment created the need for a standard, high-speed interface that would permit communication between instruments and controllers from various vendors. In 1975, the IEEE published ANSI/IEEE Standard 488-1975, *IEEE Standard Digital Interface for Programmable Instrumentation*, which contained the electrical, mechanical, and functional specifications of an interfacing system. The original IEEE 488-1975 was revised in 1978 primarily for editorial clarification and addendum. This bus is now used worldwide and is known by three names:

- General Purpose Interface Bus (GPIB)
- Hewlett-Packard Interface Bus (HP-IB)
- IEEE 488 Bus

Because the original IEEE 488 document contained no guidelines for preferred syntax and format conventions, work continued on the specification to enhance system compatibility and configurability among test systems. This work resulted in a supplement standard—IEEE 488.2, *Codes, Formats, Protocols, and Common Commands*—that you use with IEEE 488 (which was renamed IEEE 488.1).

IEEE 488.2 does not replace IEEE 488.1. Many devices still conform only to IEEE 488.1. IEEE 488.2 builds on IEEE 488.1 by defining a minimum set of device interface capabilities, a common set of data codes and formats, a device message protocol, a generic set of commonly needed device commands, and a new status reporting model.

In 1990, a consortium of test and measurement companies developed the Standard Commands for Programmable Instrumentation (SCPI) document. SCPI defines specific commands that each instrument class (which usually includes instruments from various vendors) must obey. Thus, SCPI guarantees complete system compatibility and configurability among these instruments. You no longer need to learn a different command set for each instrument, and you can easily replace an instrument from one vendor with an instrument from another.

The IEEE 488.1 Specification

The GPIB is a digital, 8-bit, parallel communications interface with maximum data transfer rates over 1 MB/s. The bus supports one system controller—usually a computer—and up to 14 additional instruments. Because the GPIB is an 8-bit parallel interface with fast data transfer rates, it has gained popularity in other applications such as intercomputer communication and peripheral control.

IEEE 488.2 and SCPI Specifications

Although IEEE 488.1 eliminated the need to find the right type of connector and determine which signal line was connected to which pin, it did not solve other problems. More than 10 years after the release of IEEE 488.1, IEEE 488.2 and SCPI solved these problems.

Problems with IEEE 488.1 Compatible Devices

Users of IEEE 488.1 compatible devices encountered the following problems:

- No common method for performing operations existed: In a system with two different meters, one meter could require a command to take a reading while the other could take a reading without a command.
- No common data format existed among communicating devices: Two communicating devices used two different formats to represent the same number.
- No common command set existed: Two devices performed identical functions, but used completely different device-dependent data messages.
- Status reporting was unique to each device: Each device reported its status information in a different format.

The IEEE 488.2 Solution

The IEEE 488.2 standard eliminates the IEEE 488.1 problems through the following solutions:

- IEEE 488.2 contains a minimum set of required device interface capabilities.
- IEEE 488.2 specifies a way of presenting data through data formats and codes.
- IEEE 488.2 defines a specific protocol for sending device messages and the syntax for multiple commands in a single string.

- IEEE 488.2 contains a common command set.
- IEEE 488.2 contains a standard status reporting model.

SCPI Specification

The SCPI specification expands the IEEE 488.2 common command set by defining a single, comprehensive command set that is suitable for all instruments. For example, all SCPI-compatible voltmeters, regardless of manufacturer or model, respond to the same command for reading AC voltage. Their response format is also the same.

SCPI embraces many of the commands and protocols that the hardware-independent portion of the IEEE 488.2 standard defines. Figure B-1 illustrates the structure of the GPIB standards.

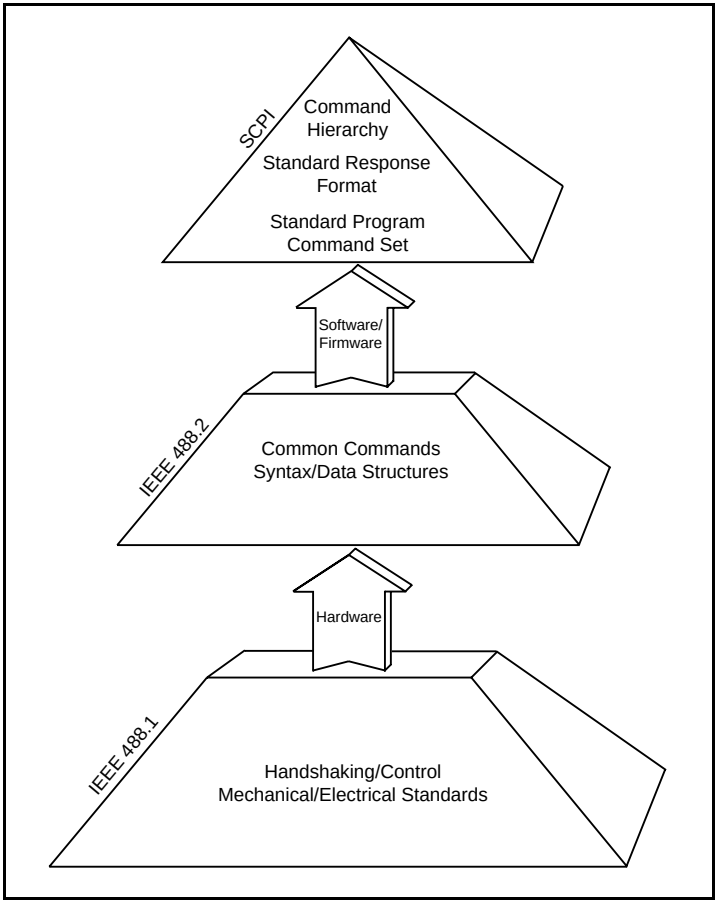


Figure B-1. Structure of the GPIB Standards

The combination of IEEE 488.2 and SCPI leads to greater productivity by featuring software command standards and instant interchangeability. Rather than learning a different command set for each instrument, you can focus on solving measurement problems.

Although you can mix SCPI and non-SCPI instruments in a system, your complete system must adhere to IEEE 488.2 for you to fully benefit from these standards.

See Appendix C, *Standard Commands for Programmable Instruments (SCPI)*, for more information.

GPIB Hardware Configuration

A GPIB hardware setup consists of two or more GPIB devices (instruments and/or interface boards) that are connected by a GPIB cable. The cable assembly consists of a shielded 24-conductor cable with a plug and a receptacle (male/female) connector at each end. With this design, you can link devices in a linear configuration, a star configuration, or a combination of these two configurations (see Figures B-2 and B-3).

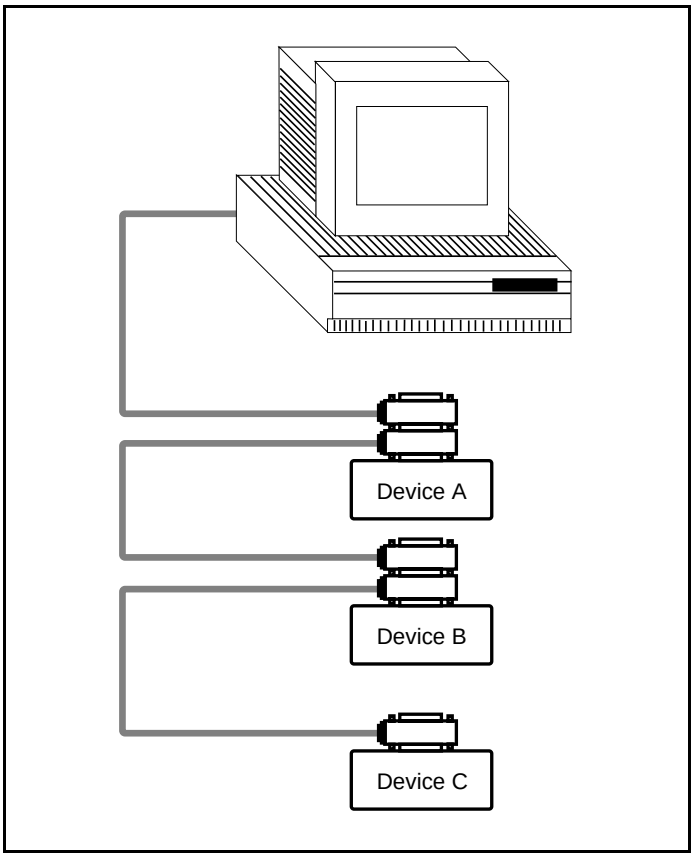


Figure B-2. Linear Configuration

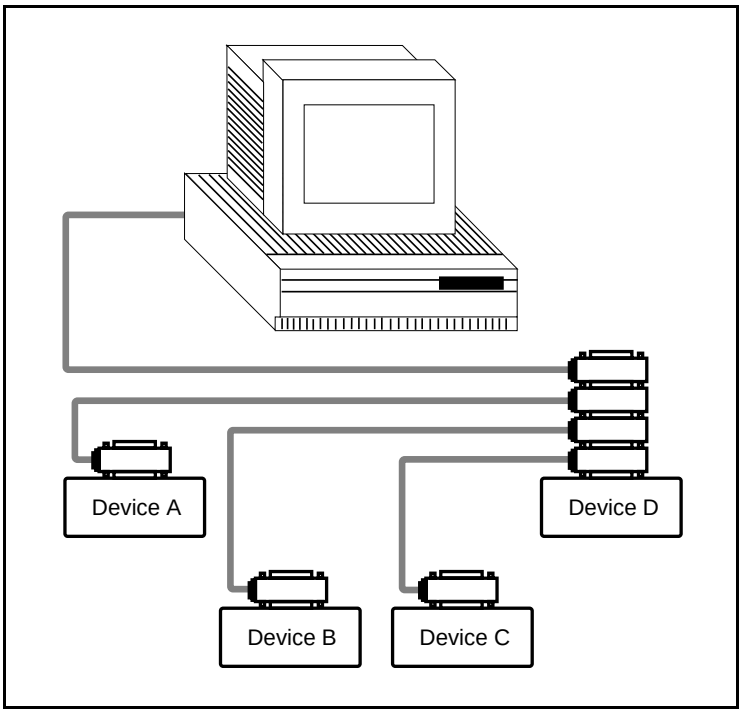


Figure B-3. Star Configuration

GPIB Signals and Lines

The GPIB has 16 signal lines and 8 ground return or shield drain lines (see Figure B-4). All GPIB devices share the same 24 bus lines. The 16 signal lines fall into three groups:

- Eight data lines.
- Five interface management lines.
- Three handshake lines.

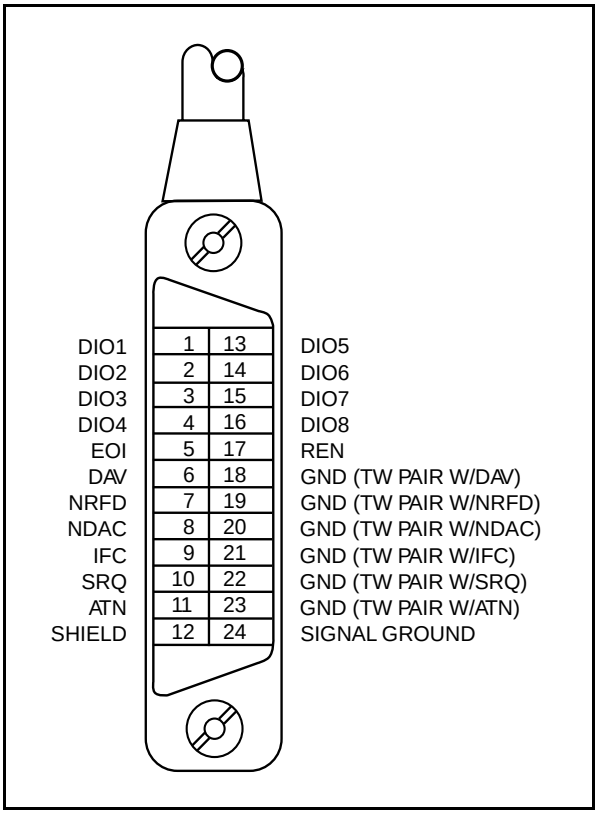


Figure B-4. GPIB Connector and Pin Assignments

Data Lines

The eight data lines, DIO1 through DIO8, carry the command and data messages on the GPIB. All commands and most data use the 7-bit ASCII or ISO code set; thus, the eighth bit, DIO8, is not used or is used for parity.

Interface Management Lines

The following lines manage the flow of information across the GPIB:

- Interface Clear (IFC)
- Attention (ATN)
- Remote Enable (REN)
- End-or-Identify (EOI)
- Service Request (SRQ)

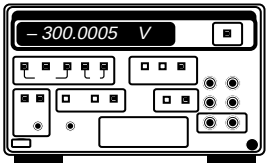
Interface Clear (IFC)

Only the System Controller can control the IFC line. The System Controller uses IFC to take control of the bus asynchronously. This action must initially be done to establish Controller status.

The IFC line is the master reset of the GPIB. When it is asserted, all devices return to a known quiescent state.

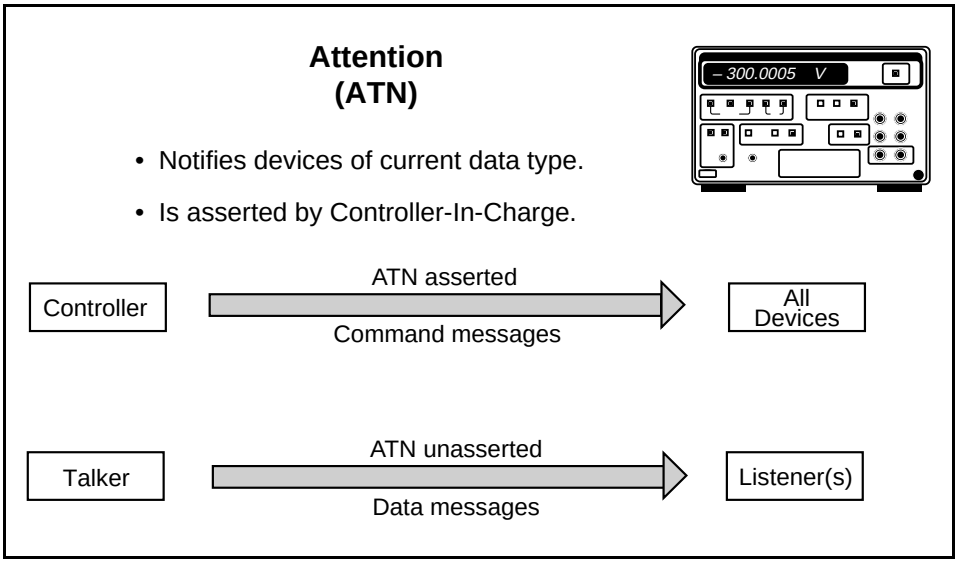
Interface Clear (IFC)

- Places all devices into quiescent state.
- Is asserted by System Controller.



Attention (ATN)

When the ATN line is asserted, all devices become Listeners and participate in the communication. ATN signifies that a GPIB command message or data message is present on the data lines. When ATN is unasserted, information on the bus is interpreted as a *data* message. When ATN is asserted, information on the bus is interpreted as a *command* message.

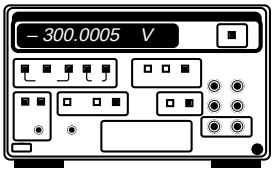


Remote Enable (REN)

The System Controller uses the REN line to put devices into a remote state. Each device has its own remote/local state capabilities. The IEEE 488 standard requires a device to go into a remote programming state whenever the REN line is asserted and addressed to listen.

Remote Enable (REN)

- Enables devices for remote programming.
- Is asserted by System Controller.

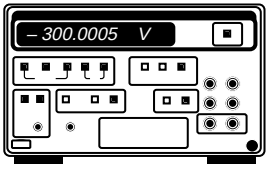


End-or-Identify (EOI)

Some devices terminate their output data by using the EOI line. A Talker asserts EOI along with the last byte of data. A Listener stops reading data when the EOI is asserted. More details of transfer termination are presented later. This line is also used in parallel polling, which will be discussed later.

End Or Identify (EOI)

- Signals end of data.
- Signals the execution of a Parallel Poll.
- Is asserted by current Talker.

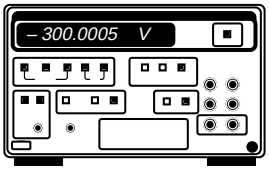


Service Request (SRQ)

A device asserts the SRQ line at any time in order to notify the CIC that it needs service. The SRQ line remains asserted until the device is serial polled. The Controller must monitor SRQ, poll the device, and determine the type of service the device needs.

Service Request (SRQ)

- Alerts Controller that service is needed.
- Is asserted by Non-Controller.



Handshake Lines

Three lines asynchronously control the transfer of message bytes among devices:

- Not Ready For Data (NRFD)
- Not Data Accepted (NDAC)
- Data Valid (DAV)

The GPIB uses a three-wire interlocking handshake scheme. This handshake scheme guarantees that message bytes on the data lines are sent and received without transmission error.

Not Ready For Data (NRFD)

The NRFD line indicates whether a device is ready to receive a data byte. When a Controller is sending commands, all devices drive NRFD. When a Talker is sending data messages, only Listeners drive NRFD.

Not Data Accepted (NDAC)

The NDAC line indicates whether a device has accepted a data byte. When a Controller is sending commands, all devices drive NRFD. When a Talker is sending data messages, only Listeners drive NRFD.

Note: ***This handshake scheme limits the transfer rate on the GPIB to that of the slowest active Listener. The transfer rate is limited because a Talker waits until all Listeners are ready (that is, NRFD is false) before sending data and waits for all Listeners to accept data (that is, NDAC is false) before transferring more data. Therefore, the slowest device dictates the maximum GPIB transfer rate.***

Data Valid (DAV)

The DAV line indicates whether signals on the data lines are stable (valid) and whether devices can safely accept the signals. When the Controller sends commands, it controls DAV, and when the Talker sends data messages, it controls DAV.

Figure B-5 illustrates the three-wire handshake process.

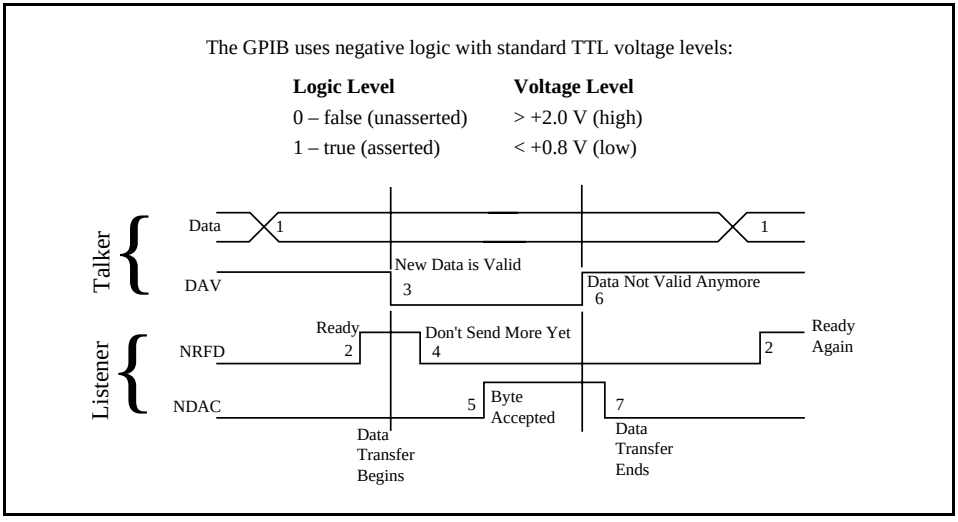


Figure B-5. Three-Wire Handshake Process

Three-Wire Handshake Process

GPIB devices use the three-wire handshake process to transfer information. The three-wire handshake process is identical for command and data transfers. During command transfers, the Controller drives the DIO and DAV lines; all devices drive the NRFD and NDAC lines. During data transfers, the Talker drives the DIO and DAV lines; all Listeners drive the NRFD and NDAC lines.

Devices drive the NDAC and NRFD lines with open-collector drivers, so if any device drives NDAC (or NRFD) to a low voltage level, the signal is logically asserted (true). If no device drives NDAC (or NRFD) to a low voltage level, the signal floats to a high voltage level; thus, the signal is logically unasserted (false).

The following actions occur during the three-wire handshake process (refer to Figure B-5):

1. The Talker (or Controller) places data on the DIO lines and waits at least T1 seconds.

2. After the T1 delay, the Talker waits until the Listener unasserts NRFD. NRFD unasserted (*not* Not-Ready-For Data) indicates that the Listener can receive the data byte.

3. The Talker asserts DAV to indicate that new data is valid on the DIO lines.

4. The Listener asserts NRFD to signal a Not Ready Status (Don't Send More Yet).

5. When the Listener accepts the current byte (by placing it in some internal buffer or by otherwise processing it), the Listener unasserts NDAC.

6. The Talker unasserts DAV.

7. The Listener asserts NDAC, then the Talker executes step 1 to begin transferring the next byte.

Physical and Electrical Specifications

To achieve the GPIB's high data transfer rate, you must limit the physical distance between devices and the number of devices on the bus. This limitation is necessary because the GPIB is a transmission line system. Any distance beyond the maximum allowable cable length, as well as any excess GPIB device loads, can surpass interface circuit drive capability.

The IEEE 488 standard dictates the following limits:

- The total length of all cables is less than or equal to 2 m times the number of connected devices—up to a total of 20 m.

- No more than 15 devices are connected to each bus, with at least two-thirds of the devices powered on.

If you must exceed these limits, you can purchase bus extenders and expanders.

Controllers, Talkers, and Listeners

All buses operate under rules that ensure that data passes reliably and that instruments do not use the bus simultaneously. To determine which device has active control of the bus, devices are categorized as *Controllers*, *Talkers*, or *Listeners*. Whenever two devices communicate, one device will be a Talker and the other will be a Listener. In addition, one device will always be a Controller.

Controllers

Most GPIB systems consist of one computer and a variety of instruments. In this type of system, the computer is typically the System Controller. If multiple computers are connected, several devices can have Controller capability, but only one Controller is active, or *Controller-In-Charge* (CIC), at a time. Active control can pass from the current CIC to an idle Controller.

For each GPIB system, you must define a System Controller. You usually define the System Controller through jumper settings on the GPIB interface board, a software configuration file, or both. Only one device on the bus, the System Controller, can make itself the CIC.

The four primary responsibilities of a Controller are the following:

- Defining the communication links.
- Responding to devices requesting service.
- Sending GPIB commands.
- Passing/receiving control.

Talkers and Listeners

You can set most GPIB devices to be either Talkers or Listeners. However, some devices only talk or only listen. Each device accepts its own command set and has its own method of terminating data strings. Talkers and Listeners have the following properties:

- Talkers
  - Are instructed by the Controller to talk.
  - Place data on the GPIB.
  - Permit only one device to talk at a time.
- Listeners
  - Are instructed by the Controller to listen.
  - Read data that the Talker places on the GPIB.
  - Permit several devices to be Listeners simultaneously.

You can compare GPIB operation to a classroom. The instructor (Controller) controls the communication of data between the students (devices). The instructor decides who talks and who listens. On the GPIB, a device cannot talk or listen unless the Controller explicitly tells it to do so.

Figure B-6 shows a system setup example.

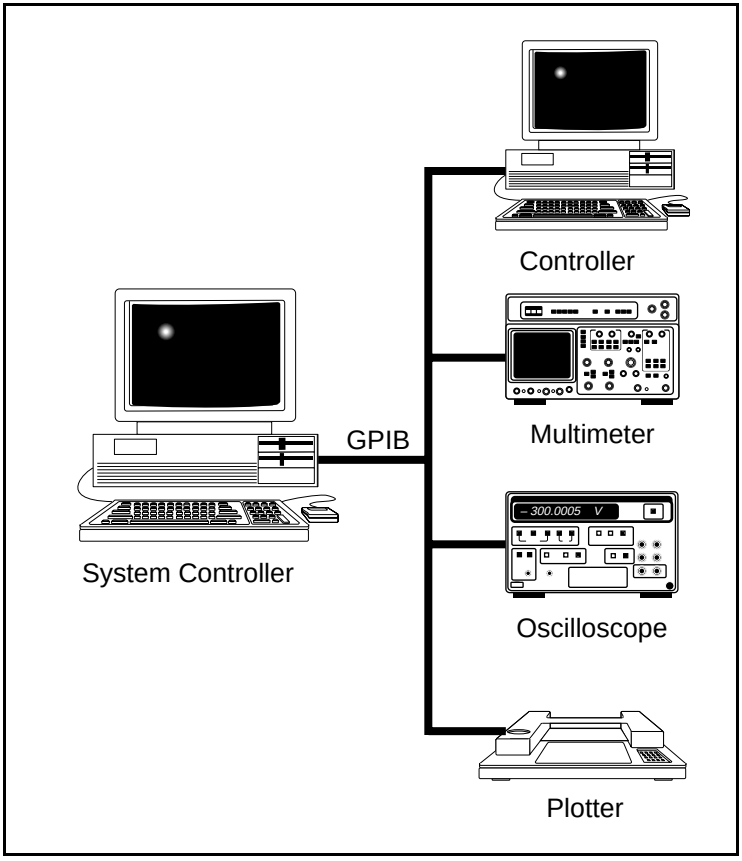


Figure B-6. System Setup Example

Data and Command Messages

In a classroom, when the instructor tells the students who is the Talker and who are the Listeners, his or her information is a command—not the actual data information that the instructor will send. On the GPIB, this distinction is not so intuitive. The bus management line, ATN, determines what type of message you are sending on the bus. If this line is unasserted, the information on the bus is a *data message*; if this line is asserted, the information is a *command message* from the Controller to all devices. The devices on the GPIB monitor the ATN line, determine the data type, and treat the data appropriately.

GPIB Addressing Protocol

In a classroom, an instructor either speaks to the entire class or to a particular student. To speak to a student, the instructor first addresses that student by name.

Addressing on the GPIB follows the same idea. Before any communication can take place on the bus, you must address the Talker and Listener. Before any data passes between devices, the Controller determines who talks and who listens.

In the classroom, we address people by their names. However, on the GPIB, each device (including the Controller) has a unique *primary GPIB address* in the range of 0 to 30 (decimal). The Controller places a command message specifying the addresses of the Talker and Listener devices on the bus.

The Controller sends a single byte (8 bits) of information for a Talker or Listener address command message. Address command messages have the following format:

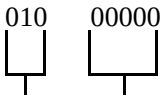
Bit	7	6	5	4	3	2	1	0
Data		TA	LA	X	X	X	X	X

Bits 0 through 4 contain the binary GPIB primary address of the device in communication, and either bit 5—Listener Address (LA)—or bit 6—Talker Address (TA)—will be set if the device is a Talker or a Listener. Bit 7 is never used and is considered a *don't care* bit. For simplicity, assume bit 7 is zero.

Consider an example in which a Controller at primary GPIB address 0 talks to a device at primary GPIB address 1. To establish the communication link, the Controller must send its GPIB talk address and the device's listen address over the GPIB. In this example, these addresses are as follows:

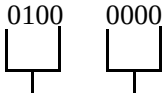
Bit Patterns Sent to Set Up Talker

Bit pattern: 01000000



Talker's GPIB Address is 0

Hexadecimal value: 0100 0000

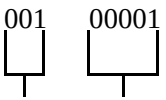


Hex 40 = ASCII "@"

Refer to the *Multiline Interface Command Messages* table (in Appendix C) and find the hex 40 location. On the same row under the *Msg* column, you see the message MTA0, which means *My Talk Address 0.* Hex 40 is the command message for setting device 0 to be a Talker.

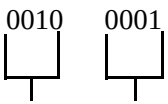
Bit Patterns Sent to Set Up Listener

Bit pattern: 00100001



Listener's GPIB Address is 1

Hexadecimal value: 0010 0001



Hex 21 = ASCII "!"

Refer to the *Multiline Interface Command Messages* table and find the hex 21 location. On the same row under the *Msg* column, you see the message MLA1, which means *My Listen Address 1*. Hex 21 is the command message for setting device 1 to be a Listener.

Reading the Multiline Interface Command Messages Table

By using the *Multiline Interface Command Messages* table, you can understand how the GPIB circuitry interprets the bit patterns to produce the proper message commands. The *Multiline Interface Command Messages* table is organized into four groups of columns. The left or first group of columns (hex 00–1F) represents the primary GPIB addresses. Moving to the right to the next group of columns (hex 20–3F), you will find the corresponding listen addresses (MLA). The listen address of a device is formed by adding hex 20 to the GPIB primary address. Again, move right to the next group of columns (hex 40–5F) for the corresponding talk addresses (MTA). You form the talk address of a device by adding hex 40 to the GPIB primary address.

Secondary Addressing

A device can have a secondary address. A secondary address is in the range of 0 to 30 decimal (IE hex). To form a secondary address command (bit pattern), add 96 decimal (60 hex) to the secondary address. You address a device with a secondary address by sending the primary GPIB address, then the corresponding secondary address. With secondary addressing, you can assign up to 961 talk and listen addresses. Most instruments do not use secondary addressing. In the *Multiline Interface Command Messages* table, the group of columns on the right (hex 60–7F) represents the secondary GPIB address commands.

Unaddressing Command Messages

The CIC uses two special command messages to clear the bus of Talkers and Listeners before assigning new Talkers and Listeners. These command messages are Untalk and Unlisten. The Untalk (UNT) command (hex 5F (ASCII "_")) unaddresses the current Talker. The Untalk command is merely a command for convenience, because addressing one Talker automatically unaddresses all others. The Unlisten (UNL) command (hex 3F (ASCII "?")) unaddresses all current Listeners on the bus. You cannot unaddress only a single Listener if you have previously addressed several Listeners. You must use the UNL command to guarantee that you address only desired Listeners.

Termination Methods

When devices send data over the GPIB, they use up to three different methods to signify the end of a data transfer. These methods are EOS, EOI, and the count method.

Termination methods in GPIB are necessary only for data messages, not for command messages.

EOS Method

The EOS method uses an EOS character, which signifies the termination of data that devices send on the GPIB. This EOS character can be any character. However, it is commonly a carriage return (hex 0D) or a line feed (hex 0A) that the Talker places as the last character in a data string. The Listener reads individual data bytes from the Talker until the Listener reads the EOS character. When the Listener reads the EOS character, it knows that there is no more data, so it completes the read operation.

You must configure the Talker and Listener to use the EOS method before the communication takes place. Many devices send specific EOS characters and look for specific characters from other devices, so it is important for you to read the documentation for all devices to see which termination method the devices use.

To use the EOS method in a classroom setting, the instructor and students would use a certain word to finish all communication within the classroom. As with the GPIB, the instructor and students would define this method and the word used before any communication took place. In the GPIB and in the classroom, the termination signal is sent by using the normal data path (data lines in GPIB, or speech in the classroom).

EOI Method

The EOI method uses the GPIB EOI line, which is separate from the eight data lines on the GPIB. In the EOI method, when the Talker sends the last byte of data in the transmission, it sets the EOI line high to specify that the byte is the last byte to be sent. The Listener monitors the EOI line and recognizes when there is no more data. You must establish ahead of time whether the Talker will use the EOI method, so you can correctly configure the Listener to watch the EOI line.

Students could use the EOI method in the classroom: they would wave device cards in the air to signal when they have finished speaking. This form of communication is separate from the method of sending data (speech), but the other Listeners can monitor this communication while they receive data (hear the speech).

Count Method

The count method uses neither the EOI line nor the EOS character. In the count method, the device that receives information specifies the number of bytes to read. Through this method, a listening device reads a specified amount of data and prevents the talking device from sending more data. If you do not clear the remaining data from the bus, you can recover it later.

Students can use the count method in the classroom. Students count the words of someone who is talking. The Listener announces that he or she will listen to only a

specified number of words. Beyond this number of words, the Listener will not hear any further information from the Talker. If the Listener wants more information, he or she requests more words from the Talker.

Combinations of Termination Methods

You can use any combination of the three termination methods to terminate communication on the GPIB. For example, you can specify an EOS character and also use the EOI line method. In this case, when the end of the string is reached, the device sending the data will send an EOS character and assert the EOI line. When you use more than one method, the first termination method recognized causes the termination. In this example, the EOS character or EOI line causes termination, depending on which method the device recognizes first.

In general, when you use more than one termination method at a time, all methods are logically ORed together for a result. Therefore, if you use all three methods, the communication termination will take place if the device sees the EOS character, the system asserts the EOI line, or the count value has been reached.

Serial Polling

Servicing SRQs

In the classroom, an instructor is in charge of the class and controls activity. The GPIB works in a similar fashion: the Controller bus controls when tasks are performed. In the classroom, a student must have permission to speak, and on the GPIB, no device can communicate unless it is addressed to talk on the bus. A device may, however, need to communicate with the Controller before the Controller tells it to talk. In a classroom, students who have something to say usually raise their hands. On the GPIB, any device can assert the SRQ line, which is separate from the data lines. SRQ informs the Controller that a device needs attention. The next section discusses how the SRQ line is asserted and how the device that asserts it is identified.

Serial Polling Devices

This section investigates how the GPIB handles the SRQ line. Remember the SRQ line purpose: signaling to the Controller that a device needs attention. When SRQ is asserted, it is the responsibility of the Controller to determine who requested service by checking all devices individually. Checking the devices individually is known as *polling* the devices. The Controller can poll devices in two ways: in serial or in parallel. This appendix discusses serial polling and parallel polling.

Serial polling obtains specific information from a device. When you serial poll, the Controller sends a special command message—Serial Poll Enable (SPE)—to the device,

directing it to return its serial poll status byte. The SPE message sets the IEEE 488.1 serial poll mode in the device, so when the device is addressed to talk, it returns a single 8-bit status byte. This serial poll status byte is different for each type of instrument; except for one bit, you must refer to the instrument user manual for information on the other bits. Bit 6 (hex 40) of any serial poll status byte indicates whether a device requested service by asserting the SRQ line. The device uses the other seven bits of the status byte to specify why it needs attention.

After the Controller reads the status byte, it sends another command message, Serial Poll Disable (SPD), to the device. The SPD message terminates the serial poll mode, thus returning the device to its normal Talker/Listener state. Once a device requesting service is serial polled, it usually unasserts the SRQ line.

When a serial poll is conducted, the following sequence of events occurs:

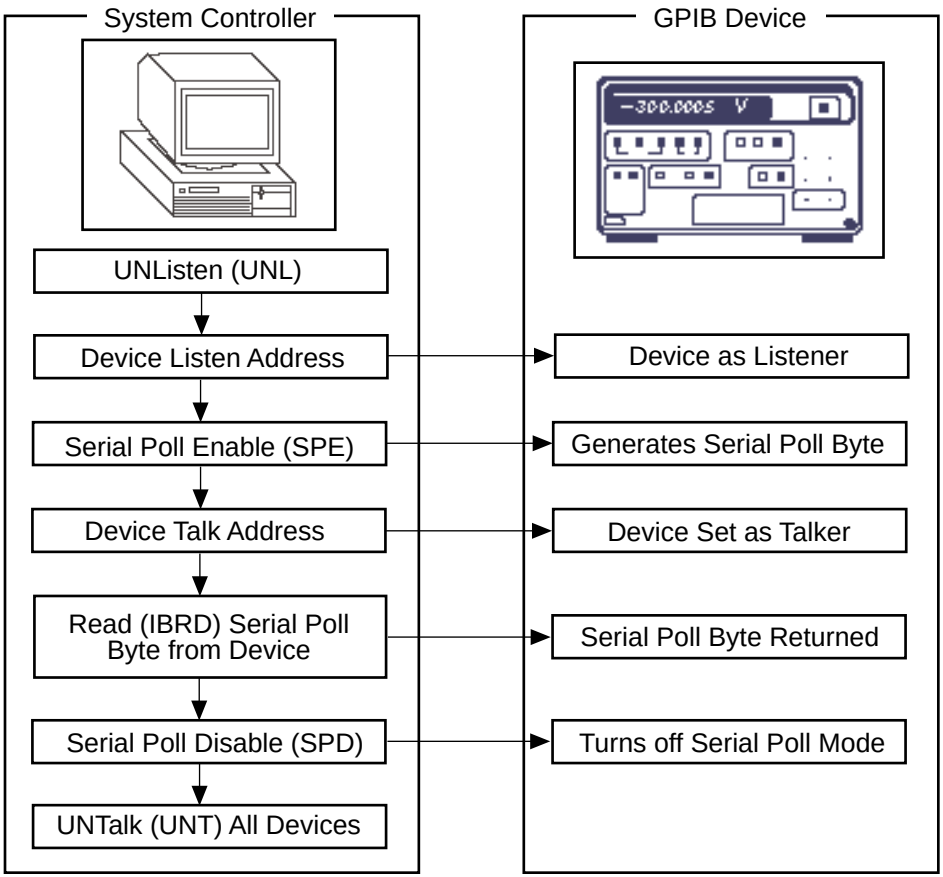


Figure B-7. Events During a Serial Poll

Status Byte Model for IEEE 488.1

IEEE 488.1 defines only bit 6, the RQS bit, of the serial poll status byte (see Table B-1). If a device is requesting service, it sets RQS. The meaning of the remaining bits is device dependent.

Table B-1. IEEE 488.1 Standard Status Data Structure

7	RQS	5	4	3	2	1	0	Status Byte Register

ESR and SRE Registers

The IEEE 488.2 standard defines a set of commands for controlling the GPIB. The standard also defines a new method of working with the SRQ line on the GPIB. This section applies only to those GPIB devices that are IEEE 488.2 compatible. If a device is only IEEE 488.1 compatible, the previous section applies.

Status Byte Model for IEEE 488.2

IEEE 488.2 describes a scheme for status reporting. This scheme is required for all IEEE 488.2 instruments. With this scheme, the Controller can obtain status information for every instrument in the system. This scheme builds on and extends the IEEE 488.1 status byte shown in Table B-1. Three bits of this status byte are defined. The IEEE 488.2 standard defines the RQS bit like the IEEE 488.1 standard. IEEE 488.2 adds the Event Status Bit (ESB) and the Message Available (MAV) bit. The manufacturer defines other bits. The RQS bit indicates the device has requested service by asserting the SRQ line. The ESB indicates that one of the standard events defined in the Standard Event Status Register has occurred. By setting the corresponding bits in the Standard Event Status Enable Register, you define which standard events will set the ESB. The MAV bit indicates whether a message is available in the instrument output queue. By setting the corresponding bits in the Service Request Enable Register, you can configure an instrument to assert the SRQ line based on the bits of its status register.

IEEE 488.2 defines a dual role for the RQS bit. This bit is also known as the Master Summary Status (MSS) bit. The MSS bit indicates whether there is at least one reason for the instrument to request service. The status of this bit is returned only in response to the status byte (STB) query; its status is not sent in response to a serial poll because this bit is not part of the IEEE 488.1 status byte (see Figure B-8).

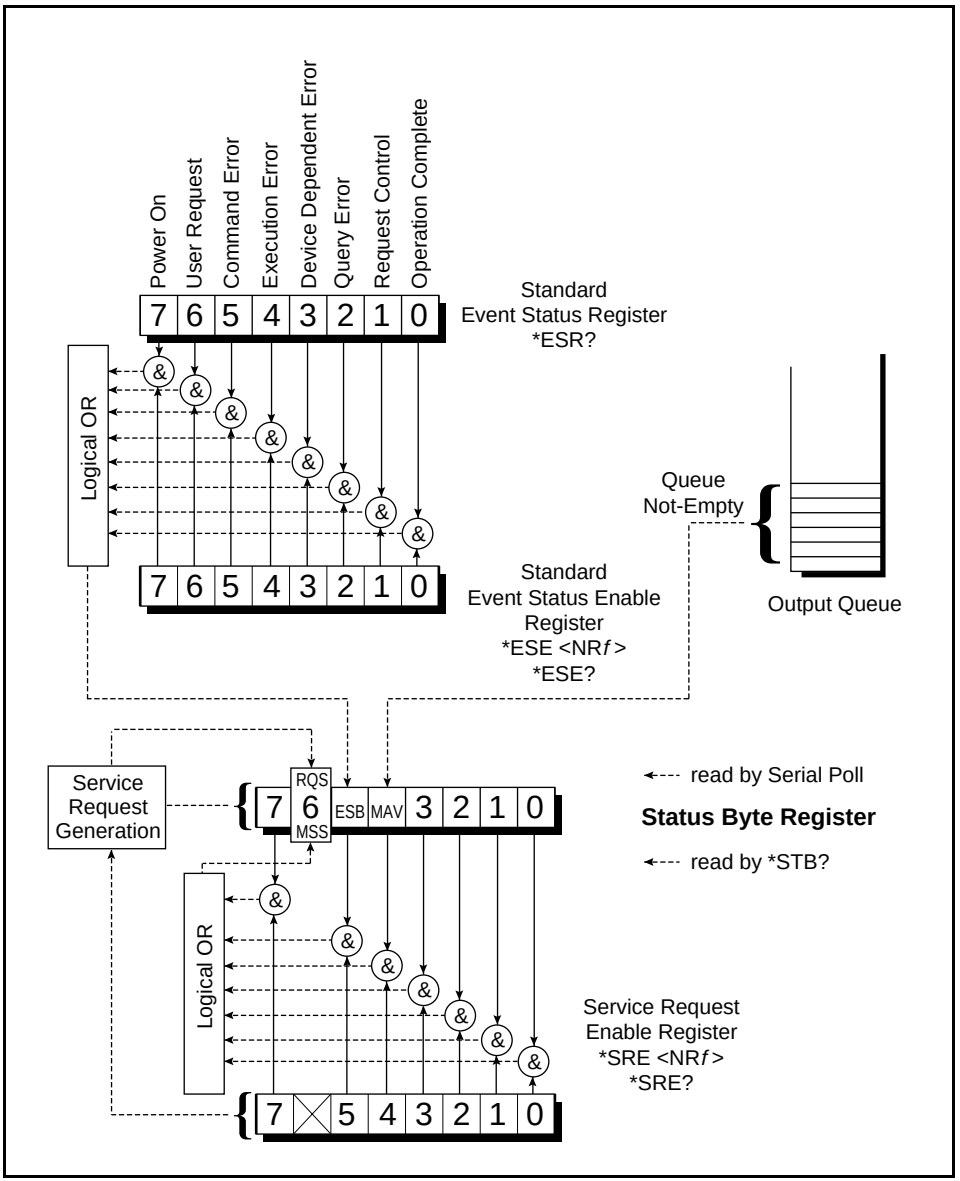


Figure B-8. IEEE 488.2 Standard Status Structures

Parallel Polling

Parallel polling is another way to get information from a device that requests service. Parallel polling differs from serial polling in two ways: all configured devices are polled simultaneously (that is, in parallel) and a Controller initiates a parallel poll sequence (any device requests the initiation of a serial poll sequence).

Overview of Parallel Polls

A parallel poll is an exchange of messages between the Controller and other system devices. The Controller sends the IDY message true to the other devices; each device responds to the IDY message by sending one PPR message (PPR1, PPR2, PPR3, PPR4, PPR5, PPR6, PPR7, or PPR8) to the Controller. Each device usually sends a different PPR message. (See the *Physical Representation of the PPR Message* section in this chapter.) Each device can send its PPR message either true or false. See Figure B-9.

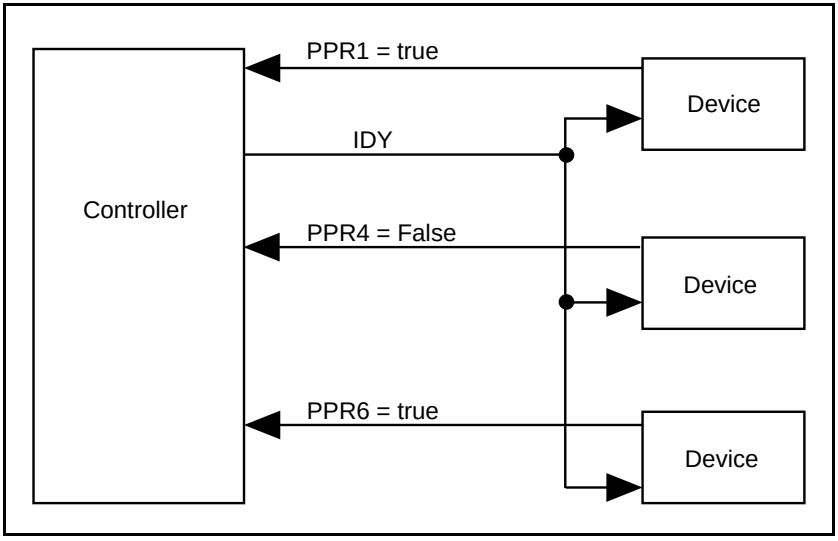


Figure B-9. Example Exchange of Messages During a Parallel Poll

Determining the Value of the PPR Message

Each device examines its local ist message and its Sense bit (S) to determine whether it will send its PPR message true or false. Table B-2 illustrates how the ist message and the Sense bit affect the value of the PPR message.

Table B-2. PPR Message Value

ist Message	Sense Bit (S)	PPR Message Sent
0 (False)	0	True
0 (False)	1	False
1 (True)	0	False
1 (True)	1	True

The ist message usually reflects a bit of status information about the device. For example, when the device has taken a measurement, it can assert its local ist message. The Sense bit is part of the configuration of a device. Each device has an independent Sense bit.

The meaning of the PPR message and the local ist message is device dependent.

Configuring a Device for Parallel Polls

To configure a device to respond to parallel polls, you must supply the device with two pieces of data:

- The PPR message that the device should send to the Controller (PPR1, PPR2, . . , or PPR8)
- The value of the Sense bit of the device.

You can configure devices locally or remotely. You *locally* configure (Parallel Poll function subset PP2) a device by setting knobs or switches on the front panel of the device (or by physically manipulating the device in some other way). You *remotely* configure (Parallel Poll function subset PP1) a device by sending messages across the GPIB from the Controller to the device. If a device has not been configured to respond to parallel polls, it does not respond to parallel polls.

Some devices support only local configuration and some support only remote configuration. Some devices do not support any parallel polls (Parallel Poll function subset PP0).

Determining the PPE Message

The PPE message contains the parallel poll configuration data for a device. Table B-3 shows how you determine the value of DIO[7:1] for the PPE message. As with all commands, the DIO[8] is a *don't care* bit.

Table B-3. Determining the PPE Message

Sense Bit (S)	PPR Message to Send	PPE Message (hex)
0	PPR1	60
0	PPR2	61
0	PPR3	62
0	PPR4	63
0	PPR5	64
0	PPR6	65
0	PPR7	66
0	PPR8	67
1	PPR1	68
1	PPR2	69
1	PPR3	6A
1	PPR4	6B
1	PPR5	6C
1	PPR6	6D
1	PPR7	6E
1	PPR8	6F

Physical Representation of the PPR Message

To send a PPR message true, a device drives the corresponding GPIB DIO signal low with an open-collector driver. For example, to send the PPR4 message true, a device drives the GPIB DIO4 signal low.

Because devices drive the DIO signals with open-collector drivers during parallel polls, more than one device can share a PPR message. If a Controller detects a PPR message being sent true, the Controller knows that one or more of the devices sharing the PPR message is sending the PPR message true.

Clearing and Triggering Devices

A Controller can clear devices in several ways. It can assert the IFC line to clear all devices, or it can send the Device Clear (DCL) command message to clear all devices on the bus. To clear a single device, a Controller can address the device to listen, then send the Selected Device Clear (SDC) command message.

After a device receives DCL or SDC, its *clear* state is device dependent. Generally, sending DCL or SDC is a less extreme method of clearing a device than asserting IFC. Most devices support the DCL and SDC method; all devices support the IFC method.

All devices in multidevice measurement systems must often be sampled as closely together as possible. You can trigger devices simultaneously by using the Group Execute Trigger (GET) command message. This command message causes all devices that have triggering capability and that are currently addressed to initiate a preprogrammed action. The action could be, for example, to take a measurement or begin a sweep.

Appendix C
Standard Commands for Programmable Instruments (SCPI)

This appendix discusses the Standard Commands for Programmable Instruments (SCPI) document, the required SCPI commands, and SCPI programming.

GPIB instrumentation standards have progressed from the IEEE 488.1 standard to the IEEE 488.2 standard to SCPI. The IEEE 488.1 standard simplified and standardized the interconnection of programmable instrumentation by defining the electrical, mechanical, and protocol specifications of the GPIB. Before IEEE 488.1, each manufacturer had its own proprietary interface.

The IEEE 488.2 standard kept the IEEE 488.1 standard intact, but it made systems more compatible and program development easier by defining standard data codes and formats, a status-reporting model, a message exchange protocol, a set of common commands for all instruments, and Controller requirements. Because the IEEE 488.1 standard did not address these issues, manufacturers implemented each item differently, thus creating complex programming and unpredictable development costs.

SCPI uses the IEEE 488.2 standard as a basis for defining a single, comprehensive command set that is suitable for all instruments. SCPI users no longer need to learn a different command set for each instrument in their systems.

You can use IEEE 488.1, IEEE 488.2, and SCPI instruments and Controllers together, but you achieve the maximum benefits with a system consisting of an IEEE 488.2 Controller and SCPI instruments.

IEEE 488.2 Common Commands Required by SCPI

All SCPI devices require the mandatory common commands that the IEEE 488.2 standard defines (see Table C-1). This command set consists of program commands and status queries that are common to all devices. These commands and queries do not handle device-specific operations; they handle more general operations such as device identification, operation synchronization, standard event status enabling and reporting, device reset and self-test, and service request enable reporting.

Table C-1. IEEE 488.2 Common Commands Required by SCPI

Command	Description
*CLS	Clear Status Command
*ESE	Standard Event Status Enable Command
*ESE?	Standard Event Status Enable Query
*ESR?	Standard Event Status Register Query
*IDN?	Identification Query
*OPC	Operation Complete Command
*OPC?	Operation Complete Query
*RST	Reset Command
*SRE	Service Request Enable Command
*SRE?	Service Request Enable Query
*STB?	Read Status Byte Query
*TST?	Self-Test Query
*WAI?	Wait-to-Continue Command

SCPI Required Commands

In addition to the IEEE 488.2 common commands and queries, SCPI defines its own set of required common commands (see Table C-2). In general, these commands build on the IEEE 488.2 common command set, but SCPI expands the standard status-reporting model defined in IEEE 488.2 with OPERation and QUEStionable status registers. For both of these registers, commands read the contents of the EVENt and CONDition registers, set the ENABle mask, and read the ENABle mask.

The SYSTem command set defines functions that are not related to instrument performance, such as commands for performing general housekeeping like setting TIME or SECurity. The subcommand query ERRor? requests the next entry from the error/event queue of the device. The PRESet command configures the SCPI and device-dependent status registers to be reported through the SCPI status-reporting model.

Table C-2. SCPI Required Commands

Command	Description
:SYSTem	Collects functions not related to instrument performance
:ERRor?	Requests the next entry from the instrument's error queue
:STATus	Controls the SCPI-defined status-reporting structures
:OPERation	Selects the Operation structure
[:EVENt]?	Returns the contents of the Event register
:CONDition?	Returns the contents of the Condition register
:ENABle	Reads the Enable mask
:QUEStionable	Selects the Questionable structure
[:EVENt]?	Returns the contents of the Event register
:CONDition	Returns the contents of the Condition register
:ENABle	Sets the Enable mask, which allows event reporting
:ENABle?	Reads Enable mask
:PRESet	Enables all required event reporting

SCPI Optional Commands

The SCPI command set that an instrument uses can include a subset of the commands covered in the SCPI specification. An instrument designed to measure voltage does not implement commands to measure frequency. An instrument can also support special commands not presently covered in the SCPI standard.

SCPI commands are not case sensitive. Moreover, a command such as TRIGger can be issued as TRIGGER or as its short-form mnemonic, TRIG. However, SCPI does not recognize any other version of this command. For example, TRIGG is *not* a valid command.

Programming with SCPI

The functional blocks of the SCPI Instrument model define the command categories. These categories, along with some other general categories, have a hierarchical structure of subcommands and parameters for more specific functions (see Figure C-1).

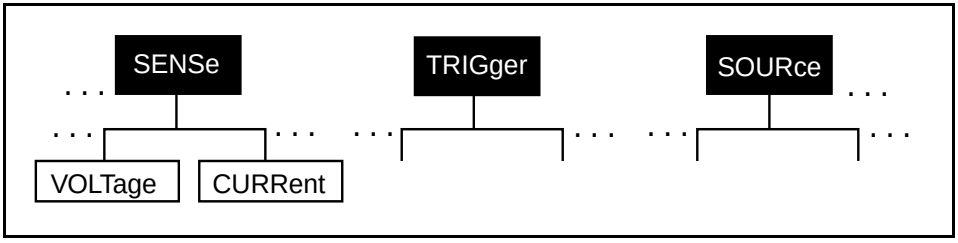


Figure C-1. Partial Command Categories

Most instruments require commands to execute a specific function. For example, a digital voltmeter can require the MEASure, VOLTage, and AUTO commands to take a voltage reading. To properly interpret these commands, SCPI defines a hierarchical command structure called a command tree. Figure C-2 illustrates a simple command tree for the SENSe command subsystem.

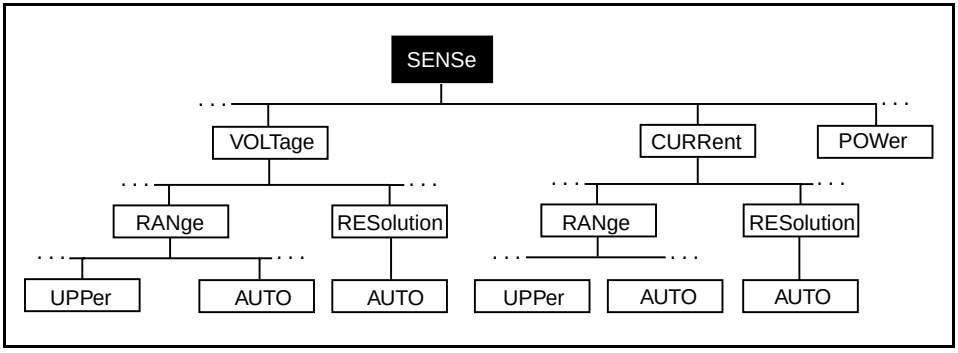


Figure C-2. Simple Command Tree for the SENSe Command Subsystem

The SENSe commands control the characteristics of the conversion process for the input sensors of the instrument. Examples include the following:

- Signal amplitude for VOLTage, CURRent, and POWer
- Filter BANDwidth
- FREQuency characteristics

The SENSe commands do not mathematically manipulate the data after it has been converted.

Constructing SCPI Commands by Using the Hierarchical Command Structure

The SENSe commands program an instrument to control the conversion of the signal into internal data that can be manipulated. SENSe commands control such parameters as range, resolution, gate time, and normal mode rejection. By using the partial command tree shown in Figure C-3, you can construct the short form command to configure an instrument for a voltage measurement that uses dynamic autoranging. This command is as follows:

```
SENS:VOLT:RANG:AUTO:DIR:EITH
```

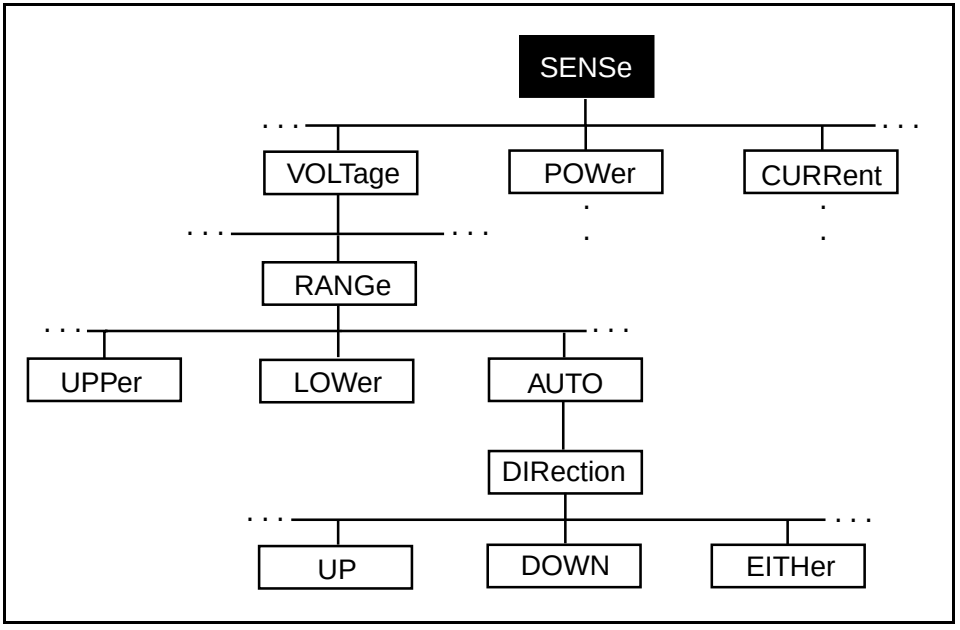


Figure C-3. Partial Command Tree for the SENSe Command Subsystem

The SOURce commands program the instrument to generate a signal based on specified characteristics and internal data. SOURce block functions specify such signal parameters as amplitude modulation, power, current, voltage, and frequency. By using the partial command tree shown in Figure C-4, you can construct the short form command to set the upper limit of the current output to 500 mA. This command is as follows:

```
SOUR:CURR:LIM:HIGH 0.5
```

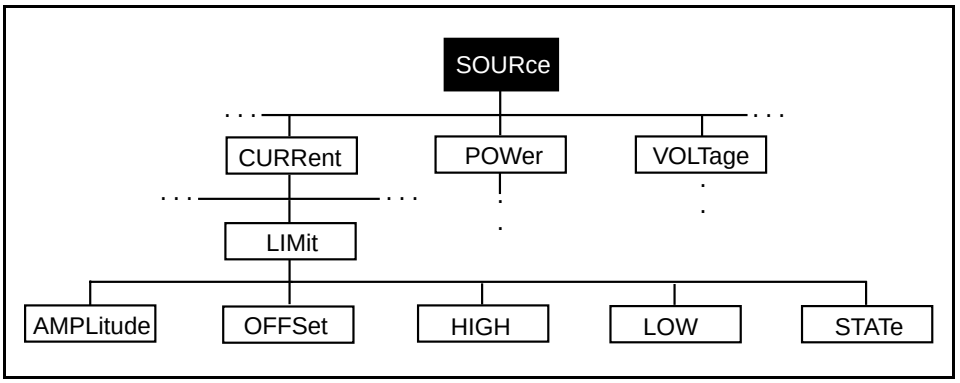


Figure C-4. Partial Command Tree for the SOURce Command Subsystem

The TRIGger commands program the instrument to synchronize its operation based on some event. Trigger sources include an internal event or condition involving the instrument functionality, an external condition such as an analog or digital signal, or a software command. By using the partial command tree shown in Figure C-5, you can construct the short form command to trigger an instrument from an external source. This command is as follows:

```
TRIG:SOUR:EXT
```

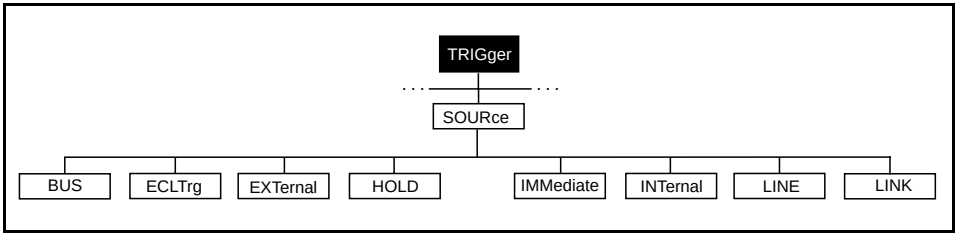


Figure C-5. Partial Command Tree for the TRIGger Command Subsystem

Parsing SCPI Commands

Colons separate each command and instruct the instrument parser to move down a level in the command tree hierarchy. In situations where two commands are issued without changing levels, a semicolon separates the commands. Commas separate parameters such as numeric, extended numeric, discrete, and Boolean. Commas are generally ignored with two exceptions:

- Spaces should not break command words.
- Spaces must not separate commands and parameters.

The colon preceding the first command in a SCPI message instructs the parser in the SCPI instrument to reset itself to the root level in the hierarchy.

Unless specifically noted, all commands have a query form as defined in the IEEE 488.2 standard. When a query command is received, the current instrument settings associated with that command are placed in the instrument output buffer. For more commands, consult the SCPI standard or the user manuals for the SCPI instruments of interest in a particular application.

Appendix D
Multiline Interface Command Messages

This appendix lists the multiline interface messages and describes the mnemonics and messages that correspond to the interface functions. The multiline interface messages are IEEE 488 defined commands that are sent and received with ATN TRUE. The interface functions include initializing the bus, addressing and unaddressing devices, and setting device modes for local or remote programming.

Multiline Interface Messages

Hex	Oct	Dec	ASCII	Msg	Hex	Oct	Dec	ASCII	Msg
00	000	0	NUL		20	040	32	SP	MLA0
01	001	1	SOH	GTL	21	041	33	!	MLA1
02	002	2	STX		22	042	34	"	MLA2
03	003	3	ETX		23	043	35	#	MLA3
04	004	4	EOT	SDC	24	044	36	$	MLA4
05	005	5	ENQ	PPC	25	045	37	%	MLA5
06	006	6	ACK		26	046	38	&	MLA6
07	007	7	BEL		27	047	39	'	MLA7
08	010	8	BS	GET	28	050	40	(	MLA8
09	011	9	HT	TCT	29	051	41	)	MLA9
0A	012	10	LF		2A	052	42	*	MLA10
0B	013	11	VT		2B	053	43	+	MLA11
0C	014	12	FF		2C	054	44	,	MLA12
0D	015	13	CR		2D	055	45	-	MLA13
0E	016	14	SO		2E	056	46	.	MLA14
0F	017	15	SI		2F	057	47	/	MLA15
10	020	16	DLE		30	060	48	0	MLA16
11	021	17	DC1	LLO	31	061	49	1	MLA17
12	022	18	DC2		32	062	50	2	MLA18
13	023	19	DC3		33	063	51	3	MLA19
14	024	20	DC4	DCL	34	064	52	4	MLA20
15	025	21	NAK	PPU	35	065	53	5	MLA21
16	026	22	SYN		36	066	54	6	MLA22
17	027	23	ETB		37	067	55	7	MLA23
18	030	24	CAN	SPE	38	070	56	8	MLA24
19	031	25	EM	SPD	39	071	57	9	MLA25
1A	032	26	SUB		3A	072	58	:	MLA26
1B	033	27	ESC		3B	073	59	;	MLA27
1C	034	28	FS		3C	074	60	<	MLA28
1D	035	29	GS		3D	075	61	=	MLA29
1E	036	30	RS		3E	076	62	>	MLA30
1F	037	31	US		3F	077	63	?	UNL

Message Definitions

DCL	Device Clear
GET	Group Execute Trigger
GTL	Go To Local
LLO	Local Lockout
MLA	My Listen Address
MSA	My Secondary Address
MTA	My Talk Address
PPC	Parallel Poll Configure
PPD	Parallel Poll Disable

Multiline Interface Messages

Hex	Oct	Dec	ASCII	Msg	Hex	Oct	Dec	ASCII	Msg
40	100	64	@	MTA0	60	140	96	`	MSA0,PPE
41	101	65	A	MTA1	61	141	97	a	MSA1,PPE
42	102	66	B	MTA2	62	142	98	b	MSA2,PPE
43	103	67	C	MTA3	63	143	99	c	MSA3,PPE
44	104	68	D	MTA4	64	144	100	d	MSA4,PPE
45	105	69	E	MTA5	65	145	101	e	MSA5,PPE
46	106	70	F	MTA6	66	146	102	f	MSA6,PPE
47	107	71	G	MTA7	67	147	103	g	MSA7,PPE
48	110	72	H	MTA8	68	150	104	h	MSA8,PPE
49	111	73	I	MTA9	69	151	105	i	MSA9,PPE
4A	112	74	J	MTA10	6A	152	106	j	MSA10,PPE
4B	113	75	K	MTA11	6B	153	107	k	MSA11,PPE
4C	114	76	L	MTA12	6C	154	108	l	MSA12,PPE
4D	115	77	M	MTA13	6D	155	109	m	MSA13,PPE
4E	116	78	N	MTA14	6E	156	110	n	MSA14,PPE
4F	117	79	O	MTA15	6F	157	111	o	MSA15,PPE
50	120	80	P	MTA16	70	160	112	p	MSA16,PPD
51	121	81	Q	MTA17	71	161	113	q	MSA17,PPD
52	122	82	R	MTA18	72	162	114	r	MSA18,PPD
53	123	83	S	MTA19	73	163	115	s	MSA19,PPD
54	124	84	T	MTA20	74	164	116	t	MSA20,PPD
55	125	85	U	MTA21	75	165	117	u	MSA21,PPD
56	126	86	V	MTA22	76	166	118	v	MSA22,PPD
57	127	87	W	MTA23	77	167	119	w	MSA23,PPD
58	130	88	X	MTA24	78	170	120	x	MSA24,PPD
59	131	89	Y	MTA25	79	171	121	y	MSA25,PPD
5A	132	90	Z	MTA26	7A	172	122	z	MSA26,PPD
5B	133	91	[	MTA27	7B	173	123	{	MSA27,PPD
5C	134	92	\	MTA28	7C	174	124	|	MSA28,PPD
5D	135	93	]	MTA29	7D	175	125	}	MSA29,PPD
5E	136	94	^	MTA30	7E	176	126	~	MSA30,PPD
5F	137	95	_	UNT	7F	177	127	DEL	

Message Definitions

PPE	Parallel Poll Enable
PPU	Parallel Poll Unconfigure
SDC	Selected Device Clear
SPD	Serial Poll Disable
SPE	Serial Poll Enable
TCT	Take Control
UNL	Unlisten
UNT	Untalk

Appendix E
Mnemonics Key

This appendix defines the mnemonics (abbreviations) that this manual uses for functions, remote messages, local messages, states, bits, registers, integrated circuits, and system functions.

The mnemonic types in this key are abbreviated to mean the following:

A	Auxiliary or Accessory Commands
B	Bit
F	Function
IC	Integrated Circuit
LM	Local Message
P	Physical Device Pin
R	Register
RM	Remote Message
SF	System Function
ST	State

Mnemonic	Type	Definition
A		
ACCRA	R	Accessory Register A
ACCRB	R	Accessory Register B
ACCRE	R	Accessory Register E
ACCRF	R	Accessory Register F
ACCRI	R	Accessory Register I
ACCRJ	R	Accessory Register J
ACCR	R	Accessory Register
ACCWR	R	Accessory Write Register
ACDS	ST	Acceptor Data State (AH function)
ACG	RM	Addressed Command Group
ACRDY	B	Acceptor Ready State bit
ACRS	ST	Acceptor Ready State
AD[5–1]	B	NAT7210 GPIB Address bits 5 through 1
ADHS	B	Acceptor Data Holdoff State bit
ADMR	R	Address Mode Register
ADR	R	Address Register
ADR0	R	Address Register 0
ADR1	R	Address Register 1
ADSC	B	Address Status Change bit
ADSC IE	B	Address Status Change Interrupt Enable bit
ADSR	R	Address Status Register
AEFN	B	FIFO A Empty Flag bit
AEHS	B	Acceptor End Holdoff State bit
AH	F	Acceptor Handshake function
AH1	F	Acceptor Handshake
ANHS1	B	Acceptor Not Ready Holdoff bit
ANHS2	B	Acceptor Not Ready Holdoff Immediately bit
ANRS	ST	Acceptor Not Ready State
APT	B	Address Pass Through bit
APT IE	B	Address Pass Through Interrupt Enable bit
ARS	B	Address Register Select bit
ATCT	B	Automatic Take Control bit
ATN	RM	Attention
ATN*	B	Attention bit
ATN IE	B	Attention Interrupt Enable bit
ATNI	B	ATN Interrupt bit
ATNI IE	B	ATN Interrupt Enable bit
AUXCR	R	Auxiliary Command Register

Mnemonic	Type	Definition
AUXMR	R	Auxiliary Mode Register
AUXRA	R	Auxiliary Register A
AUXRB	R	Auxiliary Register B
AUXRE	R	Auxiliary Register E
AUXRF	R	Auxiliary Register F
AUXRG	R	Auxiliary Register G
AUXRI	R	Auxiliary Register I

B

BCR	R	Bus Control Register
BI	B	Byte In bit
BI IE	B	Byte In Interrupt Enable bit
BIN	B	Binary bit
BO	B	Byte Out bit
BO IE	B	Byte Out Interrupt Enable bit
BSR	R	Bus Status Register

C

CDOR	R	Command/Data Out Register
CHES	B	Clear Holdoff On End Select bit
ch_rst	A	Chip Reset auxiliary command
CIC	B	Controller-In-Charge bit
CIC IE	B	Controller-In-Charge Interrupt Enable bit
Clear ERR	A	Clear ERR Interrupt auxiliary command
clrpi	A	Clear Page-In Registers auxiliary command
CMDR	R	Command Register
CO	B	Command Out bit
CO IE	B	Command Out Interrupt Enable bit
cont		Continuous mode
CPT	B	Command Pass Through bit
CPT ENABLE	B	Command Pass Through Enable bit
CPT IE	B	Command Pass Through Interrupt Enable bit
CPTR	R	Command Pass Through Register

Mnemonic	Type	Definition

D

DAC	RM	Data Accepted
dacr	A	Release DAC Holdoff auxiliary command
dai	A	Disable IMR2, IMR1, And IMR0 Interrupts auxiliary command
dai	B	Disable IMR2, IMR1, And IMR0 Interrupts bit
dal	B	Disable Listener bit
dat	B	Disable Talker bit
DAV	RM	Data Valid
DAV	B	GPIB Data Valid Signal bit
DCAS	B	Device Clear Active State bit
DCAS IE	B	Device Clear Active State Interrupt Enable bit
DCL	RM	Device Clear
DEC	B	Device Clear bit
DEC IE	B	Device Clear Interrupt Enable bit
DET	B	Device Execute Trigger bit
DET IE	B	Device Execute Trigger Interrupt Enable bit
DHADC	B	DAC Holdoff On DCL Or SDC Command bit
DHADT	B	DAC Holdoff On GET Command bit
DHALA	B	DAC Holdoff On All Listener Addresses Command bit
DHALL	B	DAC Holdoff On All UCG, ACG, And SCG Commands bit
DHATA	B	DAC Holdoff On All Talker Addresses Command bit
DHDC	B	DAC Holdoff On DCAS Command bit
DHDT	B	DAC Holdoff On DTAS Command bit
DHUNTL	B	DAC Holdoff On The UNL Or UNT Command bit
DI	B	Data In bit
DI IE	B	Data In Interrupt Enable bit
DIO	P	GPIB Data Input/Output pins
DIR	R	Data In Register
DISTCT	B	Disable Automatic Take Control bit
DL	B	Disable Listener bit
DL0	B	Disable Listener 0 bit
DL1	B	Disable Listener 1 bit
DMAE	B	DMA Enable bit
DMAEN	B	DMA Enable bit
DMAI	B	DMA Input Enable bit
DMAO	B	DMA Output Enable bit
DO	B	Data Out bit
DO IE	B	Data Out Interrupt Enable bit
DRQ	B	DMA Request Pin Status bit
DT	B	Disable Talker bit
DT	F	Device Trigger function
DT0	B	Disable Talker 0 bit

Mnemonic	Type	Definition
DT1	B	Disable Talker 1 bit
DTAS	ST	Device Trigger Active State

E

edpa	B	Enable Dual Primary Addressing Mode bit
END	B	End Received bit
END IE	B	End Received Interrupt Enable bit
END RX	B	End Received bit
EOI	RM	End or Identify
EOI	B	End-or-Identify bit
EOIOE	B	GPIB EOI Signal Output Enable bit
EOS	RM	End of String
EOS0	B	End-of-String bit 0
EOS1	B	End-of-String bit 1
EOS2	B	End-of-String bit 2
EOS3	B	End-of-String bit 3
EOS4	B	End-of-String bit 4
EOS5	B	End-of-String bit 5
EOS6	B	End-of-String bit 6
EOS7	B	End-of-String bit 7
EOSR	R	End-of-String Register
ERR	B	Error bit
ERR IE	B	Error Interrupt Enable bit
ESB	B	Event Status Bit
EXTDAC		External DAC

F

F[3–0]		Clock Frequency
feoi	A	Send EOI With The Next Byte
fget	A	Force Group Execute Trigger auxiliary command

G

GET	B	Group Execute Trigger bit
GET IE	B	Group Execute Trigger Interrupt Enable bit
GLINT	B	Global Interrupt Enable bit
gts	A	Go To Standby auxiliary command

Mnemonic	Type	Definition
H		
hdfa	A	Holdoff On All Data auxiliary command
hdfe	A	Holdoff On End Only auxiliary command
hlda		RFD Holdoff On All Data mode
HLDA	B	Holdoff On All Data bit
hlde		Holdoff On All END mode
hlde	A	Holdoff On All END Mode auxiliary command
HLDE	B	Holdoff On End bit
hldi	A	Holdoff Handshake Immediately auxiliary command
HSTS	ST	High-Speed T1 State
I		
ICR	R	Internal Count Register
ICR2	R	Internal Count Register 2
IDY	RM	Identify
IFC	RM	Interface Clear
IFC IE	B	Interface Clear Interrupt Enable bit
IFCI	B	IFC Interrupt bit
IFCI IE	B	IFC Interrupt Enable bit
IMR0	R	Interrupt Mask Register 0
IMR1	R	Interrupt Mask Register 1
IMR2	R	Interrupt Mask Register 2
INT	B	Interrupt Request Pin bit
INT0	B	Interrupt Register 0 Interrupt bit
INV	B	Invert bit
INT1	B	Interrupt Register 1 Interrupt bit
ISR0	R	Interrupt Status Register 0
ISR1	R	Interrupt Status Register 1
ISR2	R	Interrupt Status Register 2
ISS	B	Individual Status Select bit
ist	A	Parallel Poll Flag auxiliary command
L		
L	F	Listen
LA	B	Listener Active bit
LACS	ST	Listener Active State
LADCS	ST	Listener Addressed Or Active State
LADS	ST	Listener Addressed State (L function)
LIDS	ST	Listener Idle State
LLO	B	Local Lockout bit
LLOC	B	Local Lockout Change bit

Mnemonic	Type	Definition
LLOC IE	B	Local Lockout Change Interrupt Enable bit
LOCS	ST	Local State
LOK	B	Lockout bit
LOKC	B	Lockout Change bit
LOKC IE	B	Lockout Change Interrupt Enable bit
lon	B	Listen-Only bit
lon	LM	Listen Only
LPAS	B	Listener Primary Addressed State bit
LPAS	ST	Listener Primary Addressed State
LPIS	ST	Listener Primary Idle State
ltn	A	Listen auxiliary command
lul	A	Unlisten auxiliary command
lun	LM	Local Unlisten
lut	A	Local Untalk auxiliary command
LWC	B	Listen When Controller bit

M

MA	B	My Address bit
MA IE	B	My Address Interrupt Enable bit
MAC	B	My Address Change bit
MAC IE	B	My Address Change Interrupt Enable bit
MAV	B	Message Available bit
MICR	R	Modify Internal Count Register
MICR	B	Modify Internal Count Register bit
MJMN	B	Major-Minor bit
MLA	RM	My Listen Address
MSA	RM	My Secondary Address
MSS	B	Master Summary Status bit
MTA	RM	My Talk Address
MAT6	RM	My Talk Address 6

N

nba	B	New Byte Available local message bit
nba	LM	New Byte Available
nbaf	A	New Byte Available False auxiliary command
nbaf	B	New Byte Available False bit
NDAC	B	Not Data Accepted bit
NL	B	New Line Receive bit
NL IE	B	New Line Receive Interrupt Enable bit
NLEN	B	New Line End Enable bit
nonvalid	A	Nonvalid auxiliary command issued
NPRS	ST	Negative Poll Response State

Mnemonic	Type	Definition
NRFD	RM	Not Ready For Data Message
NTNL	B	No Talking When No Listener bit

O

OSA	RM	Other Secondary Address

P

P1	B	Parallel Poll Response bit 1
P2	B	Parallel Poll Response bit 2
P3	B	Parallel Poll Response bit 3
PACS	ST	Parallel Poll Addressed To Configure state
PCG	RM	Primary Command Group
PE		Pull-Up Enable
PEND	B	Pending bit
piaccr	A	Page-In Accessory Register auxiliary command
pibcr	A	Page-In Bus Control Register auxiliary command
pieosr	A	Page-In End-of-String Register auxiliary command
piimr2	A	Page-In Interrupt Mask Register 2 auxiliary command
pon	LM	Power On
PP1	B	Parallel Poll bit 1
PP2	B	Parallel Poll bit 2
PPC	RM	Parallel Poll Configure
PPD	RM	Parallel Poll Disable
PPE	RM	Parallel Poll Enable
PPR	R	Parallel Poll Register
PPR	RM	Parallel Poll Response
PPU	RM	Parallel Poll Unconfigure
pts	A	Pass Through Next Secondary auxiliary command

R

rdy	LM	Ready For Next Message
REM	B	Remote bit
REMC	B	Remote Change bit
REMC IE	B	Remote Change Interrupt Enable bit
REN	RM	Remote Enable
REOS	B	End On EOS Received bit
reqf	A	Request rsv False auxiliary command
reqt	A	Request rsv True auxiliary command
RFD	RM	Ready For Data

Mnemonic	Type	Definition
rhdf	B	Release RFD Holdoff
rhdf	A	Release RFD Holdoff auxiliary command
RL1	F	Remote/Local
rlc	B	Release Control command
RLC	B	Remote/Local Change bit
RLC IE	B	Remote/Local Change Interrupt Enable bit
rpp	A	Clear/Set Request Parallel Poll
rpp1	A	Execute Parallel Poll auxiliary command
RPP2	B	Request Parallel Poll 2 bit
rqc	B	Request Control command
RQS	RM	Request Service
rsv	LM	Request Service
rsv	B	Request Service bit
rsv2	A	Request Service Bit 2 auxiliary command
rtl	A	Return To Local auxiliary command

S

S	B	Status Bit Polarity (Sense) bit
S[6–1]	B	Serial Poll Status bits 6 through 1
SASR	R	Source Acceptor Status Register
SC	P	75162 System Controller pin
SCG	RM	Secondary Command Group
SDC	RM	Selected Device Clear
SDHS		Selected DAC Holdoffs Internal Signal
SDYS	ST	Source Delay State
seoi	A	Send EOI auxiliary command
SH	F	Source Handshake function
SH1A	B	Source Handshake State bit A
SH1B	B	Source Handshake State bit B
SHAS	ST	Source High-Speed Active State
shdw	A	Clear/Set Shadow Handshake auxiliary command
sic	A	Clear/Set Send Interface Clear auxiliary command
SIDS	ST	Source Idle State
SISB	B	Static Interrupt Status bit
SLOW	B	Slow Handshake Lines
SPAS	ST	Serial Poll Active State
SPAS IE	B	Serial Poll Active State Interrupt Enable bit
SPD	RM	Serial Poll Disable
SPE	RM	Serial Poll Enable
SPEOI	B	Send Serial Poll EOI bit
SPIS	ST	Serial Poll Idle State
SPMR	R	Serial Poll Mode Register
SPMS	ST	Serial Poll Mode State
SPMS	B	Serial Poll Mode State bit
SPSR	R	Serial Poll Status Register

Mnemonic	Type	Definition
SR1	F	Service Request function
sre	A	Clear/Set Send Remote Enable auxiliary command
SRQ	RM	Service Request
SRQ IE	B	Service Request Interrupt Enable bit
SRQI	B	Service Request bit
SRQI IE	B	Service Request Interrupt Enable bit
SRQS	ST	Service Request State
STB	RM	Status Byte
STBO	B	Status Byte Out bit
STBO IE	B	Status Byte Out Interrupt Enable bit
stdl	A	Set Short T1 Delay auxiliary command
STRS	ST	Source Transfer State
sw7210	A	Switch To Turbo+7210 Mode auxiliary command
swrst	A	Software Reset auxiliary command issued
SYNC	B	GPIB Synchronization bit
SYNC IE	B	GPIB Synchronization Interrupt Enable bit

T

T	F	Talker
TA	B	Talker Active bit
TACS	ST	Talker Active State (T function)
TADS	ST	Talker Addressed State
tca	A	Take Control Asynchronously auxiliary command
tcs	A	Take Control Synchronously auxiliary command
tcse	A	Take Control Synchronously On End auxiliary command
TCT	RM	Take Control
TE	F	Extended Talker
TIDS	ST	Talker Idle State
ton	LM	Talk Only
ton	B	Talk-Only bit
TPAS	ST	Talker Primary Addressed State
TPAS	B	Talker Primary Addressed State bit
TPIS	ST	Talker Primary Idle State
TRI	B	Three-State Timing bit
TRIG		Trigger
trig	A	Trigger auxiliary command
TRM0	B	Transmit/Receive Mode bit 0
TRM1	B	Transmit/Receive Mode bit 1

Mnemonic	Type	Definition
U		
U	B	Unconfigure bit
UCG	RM	Universal Command Group
ulpa	B	Upper/Lower Primary Address bit
UNC	B	Unrecognized Command bit
UNC IE	B	Unrecognized Command Interrupt Enable bit
unl	A	Unlisten auxiliary command
UNL	RM	Unlisten command
unt	A	Untalk auxiliary command
UNT	RM	Untalk command
USTD	B	Ultra Short T1 Delay bit
V		
valid	A	Valid auxiliary command
VSR	R	Version Status Register
vstdl	A	Very Short T1 Delay auxiliary command
X		
X	B	Don’t care bit
XEOS	B	Transmit END With EOS bit

Appendix F
Customer Communication

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve technical problems you might have as well as a form you can use to comment on the product documentation. Filling out a copy of the *Technical Support Form* before contacting National Instruments helps us help you better and faster.

National Instruments provides comprehensive technical assistance around the world. In the U.S. and Canada, applications engineers are available Monday through Friday from 8:00 a.m. to 6:00 p.m. (central time). In other countries, contact the nearest branch office. You may fax questions to us at any time.

Corporate Headquarters
(512) 795-8248
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices	**Phone Number**	**Fax Number**
Australia	03 9 879 9179	03 9 879 9422
Austria	0662 45 79 90 19	0662 45 79 90 0
Belgium	02 757 03 11	02 757 00 20
Denmark	45 76 71 11	45 76 26 00
Finland	90 502 2930	90 527 2321
France	1 48 14 24 14	1 48 14 24 24
Germany	089 714 60 35	089 741 31 30
Hong Kong	2686 8505	2645 3186
Italy	02 48301915	02 48301892
Japan	03 5472 2977	03 5472 2970
Korea	02 596 7455	02 596 7456
Mexico	5 202 2544	5 520 3282
Netherlands	03480 30673	03480 33466
Norway	32 84 86 00	32 84 84 00
Singapore	2265887	2265886
Spain	91 640 0533	91 640 0085
Sweden	08 730 43 70	08 730 49 70
Switzerland	056 20 51 55	056 20 51 51
Taiwan	02 737 4644	02 377 1200
U.K.	01635 523154	01635 523545

Technical Support Form

Technical support is available at any time by fax. Include the information from your configuration form. Use additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax (___)_____ Phone (___)_____

Computer brand _____

Model _____ Processor _____

Operating system _____

Speed _____MHz RAM _____MB

Display adapter _____

Mouse _____ yes _____ no

Other adapters installed _____

Hard disk capacity _____ MB Brand _____

Instruments used _____

National Instruments hardware product(s) _____

Revision _____

Configuration _____

(continues)

The problem is _____

List any error messages _____

The following steps will reproduce the problem _____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: **NAT7210™ Reference Manual**

Edition Date: **June 1995**

Part Number: **370875A-01**

Please comment on the completeness, clarity, and organization of the manual.

(continues)

If you find errors in the manual, please record the page numbers and describe the errors.

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

_____

Phone (_____) _____

Mail to: Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway, MS 53-02
Austin, TX 78730-5039

Fax to: Technical Publications
National Instruments Corporation
MS 53-02
(512) 794-5678

Glossary

Prefix	Meaning	Value
p-	pico-	10^{-12}
n-	nano-	10^{-9}
µ-	micro-	10^{-6}
m-	milli-	10^{-3}
M-	mega-	10^{6}

AC	alternating current
ANSI	American National Standards Institute
CIC	Controller-In-Charge
CPU	central processing unit
DIP	dual inline package
DMA	direct memory access
EOI	End-or-Identify
EOS	End-of-String
F	Farads
GPIB	General Purpose Interface Bus
hex	hexadecimal
Hz	hertz
IC	integrated circuit
I/O	input/output
ISO	International Standards Organization
m	meters
MB	megabytes of memory
MHz	megahertz
s	seconds

Index

Numbers

A

B

C

D

E

F

G

H

I

L

M

N

P

R

S

T

U

V

W

X