CB7210.2

General Purpose Interface Bus

User's Guide

MEASUREMENT COMPUTING™

Document Revision 3.0, July, 2006

Your new Measurement Computing product comes with a fantastic extra —

Management committed to your satisfaction!

Refer to www.mccdaq.com/execteam.html for the names, titles, and contact information of each key executive at Measurement Computing.

Thank you for choosing a Measurement Computing product—and congratulations! You own the finest, and you can now enjoy the protection of the most comprehensive warranties and unmatched phone tech support. It's the embodiment of our two missions:

- To offer the highest-quality, computer-based data acquisition, control, and GPIB hardware and software available—at the best possible price.
- To offer our customers superior post-sale support—FREE. Whether providing unrivaled telephone technical and sales support on our latest product offerings, or continuing that same first-rate support on older products and operating systems, we're committed to you!

Lifetime warranty: Every hardware product manufactured by Measurement Computing Corporation is warranted against defects in materials or workmanship for the life of the product. Products found defective are repaired or replaced promptly.

Lifetime Harsh Environment Warranty®: We will replace any product manufactured by Measurement Computing Corporation that is damaged (even due to misuse) for only 50% of the current list price. I/O boards face some tough operating conditions—some more severe than the boards are designed to withstand. When a board becomes damaged, just return the unit with an order for its replacement at only 50% of the current list price. We don't need to profit from your misfortune. By the way, we honor this warranty for any manufacturer's board that we have a replacement for.

30 Day Money Back Guarantee: You may return any Measurement Computing Corporation product within 30 days of purchase for a full refund of the price paid for the product being returned. If you are not satisfied, or chose the wrong product by mistake, you do not have to keep it. Please call for an RMA number first. No credits or returns accepted without a copy of the original invoice. Some software products are subject to a repackaging fee.

These warranties are in lieu of all other warranties, expressed or implied, including any implied warranty of merchantability or fitness for a particular application. The remedies provided herein are the buyer's sole and exclusive remedies. Neither Measurement Computing Corporation, nor its employees shall be liable for any direct or indirect, special, incidental or consequential damage arising from the use of its products, even if Measurement Computing Corporation has been notified in advance of the possibility of such damages.

Notice

Measurement Computing Corporation does not authorize any Measurement Computing Corporation product for use in life support systems and/or devices without prior written consent from Measurement Computing Corporation. Life support devices/systems are devices or systems which, a) are intended for surgical implantation into the body, or b) support or sustain life and whose failure to perform can be reasonably expected to result in injury. Measurement Computing Corporation products are not designed with the components required, and are not subject to the testing required to ensure a level of reliability suitable for the treatment and diagnosis of people.

Table of Contents

Overview: GPIB features

The IEEE Standard 488-1978 defines a special bus structure known as the General Purpose Interface Bus, or GPIB. For the remainder of this document the IEEE Standard 488-1978 is written as IEEE Standard unless otherwise noted.

The IEEE Standard allows you to connect programmable instruments that are produced by different manufacturers to a computer using one interface — the GPIB. The GPIB simplifies system configuration and reduces hardware complexity. You can connect any device or instrument that meets the IEEE Standard specification to the GPIB without interface circuitry (see Figure 1-1).

The IEEE Standard defines three device types — Talker, Listener, and Controller. A GPIB device can be a Talker, Listener, and/or Controller.

- A **Talker** sends data to one or more Listeners. Only one device can transmit data at a time.
- A **Listener** accepts data from a Talker. Up to 14 active listeners can be present on the bus at the same time.
- A **Controller** manages the flow of information over the bus. The controller makes sure that only one device tries to talk at a time, and to make sure the correct Listener(s) are receiving data when the Talker talks. Each system controller is ultimately in charge of the bus, and is in control when the bus is powered up.

Some devices can be a combination of these functions, such as talker/listener and talker/controller. Only one of these functions can be performed at a time. For example, a GPIB digital voltmeter acts as a Listener when its input configurations and ranges are being set, but then acts as a Talker when sending its readings to the computer.

The GPIB transfers digital data among up to 15 devices over a common set of data control lines. Communications among the devices is through the signals and protocols specified by the IEEE standard.

Data is transferred in a bit-parallel, byte-serial form over eight data I/O lines (DIO_1 to DIO_8). A three-wire handshake ensures synchronization of data transmission and reception. The data control lines are an "open collector" type, so that multiple devices can receive data at the same time. Data is transferred at the rate of the slowest device. Other control lines perform additional functions, such as device addressing and interrupt generation.

The CB7210.2 performs all of the interface functions that are defined in the IEEE 488-1978 specification. The CB7210.2 GPIB interface chip performs IEEE 488 Talker, Listener, and Controller (**TLC**) functions. You control the CB7210.2 with its 16 data registers — eight read registers and eight write registers.

In addition to managing and controlling the GPIB, the CB7210.2 provides a unique set of SN75160-based and MC3448A-based bus transceiver controls. These controls enable multiple transceiver configurations.

IEEE Standard 488-1978 characteristics

The IEEE Standard defines the following characteristics:

Electrical: Logical and electrical signal levels
Receiver and driver requirements
Loading, conductor, and ground requirements

Mechanical: Connector type
Connector contact assignments
Device connector mounting
Cable assembly

Functional: Signal lines
Interface functions
Local and remote messages
Protocol and timing relationships

Figure 1-1 illustrates the architecture of the CB7210.2 bus configuration and interface functions.

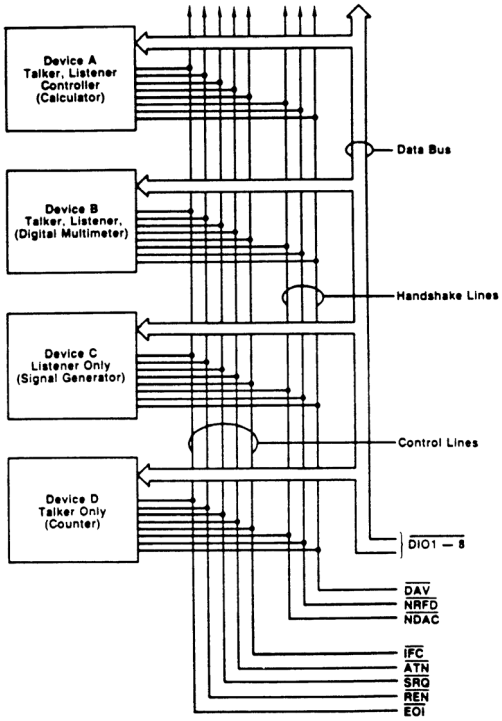


Figure 1-1. Bus configuration and interface functions

The IEEE Standard applies to:

- Systems that connect programmable and non-programmable electronic measurement devices with other instruments
- The accessories needed to develop instrumentation systems

In order for the IEEE Standard to be applicable to an interface system, the following must be true:

- the data exchanged among the instruments is digital
- the total length of the connecting cable is 20 meters or less, with no more than two meters per device
- the data rate across the interface on any signal line is 1 Mb/sec or less
- no more than 15 devices connected to one bus

Different electrical and mechanical specifications may be required for applications that do not conform to the standard.

GPIB signal configuration

The GPIB is an 8-bit parallel data transfer bus. The GPIB connector is a 24-pin ribbon cable connector. In addition to the 8-data bits, the bus carries three handshaking lines and five management and control lines. The remaining pins are used for cable shield, signal grounds, and returns. Signal names are shown in Figure 1-2.

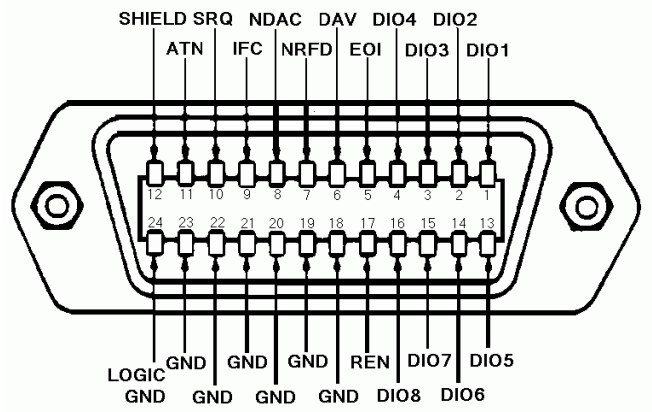


Figure 1-2. Signals on the GPIB connector

Data lines

Data lines (DIO_1 to DIO_8) transfer interface messages and device-dependent messages in asynchronous, bidirectional, bit-parallel, byte-serial form.

Handshake lines

Handshake lines transfer each byte of data on the DIO lines from talker or controller to one or more listeners. The GPIB has three handshaking lines:

- **DAV** (data valid) indicates that data on the DIO signal lines is valid.
- **NRFD** (ready for data) indicates that one or more devices are ready to accept data.
- **NDAC** (data accepted) indicates that one or more devices has accepted the data.

The three handshake signals coordinate the transfer of each data byte over the GPIB between sources and receivers. This ensures that the data is valid and that the transfer is complete.

The following sequence of events occurs during a data transfer:

1. A receiver sets the ready for data pin (NRFD/) passively high. This action indicates that the receiver is ready for data.
2. A source pulls the data valid pin (DAV/) low to indicate that the data is valid.
3. When valid data is received, all receivers set the data accepted pin (NDAC) passively high. This indicates that the data is accepted.

Bus management lines

Management, or control, lines maintain an orderly flow of data across the interface. The GPIB has five management, or control, lines:

- **ATN** (attention) indicates whether data on the DIO lines is an interface message or a device-dependent message, and which devices should respond.
- **IFC** (interface clear) places the interface system in a known quiescent state.
- **SRQ** (service request) is used to request service from the controller.

- **REN** (remote enable) selects between remote and local operation of the devices.
- **EOI** (END or identify) indicates the end of a multiple byte transfer sequence. When used with ATN, EOI executes a polling sequence.

GPIB interface functions

The CB7210.2 performs the 10 GPIB functions as specified by the IEEE Standard. You can select any of the functions to implement in a device, depending on the intended application. The GPIB interface functions are listed in Table 1-1.

Table 1-1. Interface functions

GPIB Interface Function	Function code	Description
Source Handshake	SH (SH1)	Complete source handshake capability. Allows a device to transfer data from a talker to one or more listeners using the three handshake lines.
Acceptor Handshake	AH (AH1)	Allows a device to receive data from the talker using the three handshake lines.
Talker	T (T5)	Allows a device to send status and data bytes when addressed to talk. An address consists of one or two bytes. Two bytes is sent by an extended talker. Talker capability: ▪ Basic talker ▪ Serial poll ▪ Talk only mode ▪ Unaddressed on `MLA` (My Listen Address) ▪ Send END or EOS (End of String) ▪ Dual primary addressing
Talker Extended	TE (TE5)	Extended talker capability: ▪ Basic extended talker ▪ Serial poll ▪ Talk only mode ▪ Unaddressed on `MSA` (My Secondary Address) * `LPAS` (Listener Primary Addressed state) ▪ Send END or EOS (End of String) ▪ Dual extended addressing with software assist
Listener	L (L3)	Allows a device to receive data when addressed to listen. Receipt of two bytes indicates an extended listener. Listener capability: ▪ Basic listener ▪ Listen only mode ▪ Unaddressed on `MTA` (My Talk Address) ▪ Detect END or EOS (End of String) ▪ Dual primary addressing
Listener Extended	LE (LE3)	Extended listener capability: ▪ Basic extended listener ▪ Listen only mode ▪ Unaddressed on `MSA` (My Secondary Address) * TPAS (Talker Primary Addressed state) ▪ Detect END or EOS (End of String) ▪ Dual extended addressing with software assist
Service Request	SR (SR1)	Complete service request capability. Allows a device to request service from (interrupt) the controller. The SRQ line may be asserted asynchronously.
Remote/Local	RL	Complete remote/local capability with software interpretation. Allows a device to be operated either remotely via the GPIB or local via the front panel controls.

GPIB Interface Function	Function code	Description
Remote Parallel Poll	PP (PP1)	Remote parallel poll configuration. Allows a device to present one bit of status to the Controller-in-Charge. The device does not need to be addressed to talk, and no handshake is required. Controller-initiated.
Local Parallel Poll	PP (PP2)	Local parallel poll configuration. Requires software assistance. Controller-initiated.
Serial Poll	SP	Polls each talker device to determine which device requested service. Device-initiated.
Device Clear	DC (DC1)	Device clear capability requires software assistance. Allows the controller to clear a device. Note that there is a difference between DC and the IFC bus management line.
Device Trigger	DT (DT1)	Device trigger capability requires software assistance. Allows a device to have its operation started individually or as part of a group. Device trigger is often used to synchronize several instruments.
Controller	C (C_1 to C_5)	Allows a device to send an address, a universal command, or an addressed universal command to other devices. There may be more than one controller on a system, but only one may be the controller-in-charge at any time. Device clear capability: ▪ System controller ▪ Send IFC (Interface Clear) and take charge ▪ Send REN (Remote Enable) ▪ Respond to SRQ (Service request) ▪ Send interface messages ▪ Received control ▪ Parallel poll ▪ Take control synchronously and asynchronously

Additional information on each interface function is listed in Chapter 4 "TLC Interface Functions".

GPIB signal levels

All of the electrical specifications for the driver and receiver are TTL-compatible. The IEEE Standard uses negative logic. The logic level is related to the signal level:

Logic Level	Signal level
0 = *False*	≤ 2.0 V High state
1 = *True*	≥ 0.8 V Low state

Drivers

There are two types of drivers — **open collector** and **three state**.

- Use open collector drivers when you want multiple devices to receive data at the same time.
- Use three state drivers in applications that require data rates above 250,000. bytes/sec.

Open collector only drivers	Open collector or three-state drivers
SRQ	ATN
NRFD	IFC
NDAC	REN
DIO_1-DIO_8 (parallel poll devices)	EOI
	DAV
	DIO_1-DIO_8 (non parallel poll devices)

Driver specifications

- $V_{OL} < 0.5$ V @ 48 mA continuous sink (three state or open collector)
- $V_{OH} \geq 2.4$ V @ 5.2 mA source (three state)

Receiver specifications

- $V_{IL} \leq 0.8$ V
- $V_{IH} \geq 2.0$ V

Use Schmitt-type receiver circuits for all signal lines to provide additional immunity from noise. The Schmitt-type receiver specifications are as follows:

- Hysteresis: V_{tpos}-$V_{tneg} \geq 0.4$ V
- Low state: Negative-going threshold voltage $V_{tneg} \geq 0.8$ V
- High state: Positive-going threshold voltage $V_{tpos} \leq 2.0$ V

Refer to the IEEE Standard for device load requirements.

Signal time relationships

Table 1-2 shows the mandatory time relationships between critical signal inputs and outputs to a specific device. These values ensure maximum compatibility. Message coding converts remote messages to or from interface signal line values. Two types of messages are used in message coding processes — uni-line and multi-line.

- A uni-line message is sent over a single line.
- A multi-line message is sent over a group of lines.

In Table 1-2, time values specified by a lower case "t" indicate the maximum time allowed to make a state transition. Time values specified by an upper case "T" indicate the minimum time that a function remains in a state before exiting.

Table 1-2. Interface state time values

Time Value Identifier	Applicable Interface Function	Description	Value
T_1	SH	Settling time for multi-line messages	≥ 2 µs (see note 1)
t_2	SH, AH, T, L, LE, TE	Response to ATN	≤ 200 ns
T_3	AH	Interface message accept time ± (see note 2)	> 0 (see note 3)
t_4	T, TE, L, LE, C, R/L	Response to IFC or REN failure	< 100 µs
t_5	PP	Response to ATN ^ EOI	≤ 200 ns
T_6	C	Parallel poll execution time	≥ 2 µs
T_7	C	Controller delay to allow current talker to see ATN message	≥ 500 ns
T_8	C	Length of IFC or REN *false*	> 100 µs
T_9	C	Delay for EOI (see note 4)	≥ 1.5 µs (see note 5)

Note 1: If three-state drivers are used on the DIO, DAV, and EOI lines, the value of T_1 can be one of the following values:

1. ≥ 1100 ns
2. ≥ 700 ns, if it is known that within the controller ATN is driven by a three-state driver (not recommended)
3. ≥ 500 ns for all subsequent bytes following the first sent after each false transition of ATN. The first byte shall be sent in accordance with the two values above.
4. ≥ 350 ns for all subsequent bytes following the first sent after each false transition of ATN under specified conditions.

Note 2: Time required for interface functions to accept but not necessarily respond to interface messages.

Note 3: Implementation dependent.

Note 4: Delay required for EOI, NDAC, and NRFD signal lines to indicate valid states.

Note 5: ≥ 600 ns for three-state drivers.

Remote messages

Only one multi-line message can be sent at a time. Several uni-line messages may be sent simultaneously on different lines. A remote message received via the interface is represented by a three-letter mnemonic written in uppercase. Table 1-3 lists the remote messages and associated mnemonic codes.

Table 1-3. Remote messages and mnemonic codes

Mnemonic	Message Name	Bus Signal Line(s) and Coding that assert message true																Msg type	Msg class	Notes
		Data I/O								Handshake			Bus Management							
		8	7	6	5	4	3	2	1	DAV	NRFD	NDAC	ATN	EOI	SRQ	IFC	REN			
ACG	Addressed Command Group	X	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	M	AC	10
ATN	Attention	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	U	UC	–
DAB	Data Byte	D8	D7	D6	D5	D4	D3	D2	D1	X	X	X	X	X	X	X	X	M	DD	1, 9
DAC	Data Accepted	X	X	X	X	X	X	X	X	X	X	0	X	X	X	X	X	U	HS	–
DAV	Data Valid	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	U	HS	–
DCL	Device Clear	X	0	0	1	0	1	0	0	X	X	X	X	X	X	X	X	M	UC	10
END	End	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	U	ST	9, 11
EOS	End Of String	E8	E7	E6	E5	E4	E3	E2	E1	X	X	X	X	X	X	X	X	M	DD	2, 9
GET	Group Execute Trigger	X	0	0	0	1	0	0	0	X	X	X	X	X	X	X	X	M	AC	10
GTL	Go To Local	X	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X	M	AC	10
IDY	Identify	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	U	UC	10,11
IFC	Interface Clear	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	U	UC	–
LAG	Listen Address Group	X	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	M	AD	10
LLO	Local Lock Out	X	0	0	1	0	0	0	X	X	X	X	X	X	X	X	X	M	UC	10
MLA	My Listen Address	X	0	1	L5	L4	L3	L2	L1	X	X	X	X	X	X	X	X	M	AD	3, 10
MTA	My Talk Address	X	1	0	T5	T4	T3	T2	T1	X	X	X	X	X	X	X	X	M	AD	4, 10
MSA	My Secondary Address	X	1	1	S5	S4	S3	S2	S1	X	X	X	X	X	X	X	X	M	SE	5, 10
NUL	Null Byte	1	1	0	0	0	0	0	0	X	X	X	X	X	X	X	X	M	DD	–
OSA	Other Secondary Address	OSA = SCG ∧ MSA																M	SE	10
OTA	Other Talk Address	OTA = TAG ∧ MTA																M	AD	10
PCG	Primary Command Group	PCG = ACG v UCG v LAG v TAG																M	–	10
PPC	Parallel Poll Configure	X	0	0	0	0	1	0	1	X	X	X	X	X	X	X	X	M	AC	10
PPE	Parallel Poll Enable	X	1	1	0	S	P_3	P_2	P_1	X	X	X	X	X	X	X	X	M	SE	6, 10
PPD	Parallel Poll Disable	X	1	1	1	D_4	D_3	D_2	D_1	X	X	X	X	X	X	X	X	M	SE	7, 10
PPR1	Parallel Poll Response 1	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	U	ST	–
PPR2	Parallel Poll Response 2	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	U	ST	–
PPR3	Parallel Poll Response 3	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	X	U	ST	–
PPR4	Parallel Poll Response 4	X	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	U	ST	–
PPR5	Parallel Poll Response 5	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	–
PPR6	Parallel Poll Response 6	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	–
PPR7	Parallel Poll Response 7	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	–
PPR8	Parallel Poll Response 8	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	–
PPU	Parallel Poll Unconfigure	X	0	0	1	0	1	0	1	X	X	X	X	X	X	X	X	M	UC	10
REN	Remote Enable	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	U	UC	–
RFD	Ready for Data	X	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	U	HS	–
RQS	Request Service	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	U	ST	9
SCG	Secondary Command Group	X	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	M	SE	10
SDC	Specified Device Clear	X	0	0	0	0	1	0	0	X	X	X	X	X	X	X	X	M	AC	10

Mnemonic	Message Name	Bus Signal Line(s) and Coding that assert message true																Msg type	Msg class	Notes
		Data I/O								Handshake			Bus Management							
		8	7	6	5	4	3	2	1	DAV	NRFD	NDAC	ATN	EOI	SRQ	IFC	REN			
SPD	Serial Poll Disable	X	0	0	1	1	0	0	1	X	X	X	X	X	X	X	X	M	UC	10
SPE	Serial Poll Enable	X	0	0	1	1	0	0	0	X	X	X	X	X	X	X	X	M	UC	10
SRQ	Service Request	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	U	ST	–
STB	Status Byte	S8	X	S6	S5	S4	S3	S2	S1	X	X	X	X	X	X	X	X	M	ST	8, 9
TCT	Take Control	X	0	0	1	0	0	1	X	X	X	X	X	X	X	X	X	M	AC	10
TAG	Talk Address Group	X	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	M	AD	10
UCG	Universal Command Group	X	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	M	UC	10
UNL	Unilisten	X	0	1	1	1	1	1	1	X	X	X	X	X	X	X	X	M	AD	10
UNT	Untalk	X	1	0	1	1	1	1	1	X	X	X	X	X	X	X	X	M	AD	10

The following information defines the message types, classes, and notes referenced above in Table 1-3.

Message Type	
U	uni-line message
M	multi-line message
Message Class	
AC	Addressed Command
AD	Address (talk or listen)
DD	Device-dependent
HS	Handshake
UC	Universal Command
SE	Secondary
ST	Status
0	Logical zero (HIGH signal level)
1	Logical one (LOW signal level)
x	don't care (for the coding of a received message)
x	Must not drive (for the coding of a transmitted message)
Notes:	
1	D_1-D_8 specify the device-dependent data bits (D_0 to D_7 on the TQFP chip.)
2	E1-E8 specify the device-dependent code used to initiate the EOS message.
3	L1-L5 specify the device-dependent bits of the device listen address.
4	T1-T5 specify the device-dependent bits of the device talk address.
5	S1-S5 specify the device-dependent bits of the device secondary address.
6	S specifies the sense of the PPR (Parallel Poll Response): **S** **Response** 0 1 1 1
	P_1-P_3 specify the PPR message to be sent when a particular poll is executed **P_3** **P_2** **P_1** **PPR Message** 0 0 0 PPR1 1 1 1 PPR8
7	D_1-D_4 specify don't care bits that must be sent as all zeroes, but do not need to be decoded by the receiving device.
8	S1-S6 and S8 specify the device dependent status (DIO_7 is used for the SRQ message).
9	The *true* message value must be ignored when received if the LACS message is inactive.
10	The *true* message value must be ignored when received if the ATN message is *false*.
11	Interface protocol specifies that the IDV message is sent *true*, whereas the End message is sent only when the ATN message is sent *false*.

Local messages

Local messages flow between device functions and interface functions. Each data byte that is transferred by the interface system uses the handshake process to exchange data between source and receiver. The *source* is typically a talker, and the *receiver* is a listener. A local message to an interface function is represented by a three-letter mnemonic written in lowercase. Local message mnemonics are listed in Table 1-4.

Table 1-4. Local message mnemonics

Local message mnemonic	Local message title
gts	go to standby
isr	individual service request (qual)
ist	individual status
lon	listen only
[lpe]	local poll enable
ltn	listen
lun	local unisten
nba	nev byte available
pon	power on
rdy	ready
rpp	request parallel poll
rsc	request system control
rsv	request service
rtl	return to local
sic	send interface clear
sre	send remote enable
tca	take control asynchronously
tcs	take control synchronously
ton	talk only

Figure 1-3 illustrates the waveforms on the DAV, NRFD, and NDAC signal lines during the handshake process.

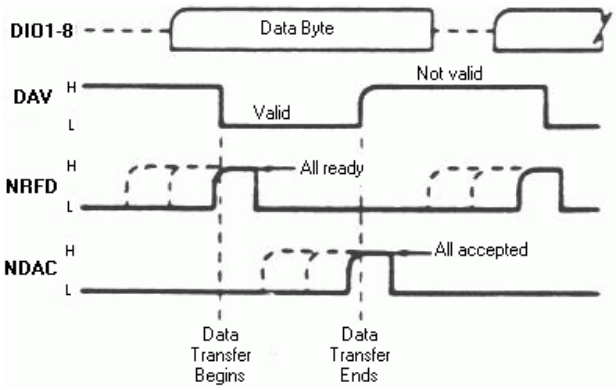


Figure 1-3. Data byte transfer waveforms

Figure 1-4 shows the same sequence of events in flowchart form. In Figure 1-4, the logical flow of events is illustrated for Source and Acceptor operations when transferring data using the handshake process.

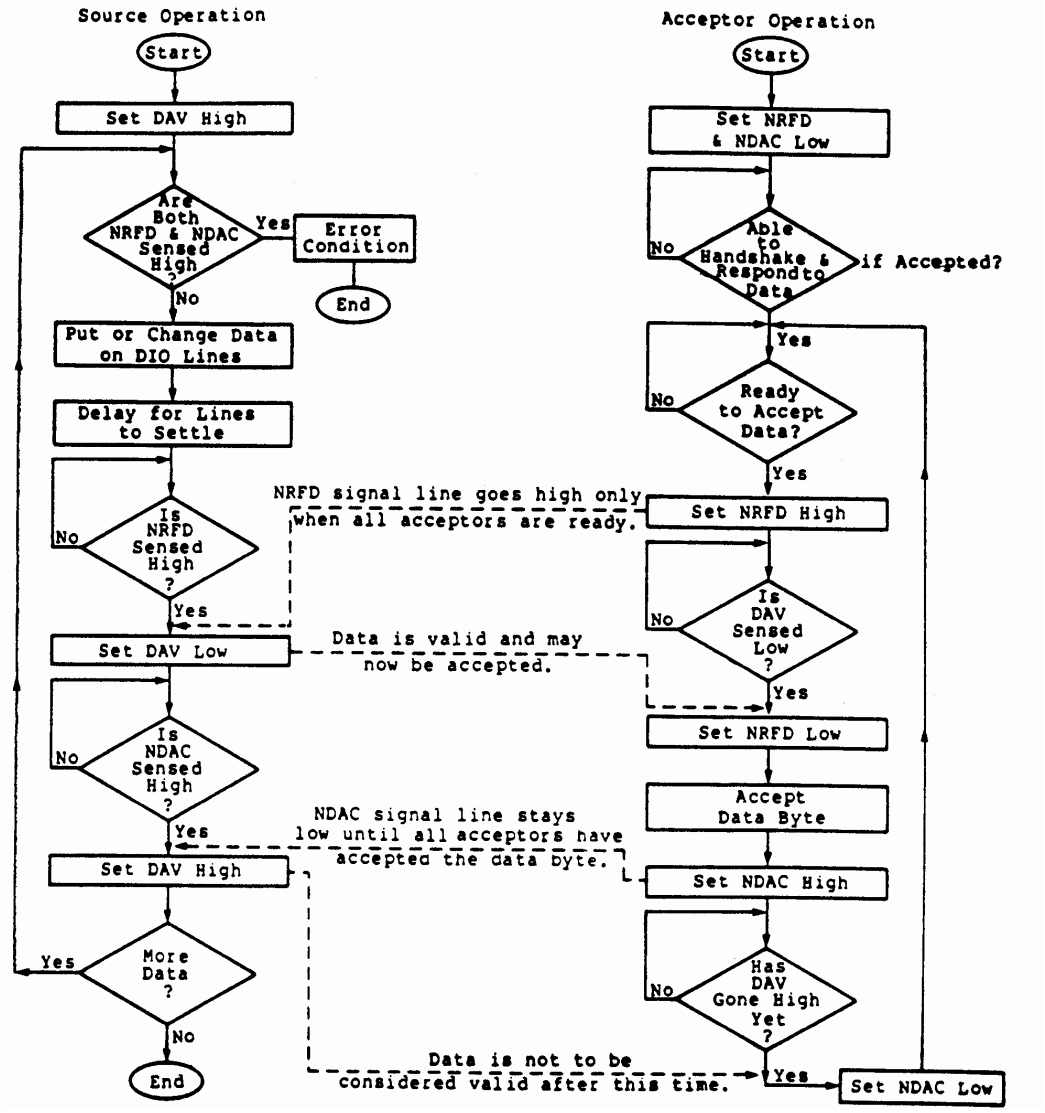


Figure 1-4. Handshake timing sequence

State diagram mnemonics

Each state that is assumed by an interface function is represented by a circle. A four-letter, uppercase mnemonic ending in **S** identifies the state, for example, CACS.

Table 1-5 lists the interface state assumed for each function, the mnemonic code for each interface state, and the state diagram(s) that the interface state is shown in. The state diagrams illustrate how each function is implemented. These diagrams are shown in Chapter 4, "TLC Interface Functions".

Table 1-5. State diagram mnemonics

Mnemonic State	Interface State	State Diagram
ACDS	Acceptor Data state	AH, L, LE, T, TE, C, DC, DT, R/L, PP
ACRS	Acceptor Ready state	AH
AIDS	Acceptor Idle state	AH
ANRS	Accept Not Ready state	AH, C
APRS	Affirmative Poll Response state	SR
AWNS	Acceptor Wait for New Cycle State	AH
CACS	Controller Active state	SH, L, LE, C
CADS	Controller Addressed state	C
CAWS	Controller Active Wait state	C
CIDS	Controller Idle state	C
CPPS	Controller Parallel Poll state	C
CPWS	Controller Parallel Poll Wait state	C
CSBS	Controller Standby state	C
CSNS	Controller Service Not Requested state	C
CSRS	Controller Service Requested state	C
CSWS	Controller Synchronous Wait state	C
CTRS	Controller Transfer state	SH, C
DCAS	Device Clear Active state	DC
DCIS	Device Clear Idle state	DC
DTAS	Device Trigger Active state	DT
DTIS	Device Trigger Idle state	DT
LACS	Listener Active state	AH, L, LE
LADS	Listener Addressed state	AH, L, LE, DC, DT, R/L, PP
LIDS	Listener Idle state	L, LE
LOCS	Local state	R/L
LPAS	Listener Primary Addressed state	LE, TE
LPIS	Listener Primary Idle state	LE
LWLS	Local with Lockout state	R/L
NPRS	Negative Poll Response state	SR
PACS	Parallel Poll Addressed to Configure state	PP
PPAS	Parallel Poll Active state	PP
PPIS	Parallel Poll Idle state	PP
PPSS	Parallel Poll Standby state	PP
PUCS	Parallel Poll Unaddressed to Configure state	PP
REMS	Remote state	R/L
RWLS	Remote with Lockout state	R/L
SACS	System Control Active state	C

Mnemonic State	Interface State	State Diagram
SDYS	Source Delay state	SH, C
SIAS	System Control Interface Clear Active state	C
SIDS	Source Idle state	SH
SIIS	System Control Interface Clear Idle state	C
SINS	System Control Interface Clear Not Active state	C
SIWS	Source Idle Wait state	SH
SNAS	System Control Not Active state	C
SPAS	Serial Poll Active state	SH, T, TE, SR
SPIS	Serial Poll Idle state	T, TE
SPMS	Serial Poll Mode state	T, TE
SRAS	System Control Remote Enable Active state	C
SRIS	System Control Remote Enable Idle state	C
SRNS	System Control Remote Enable Not Active state	C
SRQS	Service Request state	SR
STRS	Source Transfer state	SH, C
SWNS	Source Wait for New Cycle state	SH
TACS	Talker Active state	SH, T, TE
TADS	Talker Addressed state	T, TE, C
TIDS	Talker Idle state	T, TE
TPAS	Talker Primary Addressed state	LE, TE
TPIS	Talker Primary Idle state	TE

Overview: CB7210.2 TLC features

The CB7210.2 talker/listener/controller (TLC) is a GPIB interface controller that communicates with the bus specified in IEEE Standard 488-1978 and with a microcomputer system controlled by a microprocessor. The CB7210.2 connects to the GPIB via non-inverting bus transceivers. The transceivers meet the electrical specifications in the IEEE Standard.

Chip packages

The CB7210.2 is available as a 40-pin PDIP package, and also as a 44-pin TQFP package. The TQFP package can be used in PCMCIA cards or in OEM applications. Signal names for each chip package are shown in Figure 2-1.

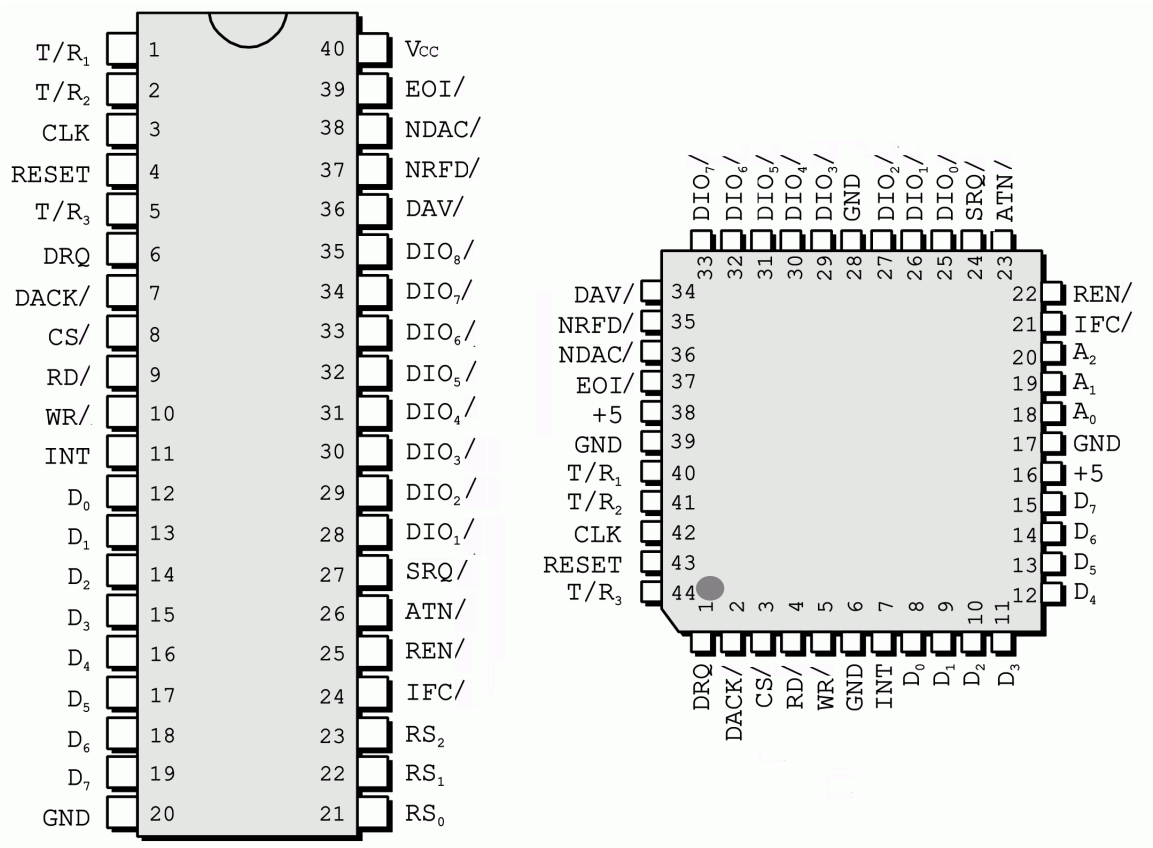


Figure 2-1. CB7210.2-PDIP and CB7210.2-TQFP pin diagrams

Functional block diagram

CB7210.2 functionality is shown in Figure 2-2.

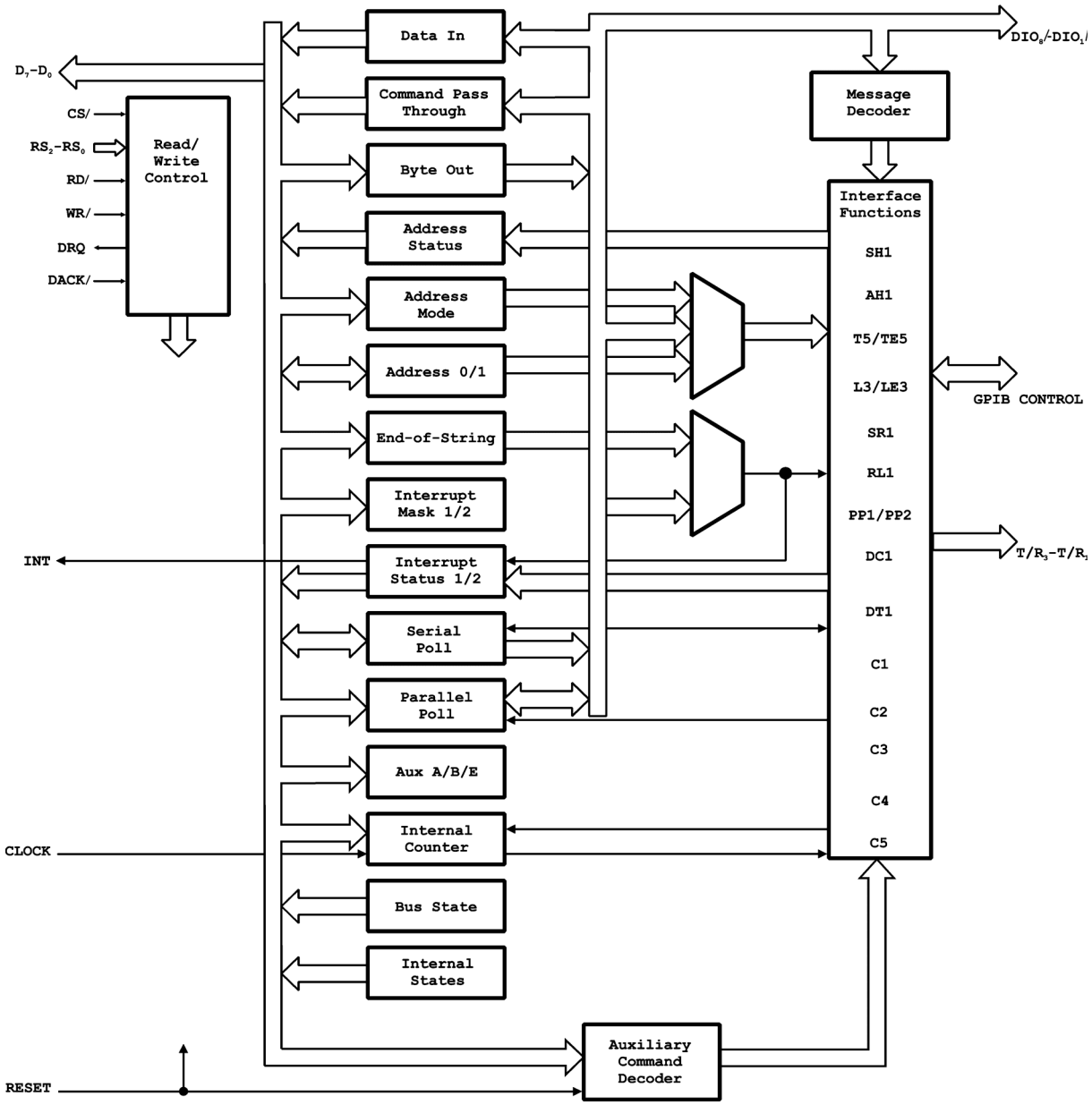


Figure 2-2. CB7210.2 block diagram

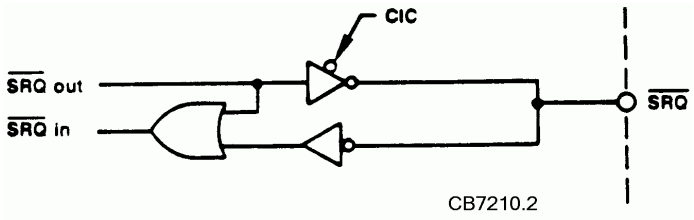
Functional description

The CB7210.2 TLC provides the following features:

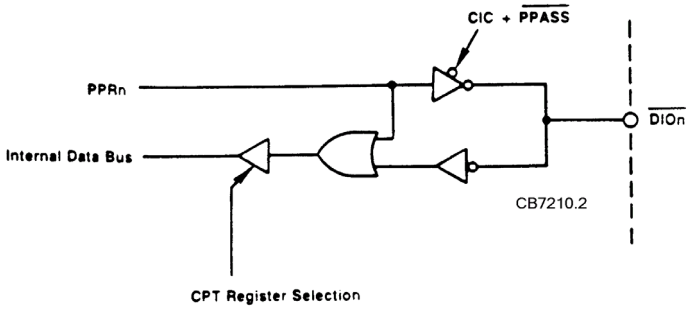
- Interface capability that meets the functional requirements of IEEE Standard 488-1978. Interface functions include:
  - Source Handshake (SH1), Acceptor Handshake (AH1), Talker/Talker Extended (T5/TES), Listener/Listener Extended, Listener/Listener Extended (L3/LE3), Service Request (SR1), Remote Local (RL1), Parallel Poll (PP1/PP2), Device Clear (DC1), Device Trigger (DT1), Controller (C_l, C_2, C_3, C_4, C_5)
- Programmable data transfer rate
- Eight read registers and eight write registers for message transmission and reception, interface function control, and status information
- Two address registers that detect the remote messages My Talk Address (MTA), My Listen Address (MLA), and My Secondary Address (MSA). The TLC has two device addresses.
- Automatic EOS message detection
- Automatic command processing and undefined command read capability
- DMA capability
- Programmable bus transceiver I/0 specification
- 1 MHz to 20 MHz clock range
- Ability to monitor all bus control lines
- Supports T_1 time delays of 2000 ns, 500 ns, and 350 ns
- 8080/8085/8086 compatible
- C MOS technology
- +5 V single power supply
- 40-pin plastic DIP package
- 44-pin TQFP package

Two special conditions exist when the controller is active — for example, when CIC = 1 (CIC = CIDS+CADS).

- When CIC·SRQS = 1, the SRQ pin becomes an input and the SRQ message is not transmitted. The SRQ message is detected inside the CB7210.2.



- When CIC·PPAS = 1, the DIO pins become inputs and the PPR message is not transmitted. Read the CPT register to detect the PPR message. The DIO pins are DIO_1 to DIO_8 on the PDIP chip and DIO_0 to DIO_7 on the TQFP chip.



Signal descriptions

In the following signal descriptions, a "/" suffix following a signal name indicates that the signal is active low.

T/R_1: Transmit/Receive Control 1 (output)

The T/R_1 signal controls the input/output direction of data flow in the GPIB bus transceivers SN75160, SN75161 and SN75l62. Connect T/R_1 to the Talk Enable (TE) pin of the bus transceivers.

When T/R_1 is low, the eight data lines of the SN75160 are in *receive* mode, and the data can be received from the GPIB and then transferred to the CB7210.2.

When T/R_1 is high, the data lines are in *transmit* mode, and the data is transmitted to the GPIB from the CB7210.2. With SN75161 or SN75l62, T/R_1 and T/R_2 implement the logic necessary to enable each signal in the correct direction for the exchange of bus management and handshaking processes.

The T/R_1 signal is pin 1 on the PDIP chip, and pin 40 on the TQFP chip.

T/R_2 and T/R_3: Transmit/Receive Controls 2 and 3 (output)

T/R_2 and T/R_3 are control lines to the GPIB bus transceivers. T/R_2 is connected to the Direction Control (DC) pin of SN75l6l by an inverter. T/R_3 is connected to the Pull-Up Enable (PE) pin of SN75l60.

The values of the Address Mode register bits TRM0 and TRM1 determine the functions of the T/R_2 and T/R_3 pins.

T/R_2	T/R_3	TRM0	TRM1
EOIOE	TRIG	0	0
CIC	TRIG	1	0
CIC	EOIOE	0	1
CIC	PE	1	1

EOIOE = TACS+SPAS + CIC·CSBS/
EOI pin output enable

EOIOE = l The EOI/ pin is an output.

EOIOE = 0 The EOI/ pin is an input.

CIC = (CIDS+CADS)/
Controller-in-charge

CIC = l The ATN/ pin is an output and SRQ/ is an input.

CIC = 0 ATN/ is an input and SRQ/ is an output.

PE = CIC·PPAS/
Pull-up enable for the DIO_1 to DIO_8 and the DAV lines

PE=l Three-state bus transceivers are used in DIO_1 to DIO_8 and the DAV lines.

PE=0 Open collector transceivers are used.

TRIG: Trigger

DTAS=1 A high pulse is generated that can simultaneously trigger multiple devices on the GPIB. This also occurs when a TRIGGER auxiliary command is issued.

On reset, TRM0 and TRM1 become 0 (TRM0=TRM1=0) so that TR_2 and T/R_3 both become low.

The T/R_2 signal is pin 2 on the PDIP chip, and pin 41 on the TQFP chip. The T/R_3 signal is pin 5 on the PDIP chip, and pin 44 on the TQFP chip.

CLK: Clock (input)

The CLK signal is used as a 1 MHz to 8 MHz reference clock for generating the state change minimum time delays (T_1, T_6, T_7, T_9), as specified in IEEE Standard 488-1978. The time delay values allow for the normal propagation delays of the transmission path and the circuit delays within the CB7210.2.

The CLK signal is pin 3 on the PDIP chip, and pin 42 on the TQFP chip.

RESET: Reset (input, active high)

The RESET signal resets the CB7210.2 to an idle state when high (active high).

You can reset the CB7210.2 using either hardware or software. To reset with hardware, apply a high pulse to the RESET pin. To reset with software, issue a CHIP RESET auxiliary command. When reset by either method, the CB7210.2 enters an idle state.

When you reset the CB7210.2, the following events occur:

- The interrupt mask and status registers are cleared
- All interface functions are placed in their idle states
- The auxiliary A, B, and E registers are cleared
- NF in the internal counter is set to 8 MHz

  This setting generates a delay of 16 µsec for 1 MHz clock in the Source Handshake (the longest possible tl delay)
- The Serial Poll Mode register is cleared
- The EOI bit in the Address 1 register is cleared
- The Interrupt Request pin (INT) goes high.

Only use one method when resetting the chip. If you simultaneously reset the CB7210.2 twice using hardware, software, or both, the CB7210.2 is no longer in the idle state.

The RESET signal is pin 4 on the PDIP chip, and pin 43 on the TQFP chip.

DRQ: DMA Request (output, active high)

The DMA Request signal requests data transfer.

DRQ goes high when you set the Dl (Data Input) or DO (Data Output) bits in Interrupt Status register 1. This indicates that a request for DMA occurred. The DMAI (DMA Input) and DMAO (DMA Output) bits in Interrupt Mask register 2 must be enabled before they can respond to the DI and DO bits

DRQ goes low when a DMA Acknowledge signal (DACK/) is input to the CB7210.2. You can connect DRQ to a DMA request channel of a DMA controller or to an interrupt pin of a CPU.

The DRQ signal is pin 6 on the PDIP chip, and pin 1 on the TQFP chip.

DACK/: DMA Acknowledge (input, active low)

The DACK/ signal connects the CPU data bus to the CB7210.2 chip's data register. This signal can come from the DMA Acknowledge of a DMA controller. In a DMA operation, it is a Chip Select to the CB7210.2.

When DACK/ is low, the contents of the Data In register are output into D_0-D_7 (microcomputer data bus) with a read signal. The data in D_0-D_7 is output to the Byte Out register with a write signal. *DACK/ must be pulled high when not in use.*

CS/	RD/	WR/	DACK/	RS0-RS2	Function
0	0	1	1	any	read registers 0R-7R
0	1	0	1	any	write to registers 0W-7W
X	0	1	0	XXX	read into Data In register
X	1	0	0	XXX	write to Byte Out register
1	X	X	1	XXX	register is not selected
X	1	1	X	XXX	(D_0-D_7 = Hi - z)
0	0	0	X	XXX	prohibit
X	0	0	0	XXX	(operation is not guaranteed)

The DACK/ signal is pin 7 on the PDIP chip, and pin 2 on the TQFP chip.

CS/: Chip Select (input, active low)

The Chip Select signal enables you to read and write to the data register specified by RS_0-RS_2. You can usually use logic external to the chip to decode this signal.

The CS/ signal is pin 8 on the PDIP chip, and pin 3 on the TQFP chip.

RD/ and WR/: Read and Write (input, active low)

The Read (RD/) and Write (WR/) signals are generated by the CPU or the DMA controller to access one of the 16 internal registers of the CB7210.2. In DMA mode, only the Data In and Byte Out registers are accessed.

The Read signal places the contents of the read register specified by RS_0-RS_2 on the D_0-D_7 CPU bus. The Write signal writes the data on D_0-D_7 into the write register specified by RS_0-RS_2.

The RD/ signal is pin 9 on the PDIP chip, and pin 4 on the TQFP chip. The WR/ signal is pin 10 on the PDIP chip, and pin 5 on the TQFP chip.

INT: Interrupt Request (output, active high/low)

The Interrupt Request signal becomes active when an Interrupt Status register bit is set and the corresponding bit in the Interrupt Mask Register is enabled. This signal becomes active when any one of 13 internal interrupt conditions (unmasked) are met.

The active state is software configurable. To set the polarity of the INT signal, write a 0 or 1 into bit 3 of the Auxiliary B register.

B3 = 0 Active High (B3 = 0 on Reset)

B3 = 1 Active Low

You can connect the Interrupt Request pin to the interrupt request pin of a CPU or an interrupt controller.

The INT signal is pin 11 on the PDIP chip, and pin 7 on the TQFP chip.

D_0-D_7: Data Bus (input/output)

The signals D_0 to D_7 make up an 8-bit bidirectional data bus to transfer data between the CPU and the CB7210.2.

The data bus output drivers are three-state devices that remain in the high impedance state except when the CPU executes a TLC read operation, or when the DMA controller performs a memory write operation.

The D_0 to D_7 signals are pins 12-19 on the PDIP chip, and pins 8-15 on the TQFP chip.

RS_0-RS_2: Register Select (input)

The Register Select pins select one of the eight registers (read or write) during a read or write operation. You can connect RS_0, RS_l, and RS_2 to the least significant bits RS_0, RS_l, and RS_2 of the CPU address bus, respectively.

The RS_0-RS_2 signals are pins 21-23 on the PDIP chip, and pins 18-20 on the TQFP chip.

IFC/: Interface Clear (input/output)

The IFC/ signal is a control line that clears the interface functions. IFC/ is pin 24 on the PDIP chip, and pin 21 on the TQFP chip.

REN/: Remote Enable (input/output)

The REN/ signal is a control line used to select remote or local control for a device. REN/ is pin 25 on the PDIP chip, and pin 22 on the TQFP chip.

ATN/: Attention (input/output)

The ATN/ signal is a control line that indicates whether data on the DIO lines is an interface message or a device-dependent message.

ATN/ is pin 26 on the PDIP chip, and pin 23 on the TQFP chip.

SRQ/: Service Request (input/output)

The SRQ/ signal is a control line used to request service from the controller. SRQ/ is pin 27 on the PDIP chip, and pin 24 on the TQFP chip.

DIO_1/-DIO_8/ (DIO_0/-DIO_7/): Data Input/Output (input/output)

The DIO_1/ to DIO_8/ signals make up an 8-bit bidirectional data bus to transfer remote multi-line messages between the CB7210.2 and the GPIB. Data is transferred via the SN75l60 bus transceiver. Data on these lines is in bit-parallel, byte-serial form.

The DIO_1/- DIO_8/ signals are pins 28-35 on the PDIP chip. The names for these signals n the TQFP chip are DIO_0/- DIO_7/, and are pins 25-27 and 29-33.

DAV/: Data Valid (input/output)

The DAV/ signal is a handshake line that indicates data on the DIO lines is valid. DAV/ is pin 36 on the PDIP chip, and pin 34 on the TQFP chip.

NRFD/: Ready for Data (input/output)

The NRFD/ signal is a handshake line that indicates a device is ready to receive data. NRFD/ is pin 37 on the PDIP chip, and pin 35 on the TQFP chip.

NDAC/: Data Accepted (input/output)

The NDAC/ signal is a handshake line that indicates that the transferred data is accepted. NDAC/ is pin 38 on the PDIP chip, and pin 36 on the TQFP chip.

Handshake signal function

The three handshake signals (DAV/, NRFD/, and NDAC/) coordinate the transfer of each data byte over the GPIB between sources and receivers to ensure that the data is valid and that the transfer is complete. During a data transfer, a receiver sends RFD/ passively high, indicating that it is ready for data. A source pulls DAV/ low, signifying that the data is valid. Upon receipt of valid data, all receivers set DAC passively high, indicating that the data has been accepted.

EOI/: End or Identity (input/output)

The EOI/ signal is a control line used to indicate the end of a multiple byte transfer. When used with ATN, EOI/ executes a parallel polling sequence.

The EOI/ signal is pin 39 on the PDIP chip, and pin 37 on the TQFP chip.

Control signal function

The control signals are used to manage an orderly flow of data across the interface lines.

Vcc: (input/output)

The PDIP chip has one +5 V Vcc terminal at pin 40. The TGFP chip has two +5 V Vcc terminals (labeled **+5**) at pin 16 and pin 38.

GND: Ground

The PDIP chip has one ground pin (labeled **GND**) at pin 20. The TGFP chip has ground pins (labeled **GND**) at pin 6, 17, 28, and pin 39.

Internal Registers

The CB7210.2 contains 16 registers — eight read registers and eight write registers.

Table 3-1. Read registers

RS2	RS1	RS0									
0	0	0	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	Data In (0R)
0	0	1	CPT	APT	DET	END	DEC	ERR	DO	DI	Interrupt Status 1 (1R)
0	1	0	INT	SRQ1	LOK	REM	CO	LOKC	REMC	ADSC	Interrupt Status 2 (2R)
0	1	1	S8	PEND	S6	S5	S4	S3	S2	S1	Serial Poll Status (3R page 0)
0	1	1	0	0	0	1	0	0	0	0	Revision (3R page 1)
1	0	0	CIC	ATN	SPMS	LPAS	TPAS	LA	TA	MIMN	Address Status (4R page 0)
1	0	0	T1	T0	AH2	AH1	AH0	SH2	SH1	SH0	State 1 (4R page 1)
1	0	0	DC	SR1	SR0	LE	L1	L0	SP	TE	State 2 (4R page 2)
1	0	0	C4-1	C4-0	DT	PE	PP1	PP0	RL1	RL0	State 3 (4R page 3)
1	0	0	C3-1	C3-0	C2-0	C1-0	C3	C2	C1	C0	State 4 (4R page 4)
1	0	1	CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0	Command Pass Through (5R)
1	1	0	X	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0	Address 0 (6R)
1	1	1	EOI	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1	Address 1 (7R page 0)
1	1	1	NDAC	NRFD	DAV	REN	IFC	SRQ	EOI	ATN	Bus State (7R page 1)

Table 3-2. Write registers

RS2	RS1	RS0									
0	0	0	BO7	BO6	BO5	BO4	BO3	BO2	BO1	BO0	Byte Out (0W)
0	0	1	CPT	APT	DET	END	DEC	ERR	DO	DI	Interrupt Mask 1 (1W)
0	1	0	0	SRQI	DMAO	DMAI	CO	LOKC	REMC	ADSC	Interrupt Mask 2 (2W)
0	1	1	S8	rsv	S6	S5	S4	S3	S2	S1	Serial Poll Mode (3W)
1	0	0	ton	lon	TRM1	TRM0	0	0	ADM1	ADM0	Address Mode (4W)
1	0	1	CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0	Command Pass Through (5W)
1	1	0	ARS	DT	DL	AD5	AD4	AD3	AD2	AD1	Address 0/1 (6W)
1	1	1	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0	End of String (7W)

Data registers

The CB7210.2 data registers transfer commands and data between the GPIB and the microcomputer system.

The Data In register holds data sent from the talker over the GPIB when the CB7210.2 is designated as the listener. Data is output over the data bus with a read operation. Data In register bits are held until the next eight bits of data are received. The Byte Out register holds data or a command written into it by a write operation for transfer to the GPIB. Byte Out register bits are updated at the trailing edge of the write strobe.

Data In register (0R)

D17	D16	D15	D14	D13	D12	D11	D10

Byte Out register (0W)

BO7	BO6	BO5	BO4	BO3	BO2	BO1	BO0

Interrupt registers

The Interrupt registers contain interrupt status bits, interrupt mask bits, and other non-interrupt bits.

Interrupt Status register 1 (1R)

CPT	APT	DET	END	DEC	ERR	DO	DI

Interrupt Status register 2 (2R)

INT	SRQI	LOI	REM	CO	LOKC	REMC	ADSC

Interrupt Mask register 1 (1W)

CPT	APT	DET	END	DEC	ERR	DO	Dl

Interrupt Mask register 2 (2W)

0	SRQI	DMAO	DMAI	CO	LOKC	REMC	ADSC

Interrupt bits

Thirteen factors can generate an interrupt from the CB7210.2. Each interrupt condition has an interrupt status bit and an interrupt mask bit associated with it. Interrupt status bits are always set to 1 if the interrupt condition is met. Table 3-3 lists the set and reset conditions for each interrupt status bit (bit=l=set).

The interrupt mask bit enables or disables the corresponding interrupt condition. When the interrupt mask bit is set, the corresponding interrupt condition is enabled. The INT bit in Interrupt Status register 2 is the logical OR of the enabled interrupt status bits. When an unmasked interrupt status bit is set, the INT bit=1 and the INT pin is active.

The CPU reads the interrupt status register to identify the condition that triggered the interrupt. All interrupt status register bits are cleared after a read. If an interrupt occurs during a read, the interrupt request is held until after the register is cleared and then placed in the register.

Table 3-3. Interrupt bit set and reset conditions

Bit	Set Conditions	Reset Condtions
CPT	[UCG+ACG · (TADS-LADS)] · undefined · ACDS · B_0 · + UDPCF · SCG · ACDS · B_0	pon+(Read INT Status 1 Reg) ·↓
UDPCF	[UCG+ACG · (TADS-LADS)] · undefined · ACDS · B_0	[UCG+ACG) · defined + (TAG+LAG)] · ACDS · B_0/ + pon
APT	ADM1 · ADM0 ·(TPAS+LPAS) · SCG · ACDS	pon + (Read INT Status 1 Reg) ↓
DET	DTAS	
END	LACS · (EOI+EOS · A_2) · ACDS	
DEC	DCAS	
ERR	SDYS · DAC · RFD+SIDS · (Write Byte Out reg) · + (SDYS→SIDS) ·	
DO	(TACS · SGNS) ↑	(Read INT Status 1 Reg) ↓ + TACS/ + SGNS/
DI	LACS · ACDS · Continuous Mode/	pon + (Read INT Status 1 Reg) ↓ + (Finish Handshake) · (Holdoff Mode) · + (Read Data In Reg) ·
SRQI	(CIC · SRQ · RQS/ · DAV/) ↑	pon +(Read INT Status 2 Reg) · ↓
LOKC	LOK · ↑ + LOK ↓	
REMC	REM · ↑ + REM ↓	
ADSC	TA · ↑ + TA ↓ + LA · ↑ + LA ↓ + CIC · ↑ + CIC ↓ + MJMN · ↑ + MJMN ↓ ·) l_{on}/ + t_{on}/	
CO	(CACS · SGNS) ·	(Read INT Status 2 Reg) · ↓ + CACS/ + SGNS/

Notes	Description
A_2	Auxiliary A Register
B_0	Auxiliary B Register
ADM0, 1	Address Mode register bits
LOK, REM	Interrupt Status register 2 bits
TA, LA, CIC, MJMN	Address Status register bits
Finish Handshake	FINISH HANDSHAKE auxiliary command given
Holdoff Mode	RFD Holdoff state
SDYS to SIDS	Transition from SDYS to SIDS

CPT: Command Pass Through

When B_0=l, the Command Pass Through (CPT) bit indicates that an undefined command has been received over the GPIB or that a secondary command has been received just after an undefined command. When the CPT bit is set, the DAC message is held and the handshake stops until the VALID auxiliary command is issued. The undefined command can be read from the Command Pass Through register. This status bit also indicates that an undefined primary command has been received.

APT: Address Pass Through

The Address Pass Through (APT) bit indicates that the secondary address (which the CPU is required to check in address mode 3) has been received.

When the APT bit is set, the DAV message is held and the handshake stops until either the VALID or NON-VALID auxiliary command is issued. Read the secondary address from the Command Pass through register.

DET: Device Trigger

The Device Trigger (DET) bit indicates that the device has been in DTAS. A high pulse is output when T/R_3 is used as the TRIG pin.

END

The END bit indicates that the transfer of a data block is complete. The END bit is set when either the END message (EOI) or EOS message is received. The EOS message is received when the contents of the EOS register and the Data In register are the same.

DEC: Device Clear

The Device Clear (DEC) bit indicates that the device is in DCAS.

ERR: Error

The Error (ERR) bit indicates that the contents of the Byte Out register are lost. ERR is set when data is sent over the GPIB without a specified listener or when a byte is written to the Byte Out register during SIDS or the SDYS->SIDS transition.

DO: Data Out, DI: Data In

The Data Out (DO) bit indicates a data write request to the Byte Out register. The Data In (DI) bit indicates that a data byte was written to the Data In register from the GPIB, and that the CPU has been asked to read the Data In register.

In continuous mode, the Dl bit is not set by a write to the Data In register.

When you are not using DMA, you can use the DRQ pin as the DO/Dl interrupt pin. When you do this, the DMAO and DMAI bits function as mask bits.

SRQI: Service Request Interrupt

The Service Request Interrupt (SRQI) bit indicates that an SRQ message was received while the controller is active (ClC=l).

When a service request comes from several devices, the RQS message is detected on the DIO line and the SROI bit is set again.

LOKC: Lockout Change, REMC: Remote Change

The Lockout Change (LOKC) bit indicates a change in the value of the LOK bit (RWLS+LWLS).

The Remote Change (REMC) bit indicates a change in the value of the REM bit (REMS+RWLS).

ADSC: Address Status Change

The Address Status Change (ADSC) bit indicates that a change occurred in one of the four bits of the Address Status register — TA, TA, LA, CIC, and MJMN.

TA	Talker Addresses
LA	Listener Addressed
CIC	Controller Active
MJMN	Set = Minor T/L Address. Reset = Major T/T Address

Read the Address Status Register for the values of these bits.

CO: Command Output

The Command Output (CO) bit indicates a request for a command to be written to the Byte Out register for transmission to the GPIB.

Non-interrupt related bits

LOK: Lockout, REM: Remote

The Lockout (LOK) and Remote (REM) bits indicate the state of the RL interface functions.

The LOK bit indicates that the device is in LWLS (Local with Lockout state) or RWLS (Remote with Lockout state).

The REM bit indicates that the device is in REMS (Remote state) or RWLS.

DMAO: DMA Output, DMAI: DMA Input

The DMA Output (DMAO) bit and the DMA Input (DMAI) bit enable and disable DMA transfers between memory and data registers.

When DMAO=l and the CB7210.2 is in data transmission enable mode, a DMA request is generated that asks for a data byte to be written to the Byte Out register.

When DMAI=1, a DMA request is generated that asks for data to be written to the Data In register from the GPIB.

Serial Poll Mode registers

The Serial Poll Mode register holds the `STB` (status byte S1-S6 and S8) that is sent over the GPIB, and the local message `rsv` (request service).

Serial Poll Status register (3R page 0)

Read

S8	PEND	S6	S5	S4	S3	S2	S1

Serial Poll Mode register (3W)

Write

S8	rsv	S6	S5	S4	S3	S2	S1

When the CPU sets `rsv` = l (the `rsv` message is issued), the state of the SR interface function becomes SRQS (Service Request state) when the controller is not serial polling the device.

When the polling of the controller puts the T/TE interface in SPAS (Serial Poll Active state), the contents of the serial poll mode register are sent over the DIO lines as `STB` (Status Byte) and `RQS` (Request Service) messages. The `rsv` bit is cleared when the SR interface function is in APRS.

Read the STB in the Serial Poll Mode register from the Serial Poll Status register. The PEND bit is set when rsv = l. The PEND bit is cleared when NPRS · `rsv/ = 1` (NPRS = Negative Poll Response state), or when the STB is read by the active controller. To confirm that a request was accepted and that the STB bit was transmitted, read the status of the PEND bit.

To clear the Serial Poll Mode register bits, apply a reset pulse or issue the auxiliary command `crst` (CHIP RESET).

Revision register

Revision register (3R page 1)

The Revision register contains a revision number for the CB7210.2. To access this register, write a Set Register Page command to the Auxiliary Mode register.

0	0	0	1	0	0	0	0

Address Status/Address Mode registers

The Address Mode register selects the address mode of the device, and also sets the mode for the transceiver control lines T/R_2 and T/R_3.

Address Status register (4R page 0)

CIC	ATN	SPMS	LPAS	TPAS	LA	TA	MJMN

Address Mode register (4W)

ton	lon	TRM1	TRM0	0	0	ADM1	ADM0

Setting the T/R_2 and T/R_3 pin function

The values of the TRM1 and TRM0 bits determine the functions of the T/R_2 and T/R_3 pins (Table 3-4.)

Table 3-4. T/R_2 and T/R_3 pin function select

TRM1	TRM0	T/R_2	T/R_3
0	0	EOIOE	TRIG
0	1	CIC	TRIG
1	0	CIC	EOIOE
1	1	CIC	PE

Setting the address mode

To set the address mode for a device, set the bits in the Address Mode register (4W). Address Mode functions are listed in Table 3-5.

Table 3-5. Address mode functions

ton	lon	ADM1	ADM0	Address Mode	Contents of Address register 0	Contents of Address register 1
1	0	0	0	Talk only	not used	not used
0	1	0	0	Listen only	not used	not used
0	0	0	1	Address mode 1	major talk or major listen address	minor talk or minor listen address
0	0	1	0	Address mode 2	primary address; talk or listen	secondary address; talk or listen
0	0	1	1	Address mode 3	primary address; major talk or major listen	secondary address; major talk or major listen

Address Mode 1

Address Mode 1 includes major and minor device addresses. MTA or MLA reception is indicated when either address equals the received address. The interface function is either T or L.

Address Mode 2

In Address Mode 2, Address register 0 holds the primary address and address register 1 holds the secondary address. The interface function is either TE or LE.

Address Mode 3

Address Mode 3 provides major and minor primary addresses. The CPU must identify the secondary address by reading the Command Pass Through register. The interface function is either TE or LE.

Talk Only and Listen Only modes

Address identification is not necessary in these modes. No address register is used.

State registers

State registers show the internal states of all state machines. The states are as described in the IEEE-488 specification. To access the State registers, write a Set Register Page command to the Auxiliary Command register.

State 1 register (4R page 1)

T1	T0	H2	AH1	AH0	SH2	SH1	SH0

State 2 register (4R page 2)

DC	R1	SR0	LE	L1	L0	SP	TE

State 3 register (4R page 3)

C4-1	C4-0	DT	PE	PP1	PP0	RL1	RL0

State 4 register (4R page 4)

C3-1	C3-0	C2-0	C1-0	C3	C2	C1	C0

Command Pass Through register

Command Pass Through register (5R)

CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0

The CPU reads the data on the DIO lines through the Command Pass Through (CPT) register. The contents of the CPT register can have three different meanings, based on the following three states.

- **CPT=0, B_0=1**

  This state indicates an undefined command (the command is not defined in IEEE Standard 488-1978), or a secondary command that is received after an undefined primary command. In this state, the CPT contains the data on the DIO line.
- **APT=1 Address Mode 3**

  This state indicates a secondary address.
- **After a parallel poll**

  In this state, the PPR message is in the parallel poll. If the PPRn message to be output by the CB7210.2 is *true* during the parallel poll execution in PPAS (Parallel Poll Active state), the PPR message is latched to the CPT register instead of being output to the DIO line.

  In this state, the PPR message is latched into the CPT register when CPPS (Controller Parallel Poll state) =l, until either the Controller Idle state CIDS =l, or a command byte is sent over the GPIB.

Auxiliary Mode register

The Auxiliary Mode register is a multipurpose register.

Auxiliary Mode register (5W)

CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0

A write to this register generates one of the following operations, according to the values of the CNT bits (CNT0=2):

- a write to the auxiliary register
- issue of an auxiliary command
- setting of state change prohibit time
- a write to the Parallel Poll register

Table 3-6. Auxiliary register bits

CNT			**COM**					**Function**
2	1	0	4	3	2	1	0	
0	0	0	C_4	C_3	C_3	C_2	C_0	Issues an auxiliary command specified by C_4 to C_0
0	0	1	0	F_3	F_2	F_1	F_0	The reference clock frequency (internal counter) determines T_1, T_6, T_7, and T_9
0	1	1	U	S	P_3	P_2	P_1	Sets the parallel poll register
1	0	0	A_4	A_3	A_2	A_1	A_0	Sets the Auxiliary A Register
1	0	1	B_4	B_3	B_2	B_1	B_0	Sets the Auxiliary B Register
1	1	0	0	0	0	E_1	E_0	Sets the Auxiliary E Register
0	1	0	1	P_3	P_2	P_1	P_0	Sets the Register Page
0	1	0	0	X	X	X	U	Ultra Fast T_1 Delay (350 ns)

Auxiliary commands

To issue an auxiliary command, write `000`$C_4C_3C_2C_1C_0$ to the auxiliary mode register:

iepon: Immediate execute pon, generate local pon message

C_4	C_3	C_2	C_1	C_0
0	0	0	0	0

A local message `pon` (power on) is generated and places the following interface functions into their idle states:

- SIDS (Source Idle state)
- AIDS (Acceptor Idle state)
- TIDS (Talker Idle state)
- SPIS (Serial Poll Idle state)
- TPIS (Talker Primary Idle state)
- LIDS (Listener Idle state)
- NPRS (Negative Poll Response state)
- LOCS (Local state)
- PPIS (Parallel Poll Idle state)
- PUCS (Parallel Poll Unaddressed to Configure state)
- CIDS (Controller Idle state)
- SRIS (System Control Remote Enable Idle state)
- SIIS (System Control Interface Clear Idle state)

If you issue `iepon` while a `pon` local message is already active by an external reset pulse or the auxiliary command `crst` (Chip Reset), the `pon` local message becomes *false*.

crst: Chip Reset

C_4	C_3	C_2	C_1	C_0
0	0	0	1	0

The `crst` command performs the same function as an external reset pulse. The CB7210.2 is reset to the following state:

- Sets the local message `pon`, and places the interface functions in their idle states.
- Clears the Serial Poll Mode register bits
- Clears the EOI bit
- Clears the Auxiliary A, B, and E register bits
- Clears the Parallel Poll flag and the local message rsc (request system control)
- Sets $N_F = 8$ (F_3=1, F_2=F_1=F_0=0). This clears the Address Mode bits TRM0 and TRMl.

rrfd: Finish Handshake (Release RFD)

C_4	C_3	C_2	C_1	C_0
0	0	0	1	1

The `rrfd` command ends the handshake by releasing the RFD message transmission from the holdoff state.

trig: Trigger

C_4	C_3	C_2	C_1	C_0
0	0	1	0	0

The `trig` command generates a high pulse in the TRIG pin (T/R_3 pin when TRM1=0). The `trig` command performs the same function as if the Interrupt Status register 1 bit DET was set. The trig command does not set the DET bit.

rtl: Return to Local

C_4	C_3	C_2	C_1	C_0
0	X	1	0	1

When C_3=0, the `rtl` command generates a local message in the form of pulses. When `rtl` is already set, calling `rtl` clears the local message.

When C_3=1, the `rtl` command sets the local message `rtl`.

seoi: Send EOI

C_4	C_3	C_2	C_1	C_0
0	0	1	1	0

The `seoi` command sends the END message with the next data byte. The `seoi` command is only valid for the TACS (Talker Active state).

nvld: Non-valid

C_4	C_3	C_2	C_1	C_0
0	0	1	1	1

The `nvld` command releases the DAC message held off by the address pass through. The CB7210.2 is allowed to operate as if an OSA message had been received.

vld: Valid

C_4	C_3	C_2	C_1	C_0
0	1	1	1	1

The `vld` command releases the DAC message held off by address pass through, and functions as if the remote message MSA (My Secondary Address) was received. The DAC message is released at the time of command pass through. DAC is also released if DCAS (Device Clear Active state) or DTAS (Device Trigger Active state) is in holdoff state.

sppf: Set/Reset Parallel Poll Flag

C_4	C_3	C_2	C_1	C_0
0	X	0	0	1

The `sppf` command sets the Parallel Poll Flag to the value of bit C_3.

- When the Auxiliary B register bit B_4=0, the value of the Parallel Poll flag is used as the local message `ist` (individual status).
- When the Auxiliary B register bit B_4=1, the value of SRQS is used as the local message `ist`.

gts: Go To Standby

C_4	C_3	C_2	C_1	C_0
1	0	0	0	0

The `gts` command sets the local message `gts` (go to standby) at the time of CACS (Controller Active state). The ATN line is set to *false* and the controller enters CSBS.

tca: Take Control Asynchronously

C_4	C_3	C_2	C_1	C_0
1	0	0	0	1

The `tca` command generates the local message `tca` (take control asynchronously) in the form of pulses. When `tca` is issued, the controller regains control by asserting the ATN line.

tcs: Take Control Synchronously

C_4	C_3	C_2	C_1	C_0
1	0	0	1	0

The `tcs` command sets the local message `tcs` (take control synchronously). This is effective only when CSBS+CSWS=I. The `tcs` message is cleared at the leading edge of CACS (Controller Active state).

tcse: Take Control Synchronously On End

C_4	C_3	C_2	C_1	C_0
1	1	0	1	0

The `tcse` command sets the local message `tcs` (take control synchronously on end) when the data block transfer end (END=l) is generated at CSBS. The `tcs` message is cleared at the leading edge of CACS Controller Active state).

ltn: Listen

C_4	C_3	C_2	C_1	C_0
1	0	0	1	1

The `ltn` command generates the local message `ltn` (listen) in the form of a pulse. Issue the `ltn` command when the controller is in CACS (Controller Active state).

reqt: Request RSV bit TRUE

C_4	C_3	C_2	C_1	C_0
1	1	0	0	0

The `reqt` command sets the local message `rsv` (request service) as TRUE.

reqf: Request RSV bit FALSE

C_4	C_3	C_2	C_1	C_0
1	1	0	0	1

The `reqt` command sets the local message `rsv` (request service) as FALSE.

ltnc: Listen with Continuous Mode

C_4	C_3	C_2	C_1	C_0
1	1	0	1	1

The `ltnc` command generates the local message `ltn` (listen) in the form of a pulse and places the device in continuous mode.

In continuous mode, the local message `rdy` is issued when ANRS is initiated, unless a data block transfer end is detected (END=l). When the end is detected, the device is placed in the RFD holdoff state. This action prevents generation of the `rdy` message.

In continuous mode, the DI bit is not set when a data byte is received. The continuous mode caused by ltnc is released when either the auxiliary command ltn (Listen) is issued or LIDS (Listener Idle state) is initiated.

lun: Local Unlisten

C_4	C_3	C_2	C_1	C_0
1	1	1	0	0

The `lun` command generates the local message `lun` (local unlisten) in the form of a pulse. Use `lun` when the controller is in CACS (Controller Active state).

epp: Execute Parallel Poll

C_4	C_3	C_2	C_1	C_0
1	1	1	0	1

The `epp` command sets the local message `rpp` (request parallel poll), which is cleared when CPPS+CIDS=l. The transition of the C interface function is not guaranteed if the local messages `rpp` and `gts` are issued simultaneously when CACS · STRS · SDYS=l.

sifc: Set/Reset IFC

C_4	C_3	C_2	C_1	C_0
1	x	1	1	0

The `sifc` command generates the local message `rsc` (request system control) and sets IFC to the value of C_3. To meet the IEEE Standard 488-1978, do not issue the `CLEAR IFC` command until IFC is held *true* for at least 100 µs.

- C_3=1=IFC
- C_3=0=IFC/

sren: Set/Reset Ren

C_4	C_3	C_2	C_1	C_0
1	x	1	1	1

The `sren` command generates the local message `rsc` (request system control) and sets REN to the value of C_3. To meet the IEEE Standard 488-1978, do not issue the `SET REN` command until REN is held *false* for at least 100 µs.

- C_3=1=REN
- C_3=0=REN/

dsc: Disable System Control

C_4	C_3	C_2	C_1	C_0
1	0	1	1	0

The `dsc` command clears the local message `rsc` (request system control).

Internal Counter

To access the Internal Counter, write `0010`$F_3F_2F_1F_0$ to the Auxiliary Mode register.

The Internal Counter generates the state change prohibit times (T_1, T_6, T_7, T_9) specified in the IEEE Standard 488-1978 with reference to the clock frequency.

T_1 (low speed) = $T_6 = T_7 = T_9 = (2N_F/fc) + t_{sync}$

T_1 (high speed) = $(2N_F/2fc) + t_{sync}$

Where:

N_F = integer represented by binary F_0-F_3 · $1 \leq N_F \leq 8$.

fc = reference clock frequency (clock input)

$0 \leq t_{sync} \leq$ max (reference clock high or low period)

T_1 (high speed) is used for all bytes following the first byte sent after each *false* transition of ATN if B281.

Use T_1 when you are using three-state bus drivers on the DIO, DAV, and EOI lines. Use T_1 (low speed) for all other cases.

When N_F (MHz) = fc, then:

T_1 (low speed) = $T_6 = T_7 = T_9 = 2\ \mu s + t_{sync}$

T_1 (high speed) = $500\ ns + t_{sync}$

When N_F (MHZ) < fc, IEEE Standard 488-1978 is not satisfied.

t_{sync} is a synchronization error greater than zero but less than the larger of the reference clock high and low.

$0 \leq t_{sync} \leq$ max reference clock high or low period

For a 50% duty clock:

$0 \leq t_{sync} \leq$ ½ reference clock period

Auxiliary A Register

To access the Auxiliary A Register, write `100`$A_4A_3A_2A_1A_0$ to the Auxiliary Mode register.

Auxiliary A Register bits A_0 and A_1 control the GPIB data receiving modes. Bits A_2, A_3 and A_4 control how the EOS message is used. The messages associated with data transfer are holdoff and EOS/END.

Table 3-7. Auxiliary A Register data receiving modes

A_1	A_0	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

Normal Handshake mode

In Normal Handshake mode, the local message `rdy` (ready) is generated when data is received. When the received data is read from the Data In register, `rdy` is generated in `ANRS` (Accept Not Ready state). This causes the RFD message to be transmitted and the handshake continues.

RFD Holdoff on all Data mode

In RFD Holdoff on All Data mode, RFD is not sent *true* after data is received until the microprocessor issues the auxiliary command `rrfd` (FINISH HANDSHAKE). Unlike Normal Handshake mode, this mode does not generate the `rdy` message even if the received data is read through the Data In register. The `RFD` message is not generated.

RFD Holdoff on End mode

RFD Holdoff on End mode is the same as RFD Holdoff on All Data mode when the end of the data block (EOS or END message) is detected. Handshake holdoff is released by the auxiliary command `rrfd` (FINISH HANDSHAKE).

Continuous mode

In Continuous mode, the `rdy` message is generated in ANRS (Accept Not Ready state) until the end of the data block is detected. A holdoff is generated at the end of a data block. Issue the auxiliary command `rrfd` (FINISH HANDSHAKE) to release the holdoff.

Continuous mode is useful for monitoring the data block transfer without data reception. In this mode, the Dl bit is not set by the reception of a data byte.

Table 3-8. Auxiliary A Register Functions

Bit name	Function
A_2	Permits or prohibits setting the END bit at reception of the EOS message 0 = prohibit 1 = permit
A_3	Permits or prohibits automatic transmission of the END message at the same time as the EOS message in TACS (Talker Active state) 0 = prohibit 1 = permit
A_4	Selects seven or eight bits as the valid length of the EOS message 0 = 7-bit EOS 1 = 8-bit EOS

Auxiliary B Register

To access the Auxiliary B Register, write `1011`$B_4B_3B_1B_0$ to the Auxiliary Mode register.

Table 3-9. Auxiliary B Register functions

Bit name	Function
B_0	Permit or prohibits the detection of undefined commands; permits or prohibits the setting of the CPT bits on receipt of an undefined command. 0 = prohibit 1 = permit
B_1	Permits or prohibits the transmission of the END message in SPAS (Serial Poll Active state). 0 = prohibit 1 = permit
B_2	Sets T_1 (high speed) as T_1 of handshake after transmission of second byte following data transmission 1 = T_1 (high speed)
	Sets T_1 (low speed) as T_1 in all cases. 0 = T_1 (low speed)
B_3	Specifies the active level of the INT pin. 0 = INT/ 1 = INT
B_4	1 = The value of the SRQS (Service Request state) is used as the local message `ist` (individual status). SRQS = ist = 1; SRQS = ist = 0 0 = The value of the Parallel Poll flag (PPF) is taken as the `ist` local message ist = Parallel Poll flag

Auxiliary E Register

To access the Auxiliary E Register, write `11000`$_0$`E`$_1$`E`$_0$ to the Auxiliary Mode register.

Table 3-10. Auxiliary E Register functions

Bit name	Function
E_0	Enables or disables DAC holdoff by initiating DCAS (Device Clear Active State) 1 = enables 0 = disables
E_1	Enables or disables DAC holdoff by initiating DTAS (Device Trigger Active State) 1 = enables 0 = disables

Parallel Poll register

To access the Parallel Poll register, write `0llUSP`$_3$`P`$_2$`P`$_1$ to the Auxiliary Mode register.

Bit name	Function
U	1 = No parallel poll response 0 = Parallel poll response
S	Specify the status bit polarity 1 = In Phase 0 = Reverse Phase
P_3 P_2 P_1	Specify status bit output line (DIO_1-DIO_8)

When you use the subset PP1 (remote configuration) as the PP interface function, do not write to this register. The Parallel Poll response (PPR_N) is automatically sent out according to the `PPE` (Parallel Poll enable) message issued by the GPIB controller. For example, when the values of the S bit and `ist` are equal, the PPR_N message is sent out *true* according to the specification of $P_3P_2P_1$ (=N-l).

When you use the subset PP2 (local configuration) as the PP interface function, you must write to this register in advance. The "U" bit implies the local message `lpe` (local poll enable):

- When U=0, S and $P_3P_2P_1$ have the same meaning as the bit of the same name in the PPE message. The write operation is the same as the receipt of the `PPE` message.
- When U=1, the S and $P_3P_2P_1$ bits do not carry any meaning. Reset these bits to zero.

Address registers

The CB7210.2 automatically detects two types of addresses (major address and minor address) which are held in Address registers 0 and 1 (see Table 3-5).

Address register 0 (6R)

X	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0

Address register 1 (7R page 0)

EO1	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1

Address register 0/1 (6W)

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1

Table 3-11 defines the states of the non-interrupt address status bits. The values of the non-interrupt address status bits confirm that the device has entered a particular state.

Table 3-11. Device states of the non-interrupt address status bits

Bit name	Value	Device state
MJMN	1	A minor talk or minor listen address has been received.
	0	A major talk or major listen address has been received.
TA	1	The device is in TADS (Talker Addressed state)
LA	1	The device is in the LADS (Listener Addressed state)
TPAS	1	The device is in TPAS (Talker Primary Addressed state)
LPAS	1	The device is in LPAS (Listener Primary Addressed state)
SPMS	1	The device is in SPMS (Serial Poll Mode state). For example, it has received the SPE remote message.
ATN	1	The device is in Data Mode; the ATN line is high.
CIC	1	The device is controller-in-charge.

To set the device address, write the address to address register 0/1. Address register 0/1 bits are in Table 3-12.

Table 3-12. Address register 0/1 bits

Bit name	Value		Function
ARS	0	Address register 0	Selects the address (ADl to AD5) to write the low-order bits into.
	1	Address register 1	Permits or prohibits the set address.
DT	0	permitted	(ADl to AD5) detected as a talk. This bit corresponds to DTl or DT0 of the address registers.
	1	prohibited	
DL	0	permitted	Permits or prohibits the set address (ADl to AD5) detected as a talk. This bit corresponds to DTl or DT0 of the address registers.
	1	prohibited	
AD1 to AD5	0-30		Indicates device addresses. These bits correspond to AD5-0 to AD1-0 and AD5-1 to AD1-1.

For example, when the following codes are written into address register 0/1 in Address Mode 1, the device has both the major and minor talk addresses:

AAAAA Major Talk Address

0	0	X	A	A	A	A	A

BBBBB Major Talk Address

1	0	X	B	B	B	B	B

The CB7210.2 operates as if the remote message MTA (My Talk Address) is received when the talk address of either AAAAA or BBBBB is received.

Based on the setting of the ARS bit, you can read the AD1-AD5 bits and the DT and DL bits written to address register 0/1 in either address register 0 or 1. However, in address register 0, the value of bit 7 (ADS) is unknown. In address register 1, bit 7 (EOI) indicates the value of the EOI line latched when a data byte is received.

Bus State register

Bus Status register (7R page 1)

NDAC	NRFD	DAV	REN	IFC	SRQ	EOI	ATN

The Bus State register indicates the current state of all bus control lines. To access the Bus State register, write a Set Page Register command to the Auxiliary Command register.

End of String (EOS) register

End of String Register (7W)

EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0

The End of String (EOS) register holds the seven- or eight-bit EOS message byte used by the GPIB to detect the end of a data block transfer. The length of the EOS byte is selected by A_4 (bit 4 of the Auxiliary A register).

When data is received with A_2 (bit 2 of the Auxiliary A register), End of String (EOS) is detected when both the received data and the contents of EOS are equal. When this occurs the END bit is set.

When data is transmitted with A_3 (bit 3 of the Auxiliary A register), the END message is sent at the same time as the transmitted data, if the data and the contents of EOS register are equal.

TLC Interface Functions

This chapter explains how to use the Talker/Listener/Controller (TLC) functions and how to configure the TLC in each operating mode. TLC functions are listed below:

- Source Handshake
- Acceptor Handshake
- Listener and Listener Extended
- Talker and Talker Extended
- Controller
- Service Request
- Serial Poll
- Device Clear
- Device Trigger
- Remote/Local
- Parallel Poll

State diagrams illustrate how each interface function is implemented. The state diagrams contain interface functions, remote messages, and local messages as applicable to the TLC function.

Each state that is assumed by an interface function is represented by a circle. A four-letter, uppercase mnemonic ending in **S** identifies the state. Possible transitions between states of an interface function are represented as arrows between the circles. The expression that is next to the path indicates the condition which must be *true* to enable the transition path indicated by the arrows.

- An **expression** contains one or more local messages, remote messages, state linkages, or minimum time limits used with the operators AND, OR, or NOT.
- A **remote message** received via the interface is represented by a three-letter mnemonic written in uppercase, for example, `SPE`.

  Remote message mnemonics are listed in Table 1-3.
- A **local message** to an interface function is represented by a three-letter mnemonic written in lowercase, for example, `rdy`.

  Local message mnemonics are listed in Table 1-4.
- A **linkage from another state diagram** is represented by a four-letter mnemonic enclosed in an oval, for example, `LACS`. A state linkage is *true* if the enclosed state is currently active; otherwise, it is *false*.

  State linkage mnemonics are listed in Table 1-5.
- A **minimum time limit** is represented by the symbol t_n. If a transition is qualified by a maximum time limit (within t_n), the indicated state is entered within the specified amount of time after the expression becomes *true*.

  Time limit values are listed in Table 1-2.
- **State diagram mnemonics** are listed in Table 1-5.

Source Handshake (SH1)

The CB7210.2 has complete Source Handshake capability. The Source Handshake state diagram is shown in Figure 4-1.

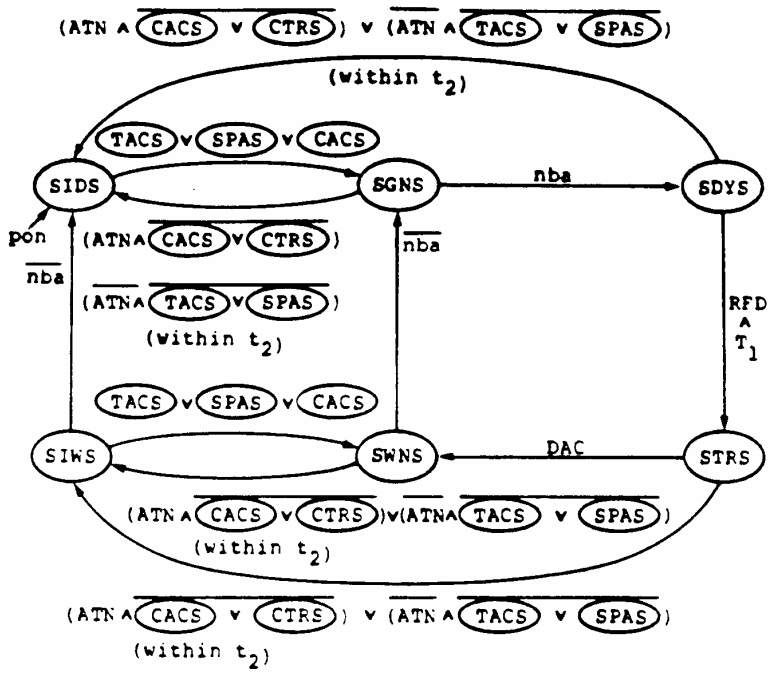


Figure 4-1. Source Handshake state diagram

Acceptor Handshake (AH1)

The CB7210.2 has complete Acceptor Handshake capability, with DAC (data accepted) and RFD (ready for data) Holdoff on certain events.

The Acceptor Handshake state diagram is shown in Figure 4-2.

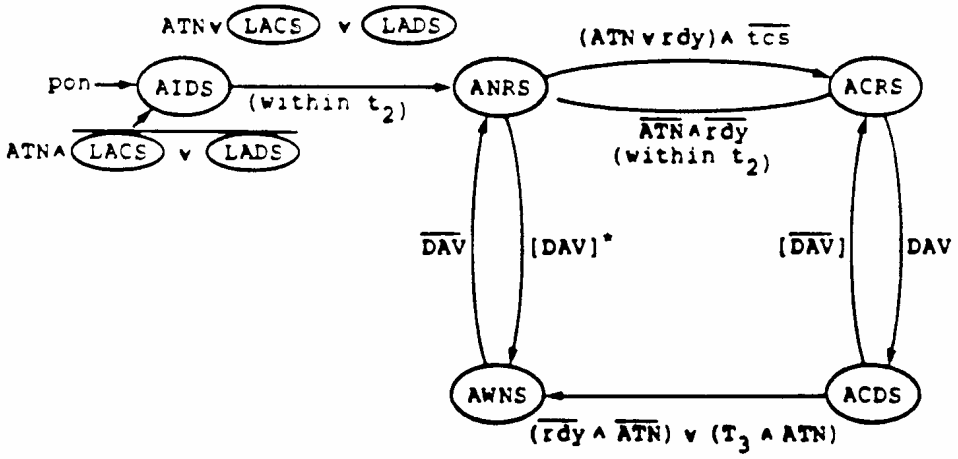


Figure 4-2. Acceptor Handshake state diagram

Listener (L3) and Listener Extended (LE3)

The CB7210.2 has complete listener capability. This includes the following features:

- Basic listener
- Listen only mode
- Unaddressed on MTA (My Talk Address)
- Detect END or EOS (End of String)
- Dual primary addressing

To enable the TLC to receive data from the GPIB, use the Listener function with the Acceptor Handshake function (Figure 4-2).

The Listener state diagram implements the Listener function in three states — LIDS (Listener Idle state), LADS (Listener Addressed state), and LACS (Listener Active state). The Listener state diagram is shown in Figure 4-3.

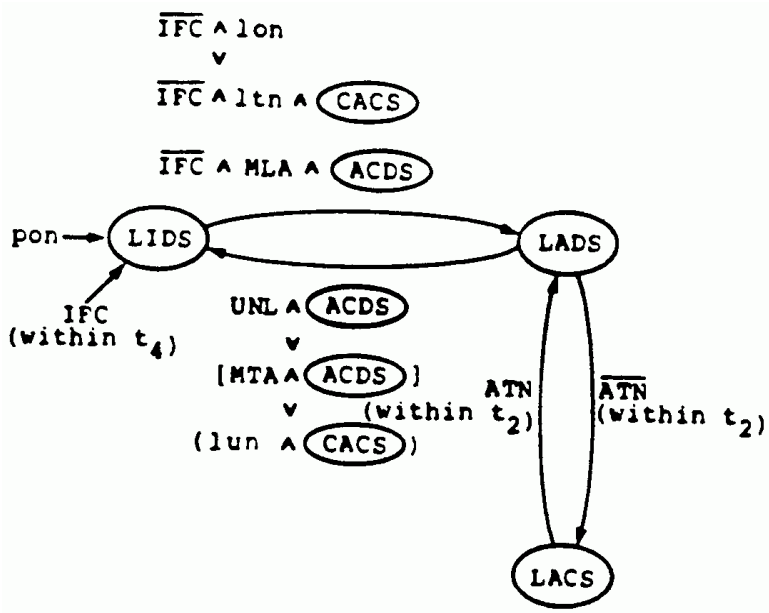


Figure 4-3. Listener state diagram

The Listener function is implemented as follows:

1. When initialized, the TLC enters the Listener Idle state (LIDS). In this state, the TLC is ready to become an active listener on the GPIB.

2. The controller sends the remote message MLA (My Listen Address) to the device:

MSB							LSB
X	0	1	L	L	L	L	L
	Listener Code			Listener address (All is not allowed)			

   Listener Address range = 20-3EH

   UNLISTEN (UNL) command = 3FH

3. The TLC responds by setting the LA address status bit, indicating that it is in the Listener Addressed state (LADS).

4. When the ATN signal goes inactive (high), the TLC goes to the Listener Active state (LACS) and waits for data.

When the TLC is in the LACS state (Listener Active state), the DI status bit is set when a data byte sent over the GPIB by a talker is in the Data In register.

The DI status bit is reset when the CPU reads the Data In register.

The Listener Extended function (LE3) defines two additional states — LPIS (Listener Primary Idle state), and LPAS (Listener Primary Addressed state). The Listener Extended state diagram is shown in Figure 4-4.

The CB7210.2 has complete Extended Listener capability:

- Basic extended listener
- Listen Only mode
- Unaddressed on MSA (My Secondary Address) * TPAS (Talker Primary Addressed state)
- Detect END or EOS (End of String)
- Dual extended addressing with software assist

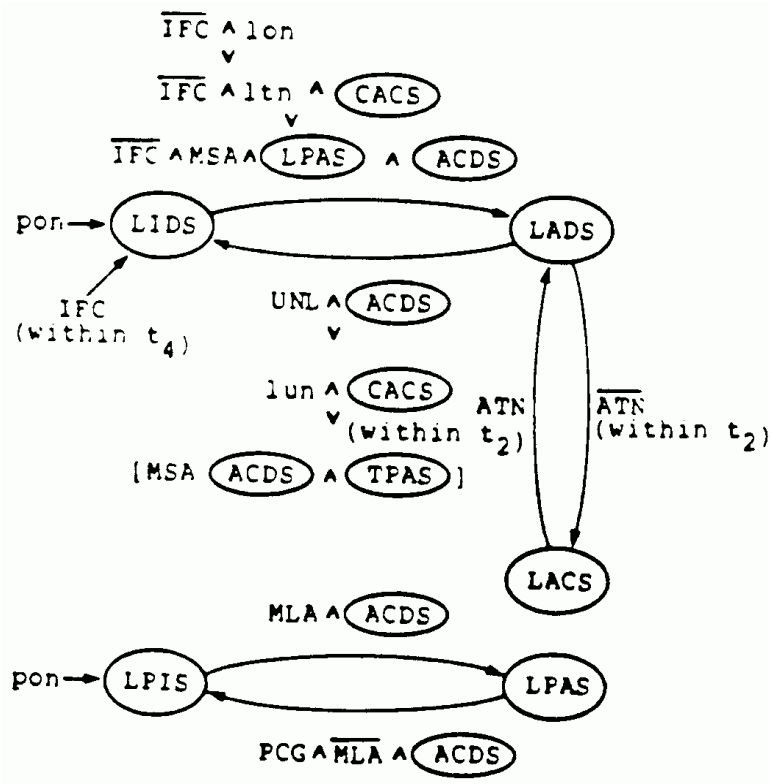


Figure 4-4. Listener Extended state diagram

1. When the TLC recognizes the MLA (My Listen Address), it enters the Listener Primary state (LPAS) and sets the TPAS status bit.

   The controller sends the secondary address to the device:

MSB							LSB
X	1	1	S	S	S	S	S
Secondary Code			Secondary address				

   Secondary address range = 60-7EH

2. When the secondary address is received, the TLC goes to the Listener Addressed state (LADS) and sets the LA status bit.

The next TLC operation is similar to that of the Listener function. The TLC allows the following listener addressing modes:

- Address Mode 1: Dual primary listen addresses
- Address Mode 2: Listener Extended mode (requires a secondary address following a primary address)
- Address Mode 3: Dual secondary addressing; Listen-only mode

Address Mode 1

To configure the TLC in address mode 1, write the value 3lH to the Address Mode register.

Address Mode register

ton	lon	TRM1	TRM0	0	0	ADM1	ADM0
0	0	1	1	0	0	0	1

The controller addresses the TLC with either the major or the minor address. Store the addresses in Address register 0/1. The value of the Address Register Select bit (`ARS`) in the Address Mode register determines whether the major address is in Address register 0 or 1.

`ARS` = 0: the major address is in Address register 0

`ARS` = l: the major address is in Address register 1

Address register 0/1

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
0/1	1	0	L	L	L	L	L

`LLLLL` = listen address

`DT` = disable talker (1)

`DL` = disable listener (1)

If only the major address is required, you must disable the minor address. If only the minor address is required, you must disable the major address.

Address register 1

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
1	1	1	0	0	0	0	0

Once the `MLA` (My Listen Address) remote message is received, the TLC sets the Listener Primary Addressed state (LPAS) and the `LA` status bits. To determine whether the `MLA` is major or minor, write a software routine that checks the value of the `MJMN` status bit:

`MJMN` = 1: major address

`MJMN` = 0: minor address

Address Mode 2

To enable Address Mode 2, store the value 32H in the Address Mode register.

Address Mode register

ton	lon	TRM1	TRM0	0	0	ADM1	ADM0
0	0	1	1	0	0	1	0

When configured in this mode, the TLC operates according to the Listener Extended (LE) state diagram (refer to Figure 4-4).

Address register 0 holds the primary address, and Address register 1 holds the secondary address.

Address register 0 (primary)

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
0	1	0	L	L	L	L	L

Address register 1 (secondary)

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
1	1	1	S	S	S	S	S

Secondary address range = 60 - 7EH

When the TLC receives the primary remote message `MTA` (My Talk Address), the TLC sets the `LPAS` (Listener Primary Addressed state) address status bit. When the TLC receives another listener primary address, the `LPAS` address status bit is reset.

After detecting the secondary address, the TLC sets the `LA` (Listener Addressed) address status bit to indicate that it is an active listener.

Address Mode 3

To enable Address Mode 3, store the value 33H in the Address Mode register.

Address Mode register

ton	lon	TRM1	TRM0	0	0	ADM1	ADM0
0	0	1	1	0	0	1	1

Address Mode 3 provides two types of primary addresses: major and minor.

Address register 0 holds the major address, and Address register 1 holds the minor address. To identify the secondary address, the CPU reads the Command Pass Through (CPT) register.

When the TLC recognizes the Primary Listen Address, it sets the `LPAS` address status bit. If another Primary Listen Address is sent, the `LPAS` address status bit is reset and the Listener function goes to the Listener Idle state (LIDS). To determine whether the received Primary Address is major or minor, the CPU reads the `MJMN` address status bit.

When the TLC receives a secondary address, the following events occur:

1. The TLC stops the handshake by holding off the DAC handshake signal until the `vld` (valid) or `nvld` (non-valid) auxiliary command is issued.
2. The TLC sets the `APT` status bit to indicate that the CPU may examine the secondary address in the CPT register.
3. The CPU reads the secondary address through the CPT register, and determines if the address is valid or invalid.
   - If the secondary address is valid, the CPU issues the `vld` (VALID) auxiliary command. Upon receipt of the `vld` command, the TLC assumes that the My Secondary Address (MSA) has been received. The TLC then sets the LA bit and releases the DAC handshake signal to finish the handshake.
   - If the secondary address is invalid, the CPU issues the `nvld` (non-valid) auxiliary command. When the TLC receives the `nvld` command, it remains in the Listener Primary Addressed state (LPAS) and releases the DAC signal to finish the handshake.

Listen-only mode

To enable Listen-only mode, store the value 70H in the Address Mode register.

Address Mode register

ton	lon	TRMl	TRM0	0	0	ADM1	ADM0
0	1	1	1	0	0	0	0

If ATN is low, setting bit 5 of the Address Mode register places the TLC into a Listener Addressed state (LADS). If ATN is high, setting bit 5 of the Address Mode register places the TLC into a Listener Active state (LACS).

The TLC does not require an address when used in Listen-only mode — disable address registers 0 and 1.

Address register 0 [6R]

ARS	DT	DL	AD5-0	AD4-0	AD3-0	AD2-0	ADl-0
0	1	1	0	0	0	0	0

Address register 1 [7R]

ARS	DT	DL	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1
1	1	1	0	0	0	0	0

If there is no controller on the GPIB, configure the TLC in Listen-only mode. This action allows the TLC to receive data from the active talker on the bus.

Talker (T5) and Talker Extended (TE5)

The CB7210.2 has complete talker capability. This includes the following features:

- Basic talker
- Serial poll
- Talk only mode
- Unaddressed on `MLA` (My Listen Address)
- Send END or EOS (End of String)
- Dual primary addressing

The TLC uses the talker function with the source function to send data on the GPIB. The TLC implements the talker function in four states, as shown in the talker state diagram (Figure 4-5):

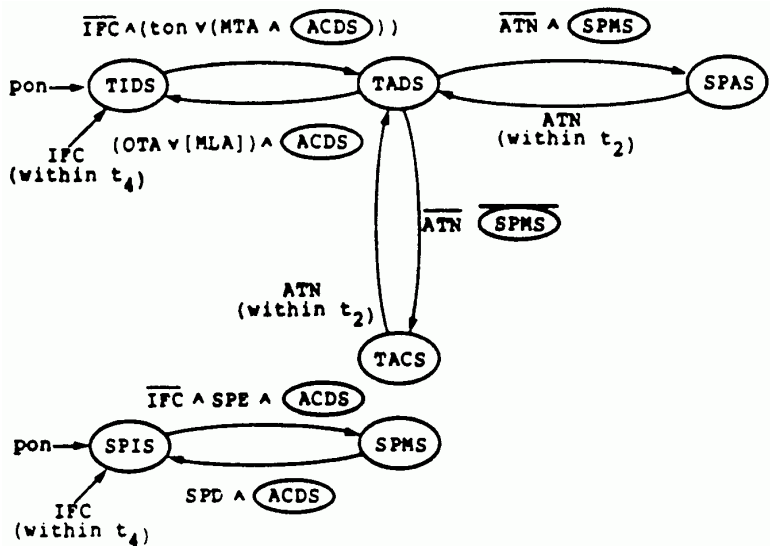


Figure 4-5. Talker state diagram

SPIS and SPMS are the two states that are separate from the talker state diagram. These states enable the Talker function to respond to a serial poll conducted by the controller. The talker function is implemented as follows:

1. Once initialized, the TLC is in the Talker Idle state (TIDS) and ready to become a talker on the GPIB. The controller sends the remote message `MTA` (My Talk Address) to the device:

MSB							LSB
X	1	0	T	T	T	T	T
	Talker code		Device address (All is not allowed)				

   Talker address range = 40 - 5EH

   Untalk command (UNT) = 5FH

2. The TLC responds by setting the TA (Talker Addresses) status bit to indicate that it is in the Talker Addressed state (TADS).

3. When the ATN signal goes high, the TLC enters the Talker Active state (TACS).

   In this state, the DO status bit is set so that the TLC is ready to talk and the Byte Out register is empty.

4. Writing to the Byte Out register places the data byte on the DIO_1-DIO_8 data lines and resets the DO status bit. DO status bit is set again when the listener(s) sends the DAC handshake signal *true*.

The Talker Extended function defines two more states — Talker Primary Idle state (TPIS) and Talker Primary Addressed state (TPAS). The Talker Extended (TE) state diagram is shown in Figure 4-6.

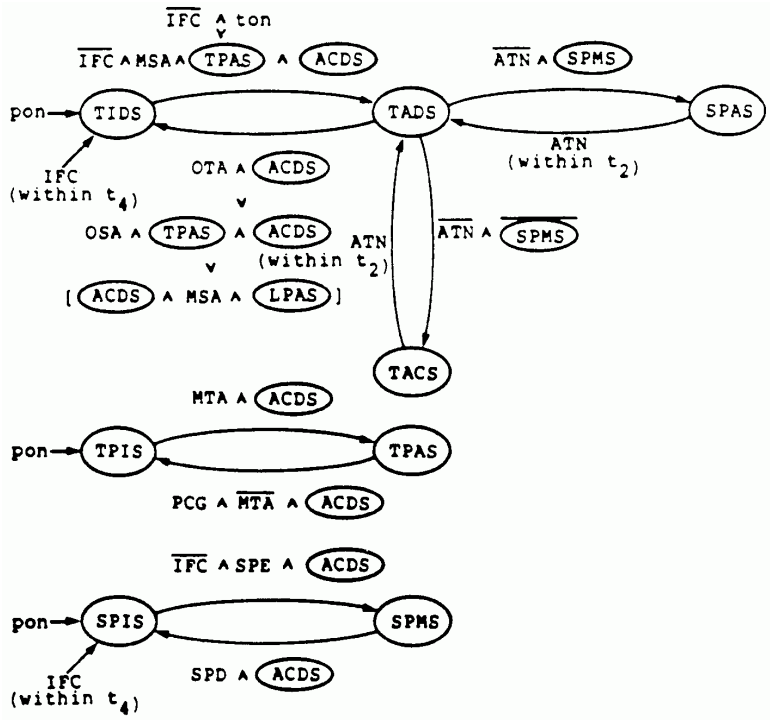


Figure 4-6. Talker extended diagram

The Talker Extended function is implemented as follows:

1. Once the TLC receives the MTA, it goes to the Talker Primary Addressed state (TPAS) and sets the `TPAS` status bit.

2. The controller sends the secondary address as follows:

MSB							LSB
X	1	1	S	S	S	S	S
	Secondary code		Secondary address				

3. After recognizing the secondary address, the TLC enters the Talker Addressed state (TADS) and sets the `TA` address status bit (Talker Addresses).

The following TLC operations are similar to those of the talker function discussed above. The TLC allows the following talker addressing modes:

- Address Mode 1: Dual primary talk addressing
- Address Mode 2: Talker Extended mode, secondary addressing
- Address Mode 3: Dual secondary addressing; Talk-only mode

Address Mode 1

To configure the TLC in Address Mode 1, store the value 31H in the Address Mode register.

Address Mode register

ton	lon	TRMl	TRM0	0	0	ADM1	ADM0
0	0	1	1	0	0	0	1

The controller addresses the TLC with either the major or minor address. Store the major address in address register 0, and the minor address in address register 1. The setting of the ARS Address 0/1 register bit determines which register contains the major address:

ARS = 0: The major address is in address register 0.
ARS= 1: The major address is in address register 1.

Address register 0/1

ARS	DT	DL	ADS	AD4	AD3	AD2	AD1
0/1	0	1	T	T	T	T	T

TTTTT = talk address

If you only require the major address, you must disable the minor address. If you only require the minor address, you must disable the major address.

Address register 1

ARS	DT	DL	ADS	AD4	AD3	AD2	AD1
1	1	1	0	0	0	0	0

When the TLC receives the primary remote message MTA (My Talk Address), the TLC sets the TPAS (Talker Primary Addressed State) and TA (Talker Addresses) status bits. To determine whether the MTA is major or minor, write a software routine that checks the value of the MJMN status bit:

MJMN = 0: Minor address
MJMN = l: Major address

The TLC is now ready to send data over the GPIB.

Address Mode 2

To enable Address Mode 2, store the value 32H in the Address Mode register.

Address Mode register

ton	lon	TRM1	TRM0	0	0	ADM1	ADM0
0	0	1	1	0	0	1	0

When configured in this mode, the TLC operates according to the talker extended (TE) state diagram (refer to Figure 4-6).

Address register 0 holds the primary address, and Address register 1 holds the secondary address.

Address register 0 (primary)

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
0	1	0	T	T	T	T	T

Address register 1 (secondary)

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
1	1	1	S	S	S	S	S

Secondary address range = 60 - 7EH

When the TLC receives the primary remote message MLA (My Listen Address), the TLC sets the TPAS (Talker Primary Addressed state) address status bit. When the TLC receives another talker primary address, the TPAS address status bit is reset.

After detecting the secondary address, the TLC sets the TA address status bit (Talker Addresses) to indicate that it is an active listener.

Address Mode 3

To enable Address Mode 3, store the value 33H in the Address Mode register.

Address Mode register

ton	lon	TRMl	TRM0	0	0	ADM1	ADM0
0	0	1	1	0	0	1	1

Address Mode 3 provides two types of primary addresses — major and minor.

Address register 0 holds the major address, and Address register 1 holds the minor address. The CPU identifies the secondary address by reading the Command Pass Through (CPT) register.

When the TLC receives the primary talk address, it sets the `TPAS` (Talker Primary Addressed State) address status bit. If another primary talk address is sent, `TPAS` is reset and the TLC goes to the Talker Idle state (TIDS). The CPU checks the `MJMN` status bit to determine whether the primary talk address received is the major address or the minor address.

When the TLC receives a secondary address, the following events occur:

1. The TLC stops the handshake by holding off the DAC handshake signal until either the `vld` (valid) or `nvld` (non-valid) auxiliary command is issued.
2. The TLC then sets the `APT` status bit to indicate that the secondary address is present and that the CPU can examine it by reading the CPT register.
3. The CPU reads the secondary address through the CPT register, and determines if the address is valid or invalid:
   - If the secondary address is valid, the CPU issues the `vld` auxiliary command. When the TLC receives the `vld` command, it assumes that the remote message `MSA` (My Secondary Address) is received, and enters TADS (Talker Addressed state). This action sets the status of the `TA` address status bit. To finish the handshake, the TLC releases the `DAC` (Data Accepted) handshake signal.
   - If the secondary address is invalid, the CPU issues the `nvld` auxiliary command. When the TLC receives the `nvld` command, it releases the DAC handshake signal to complete the handshake, and remains in the Talker Primary Addressed state (TPAS).

Talk-only mode

To enable talk-only mode, store the value B0H in the address mode register.

Address mode register

ton	lon	TRMl	TRM0	0	0	ADM1	ADM0
1	0	1	1	0	0	0	0

Once configured in this mode, the TLC goes to the Talker Addressed state (TADS) if the ATN signal is *true* (low), or the Talker Active state (TACS) if the ATN signal is *false* (high).

The TLC is not addressed in Talk-only mode – disable Address registers 0 and 1.

Address register 0

ARS	DT	DL	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0
0	1	1	0	0	0	0	0

Address register 1

ARS	DT	DL	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1
1	1	1	0	0	0	0	0

If there is no controller on the GPIB, configure the TLC in Talk-only mode to enable the TLC to receive data from the active talker on the bus.

Controller (C_1 through C_5)

The TLC uses the Controller function to send remote messages (device addresses and commands) to devices on the GPIB. There may be more than one controller on a system, but only one can be the Controller-in-Charge (CIC) at any time.

The CB7210.2 has complete device clear capability. This includes the following features:

- System Controller
- Send IFC (Interface Clear) and take charge
- Send REN (Remote Enable)
- Respond to SRQ (Service request)
- Send interface messages
- Received control
- Parallel Poll
- Take control synchronously and asynchronously

The Controller function state diagram is shown in Figure 4-7.

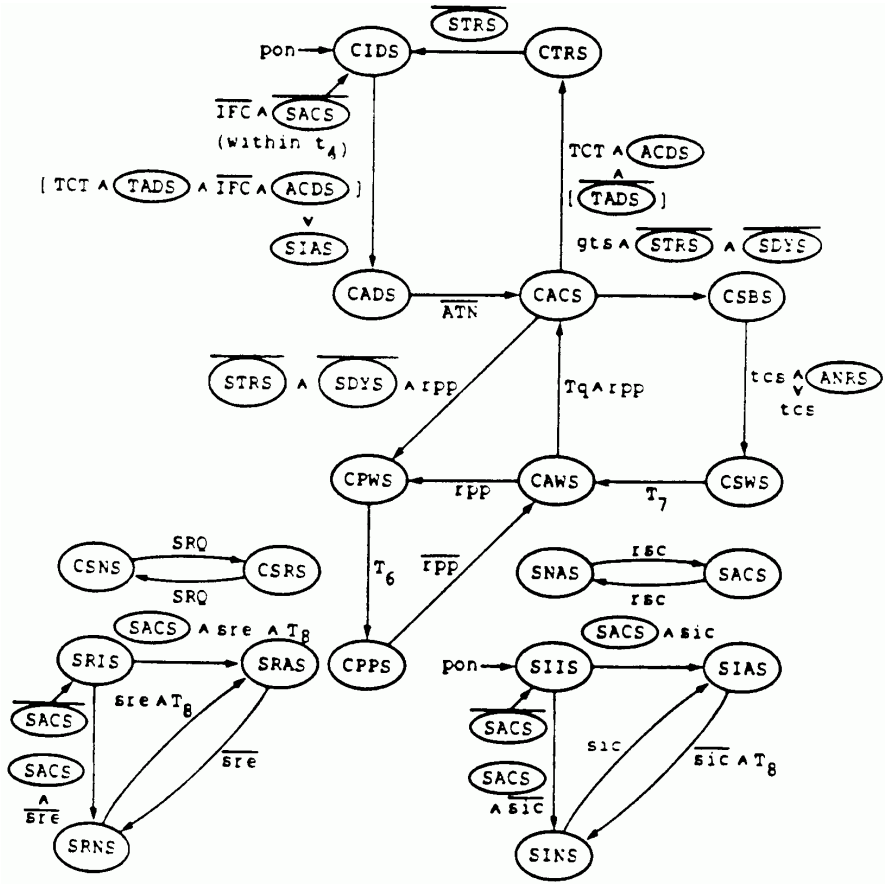


Figure 4-7. Controller state diagram

The controller function is implemented as follows:

1. Once the TLC is initialized, the controller function goes to the Controller Idle state (CIDS). The TLC can enter controller mode under local (auxiliary) or remote command.

2. If the TLC is the system controller, the remote message `IFC` (Interface Clear) can be sent. To issue the auxiliary command `sifc` (set IFC), write `00011110` to the Auxiliary Mode register. After a software delay of 100 µs or more, the Clear IFC auxiliary command is issued by writing `00010110` to the same register.

3. The TLC enters the Controller Active state (CACS) whenever an `IFC` message is sent to other devices on the GPIB. In the controller active state, the ATN signal is sent *true* (low).

   Control can be obtained when the TLC is in the Talker Addressed state (TADS) and the `TCT` (Take Control) remote command is issued by the Controller-in-Charge. The TLC becomes Controller-in-Charge when the previous controller releases the ATN line.

4. Once the TLC enters CACS, the Controller-in-Charge status bit is set to confirm that the TLC is in CACS. The `CO` (command output) status bit is set, requesting that a command be written to the byte out register.

Service Request (SR1)

The Service Request function allows the TLC to request service from the active controller. The Service Request function is implemented in three states:

- NPRS (Negative Poll Response state)
- SRQS (Service Request state)
- APRS (Affirmative Poll Response state)

The Service Request state diagram is shown in Figure 4-8.

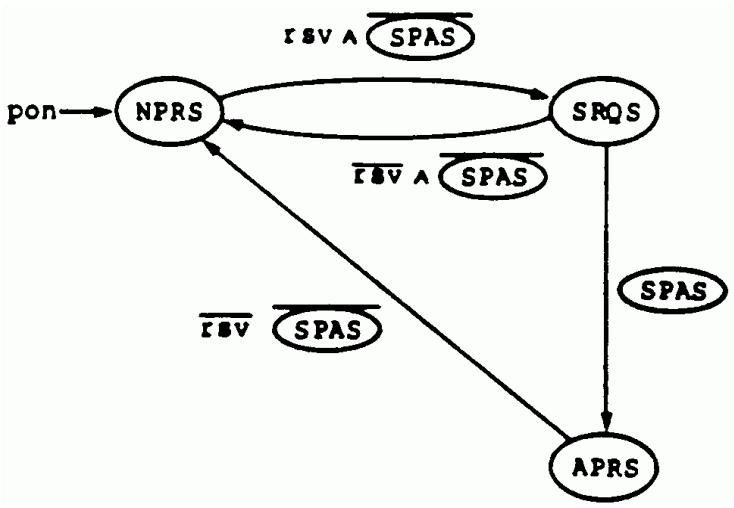


Figure 4-8. Service Request state diagram

The Service Request function is implemented as follows:

1. The `STB` (Status Byte) is written with the local message request service (rsv=l) to the Serial Poll Mode register to request service. The Serial Poll Mode register is shown here:

S8	1	S6	S5	S4	S3	S2	S1

S7=rsv

2. If the TLC is not in the Serial Poll Active state (SPAS), the SRQ message is sent *true* when the rsv message (S7) is set and the SR function enters the Service Request state (SRQS).

   If the TLC is already in SPAS, the SRQ message remains *false* until serial polling is complete (SPAS=0).

The `PEND` bit (S7) in the Serial Poll Status register indicates whether a service request is accepted or left pending. The PEND bit is set when rsv=1, and cleared when either the STB is read out by the active controller, or the local message rsv is cleared before SPAS=l.

The `STB` (Status Byte) is sent out on the data lines when the TLC is in the Serial Poll Active state (SPAS). It is transmitted only once, even if the active controller does not assert ATN after the first transfer. The END message is sent out if Bl=l (bit 1 of the auxiliary B register).

Serial Poll (SP)

When the TLC as Controller-in-Charge receives a service request, it uses the Serial Poll function to poll each device (talker) to determine which device is requesting service. The Serial Poll state diagram is included in the Talker state diagram (Figure 4-5) and the Talker Extended state diagram (Figure 4-6).

When the TLC receives a service request, it implements the Serial Poll function, as follows:

1. The TLC sets the `SRQI` status bit and tells the controller to initiate the poll.
2. The controller sends the remote message `SPE` (Serial Poll Enable).

   This action places all talkers in the Serial Poll Mode state (SPMS), and sets the `SPMS` status bit in the address status register. Each device is sequentially addressed to talk.
3. Once the controller releases the ATN line, the addressed device goes to SPAS and places its status on the GPIB data bus.
4. The controller accepts the status byte. Depending on whether the DIO7 bit is *true* or *false*, the following action occurs:
   - If the DIO7 bit of the status byte is *true*, the controller verifies that this is the device that requested service, makes the ATN line *true*, and completes the serial poll by sending the remote message Serial Poll Disable (SPD).

     When the requesting device is in SPAS, it releases the SRQ line. If another device also requests service at the same time, the SRQ line remains low.
   - If the DIO7 bit of the status byte of the addressed device is *false*, the controller knows that this device did not request service, and checks the next device in its polling routine. The controller can send `SPD` (Serial Poll Disable) after it receives the status byte of the requesting devices and the SRQ (service request) line is high.

For the TLC controller to accept the status byte, perform the following sequence:

1. Set the TLC in RFD Hold-off on All Data Mode by setting Al=0 and A0=1 in the Auxiliary A Register.
2. Issue the `ltn` (Listen) auxiliary command to generate the `ltn` (listen) local message.
3. Issue the auxiliary command `gts` (go to standby) to place the TLC in CSBS (Controller Standby state).
4. To release RFD, issue the `rrfd` (Finish Handshake) auxiliary command after receiving the status byte.

Figure 4-9 shows the procedure for a serial poll conducted by the Controller-in-Charge.

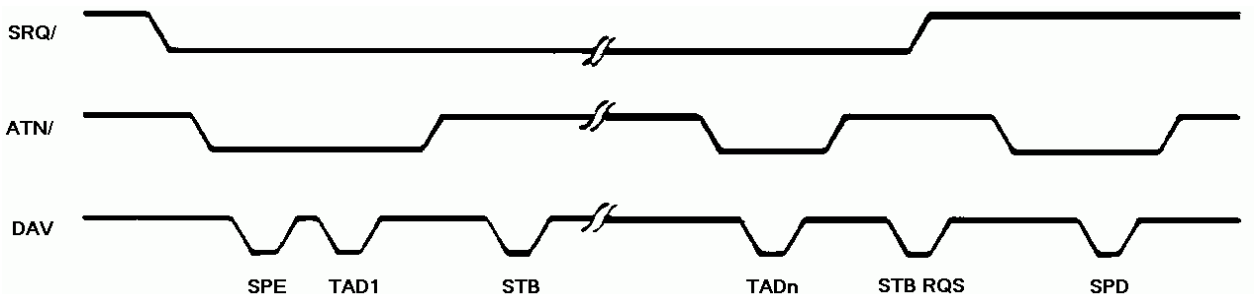


Figure 4-9. Serial poll routine

Device Clear (DC1)

The Device Clear function allows you to set the TLC to an initial state. There are two states in the Device Clear state diagram — Device Clear Idle state (DCIS) and Device Clear Active state (DCAS). The Device Clear state diagram is shown in Figure 4-10.

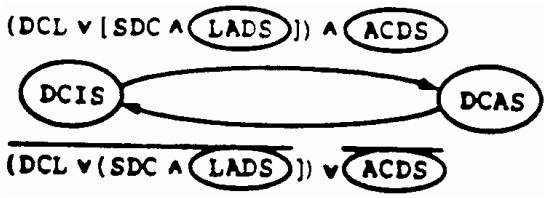


Figure 4-10. Device Clear state diagram

The TLC goes to DCAS when either of the following conditions is present:

- The TLC is addressed to be a listener (as with LADS) and the controller sends the Specified Device Clear (SDC) command `X0000100`

  or

- The controller sends the Device Clear (DCL) command `X0010100`.

When the TLC enters DCAS, the `DET` status bit is set to inform the CPU to take action that is appropriate to the Device Clear software routine.

Writing `110000X1` to Auxiliary E Register allows the TLC to stop the handshake by holding off the DAC handshake line whenever either the SDC or DCL command is sent over by the controller.

The TLC stays in DCAS as long as the `SDC` (Specified Device Clear) and `DCL` (Device Clear) commands are *true*.

Device Trigger (DT1)

The Device Trigger function triggers one or more devices on the GPIB to initiate a pre-program action. There are two states in the device trigger state diagram — Device Trigger Idle state (DTIS) and Device Trigger Active state (DTAS). The Device Trigger state diagram is shown in Figure 4-11.

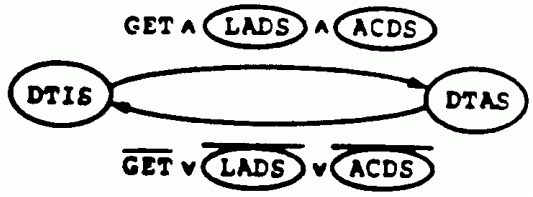


Figure 4-11. Device Trigger state diagram

The TLC goes to DTAS when it is in LADS (Listener Addressed state) and the controller sends the Group Execute Trigger (GET) command. When the TLC enters DTAS, the DET status bit is set to inform the CPU of the receipt of the GET command.

Write 110000Xl to Auxiliary E Register to allow the TLC to stop the handshake by holding off the DAC handshake line when the GET command is sent by the controller. This gives the CPU time to take any necessary action.

The `trig` (Trigger) auxiliary command `00000100` generates a high pulse in the T/R_3 pin when TRMl=0 and TRM0=0 or 1. Issuing a `trig` command does not set the DET status bit.

Remote/Local (RL1)

The Remote/Local (RL) function allows you to configure the device from either its front panel control switches or from the controller.

If the device has a front panel remote/local switch, use this switch to regain local control of the device. However, the Controller-in-Charge may issue the LLO (Local Lockout) universal command which disables the front panel remote/local switch. The Remote/Local state diagram is shown in Figure 4-13.

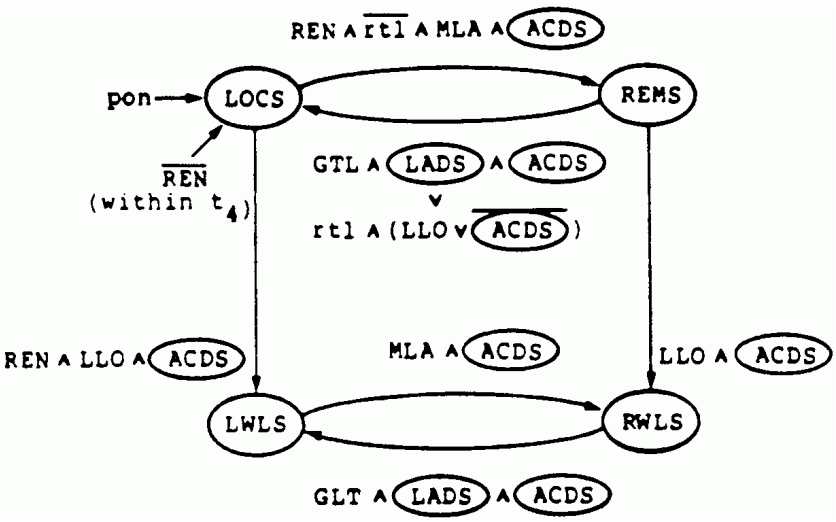


Figure 4-12. Remote Local state diagram

All devices return to local mode when the system controller sends the REN (Remote Enable) line *false*. However, the Controller-in-Charge may place selected devices in local mode by sending the `GTL` (Go To Local) addressed command.

Any device which is in LADS (Listener Addressed state) enters local mode when it receives the `GTL` command, and remains in local mode until `MLA` (My Listen Address) is sent. There are four states in the Remote/Local state diagram:

- Local state (LOCS)
- Remote state (REMS)
- Local with Lockout state (LWLS)
- Remote with Lockout state (RWLS)

To place the TLC in REMS, the controller must send the REN line *true* followed by the remote message `MLA` (My Listen Address).

The REM and LOK non-interrupt status bits are set when the following conditions occur:

REM	LOK	State of TLC
0	0	LOCS
0	1	LWLS
1	0	REMS
1	1	REWS

The `LOKC` (Lockout Change) and `REMC` (Remote Change) interrupt status bits are set when a change occurs in the value of the LOK and REM bits, respectively.

When the auxiliary command `rtl` (Return to Local) is issued, the TLC leaves REMS and enters the Local state (LOCS).

Parallel Poll (PP1 and PP2)

The Parallel Poll function allows a device to send one bit of status information to a controller that is requesting a response.

Unlike serial poll, which is device-initiated, parallel poll is controller-initiated. There are two methods for configuring a device for parallel poll — remote (PP1) and local (PP2) configurations. The parallel poll state diagram is shown in Figure 4-13.

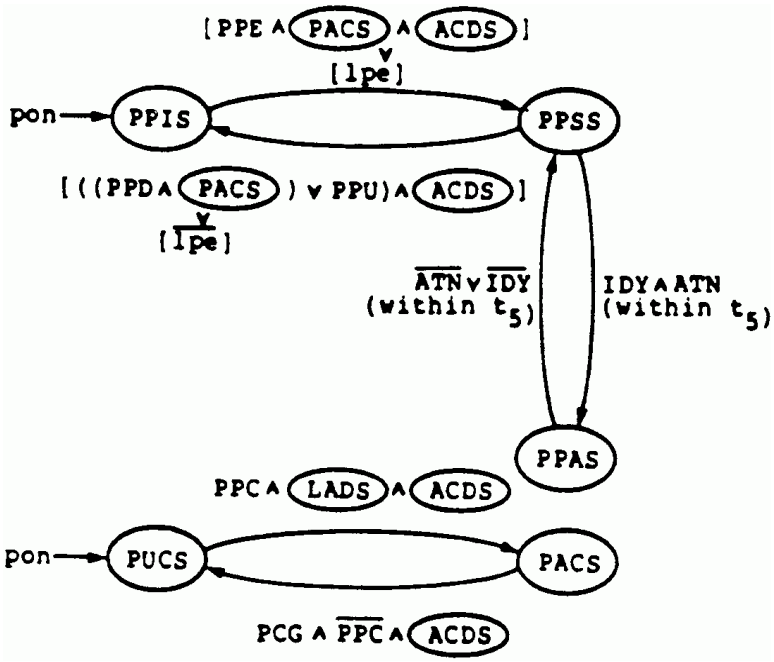


Figure 4-13. Parallel Poll state diagram

In remote configurations (PP1), the Controller-in-Charge uses the Source Handshake with ATN asserted to transmit a sequence of commands to initiate the Parallel Poll response.

A typical parallel poll sequence is listed here defines the commands issued by the Controller-in-Charge:

1. `UNL` - Unlisten command. Places all listeners in LIDS (Listener Idle state).
2. `MLA` - My Listen Address command. Places the device in LADS (Listener Addresses state).
3. `PPC` - Parallel Poll Configure command. Places the device in PACS (Parallel Poll Addressed to Configure state).
4. `PPE` - Parallel Poll Enable command. Places the device in PPSS (Parallel Poll Standby state).

   The `PPE` command configures the devices that have received the PPC command. `PPE` configures the devices to respond to a parallel poll on a particular GPIB DIO line. The DIO line must have a particular polarity (high or low).

   This command is defined by IEEE Standard 488-l978 as follows:

MSB							LSB
X	1	1	0	S	P3	P2	P1

`S` specifies the value of the Parallel Poll Response (PPR):

S	Response
0	0
1	0

Bits P_1-P_3 specify the PPR message to send when you execute a Parallel Poll:

P_1	P_2	P_3	PPR Message
0	0	0	PPR1
0	0	1	PPR2
0	1	0	PPR3
0	1	1	PPR4
1	0	0	PPR5
1	0	1	PPR6
1	1	0	PPR7
1	1	1	PPR8

You can configure a device by a sequence of commands, such as `UNL` (Unlisten), `MLA` (My Listen Address), `PPC` (Parallel Poll Configure), and `PPD` (Parallel Poll Disable – 70H), so that it never responds to a parallel poll. The PPD command places the device in the Parallel Poll Idle state (PPIS).

After the controller configures all of the required devices, it can send the local message `rpp` (request parallel poll) *true*. This action places the controller in the Controller Parallel Poll state (CPPS).

- When in the PPIS or CPPS state, the controller executes parallel poll by sending ATN and EOI *true*, and obtains all the status bits. If the controller transmits the `PPU` (parallel poll unconfigure) command, no devices will respond to parallel poll. This command places them in PPIS.
- If the TLC is Controller-in-Charge, Parallel Poll is executed by issuing the `epp` (execute parallel poll) auxiliary command. When you issue the `epp` command, the local message `rpp` (request parallel poll) is set, and the CO status bit is reset. When the TLC returns to CACS (Controller Active state), CO is set again to indicate that the parallel poll execution is finished. You can read the status bits from the CPT register and clear `rpp`.
- If the TLC is addressed to listen, the Controller-in-Charge can remotely configure the TLC for parallel poll.

Use the Auxiliary B Register bit B_4 to set the value of the individual status (local message `ist`):

B_4=0 The value of the Parallel Poll Flag is taken as the `ist` local message.

`ist` = PPF (Parallel Poll Flag)

B_4=1 The value of SRQS is used as the local message `ist`.

The value of the Parallel Poll Flag is ignored.

`ist` = SRQS (Service Request state)

- To set PPF, write `00001001` to the Auxiliary Mode register.
- To reset PPF, write `00000001` to the Auxiliary Mode register.

In the local parallel poll configuration (PP2), the specifications are made from the device. To configure the TLC for the Parallel Poll Response, write `011USP`$_3$`P`$_2$`P`$_1$ to the Auxiliary Mode register.

`U`=0 `011USP`$_3$`P`$_2$`P`$_1$ is the `lpe` (local poll enable) local message.

`U`=l The TLC does not respond to the poll.

The TLC is configured in the `S` bit. The PPRn will be sent *true* only if the Parallel Poll Flag (set by the local message `ist` (individual status) matches this bit. During normal operation, the CPU will set or clear PPF (`ist` if B_4=0), according to the device's need for service. Consequently, the TLC is set up to give the proper response to an `IDY` request (Identify: ATN · EOI) without directly involving the CPU.

Using the TLC

This chapter explains how to perform various tasks using the CB7210.2 TLC, such as:

- Transmit a command
- Process a command that is not defined
- Start a data transfer
- Transmit and receive data
- Complete and stop a block transfer
- Specify the controller as Listener

Transmitting commands

After receiving a request for command transmission (C_0=l), write code to the Byte Out register to transmit a command. Repeat this process to send several command bytes.

Processing undefined commands

When an undefined command is received when B_0=1, the TLC sets the CPT status bit and holds off the `DAC` (Data Accepted) bit to stop the handshake. The CPU reads the undefined command from the CPT register. When you issue the `vld` (Valid) auxiliary command to the TLC, the `DAC` is released.

When an undefined command is received when B_0=0, the handshake does not complete until a defined command is received. The CPT status bit remains 0, and the received code (command) is ignored.

A secondary command is received immediately after an undefined primary command. This type of command is handled as an undefined command.

Beginning data transfer

Issue the `gts` (go to standby) auxiliary command after the TLC (configured as Controller-in-Charge) specifies the talker and the listeners. The `gts` command places the TLC in CSBS (Controller Standby state) and makes ATN high. Data transfer begins after the last command you write to the Byte Out register is transmitted.

Transmitting data

When the TLC is ready to send data (D0=l), the CPU writes one byte of data to the Byte Out register. This process is repeated to send several bytes. The DO bit clears when you read Interrupt Status register 1, or when you write to the Byte Out register.

Receiving data

When the CPU receives a data receive request (DI=l), it reads the contents of the Data In register. Data is received four ways:

- Normal Handshake mode
- RFD Holdoff on All Data mode
- RFD Holdoff on End mode
- Continuous mode

Normal Handshake mode (A0=0, A1=0)

When the TLC receives data as a listener, the DI status bit is set and the `RFD` message is sent *false*. After the CPU reads the data byte from the Data In register, the `RFD` message is sent *true*. This informs the talker that the listener is ready for the next byte.

RFD Holdoff on All Data mode (A0=1, A1=0)

When the TLC receives data as a listener, Dl is set and the `RFD` (Ready For Data) message is sent *false* (as in the normal handshake mode). The `RFD` message is held *false* until the `rrfd` (Finish Handshake) auxiliary command is issued, even if the CPU reads the received data from the Data In register. While the `RFD` message is held *false*, the active talker cannot send the next byte. This allows the CPU time to perform necessary actions.

RFD Holdoff on End mode (A0=0, A1=1)

In this mode, data reception continues just as in normal handshake mode until the `EOI` (`END`) message or the `EOS` (End of String) message is received. This sets the END status bit and stops the handshake by holding off on `RFD` (Ready For Data). To release `RFD`, issue the `rrfd` (Finish Handshake) auxiliary command.

Continuous mode (A0=1, A1=1)

Use Continuous mode when the TLC is configured as Controller-in-Charge. In this mode, the TLC continuously cycles through the Acceptor Handshake state diagram, and automatically generates the local message `rdy` (ready) when it enters ANRS (Accept Not Ready state). When the end of a data block (EOI or EOS received) is received, the TLC generates a holdoff. This causes the GPIB to hang up in ANRS. At this point you can take control. To complete the handshake, issue the `rrfd` (Finish Handshake) auxiliary command.

When the `ltnc` (Listen with Continuous Mode) auxiliary command is issued at CACS (Controller Active state), the operation is the same as Continuous mode.

Completing data block transfer

You can use the EOS message or the EOI line to detect or transmit the end of a data block.

Using the EOS message

The EOS register holds the EOS code. The value of A_4 determines the length of the EOS code:

A_4=0 7-bit EOS

A_4=1 8-bit EOS

You can define the EOS code. However, if you use code that appears in the data byte you will not be able to identify the EOS message. Do not use EOS code if the data bytes are a full eight bits. With ASCII data, you can use the `LF` code as EOS and the other codes as data bytes to ensure that the EOS message is unique.

Transmitting and detecting the EOS message

To transmit an EOS message, write the EOS code to the Byte Out register when DO=l. This is the same method used to transmit a data byte.

If A_2=1 when the EOS message is received, the END status bit is set. When both the `STB` (Status Byte) code in the Serial Poll register and the EOS (End of String) code in the EOS register agree, this *does not* imply that the EOS message is received.

Using the EOI line

You can use the EOI line to detect the end of a data block, as the `END` (EOI) message is sent *true* when the last byte of the data block is transmitted.

Transmitting and detecting the END message

When the `seoi` (Send eoi) auxiliary command is issued to the TLC, the `END` message is transmitted simultaneously with the next byte to be written to the Byte Out register. When A_3=l, the `END` message is sent *true* once the `EOS` message is transmitted.

The END status bit is set when the `END` message or the `EOS` message is received. The EOI bit indicates which message has arrived. For example, `EOI=1` is the `END` message, and `EOI=0` is the `EOS` message.

Stopping a data transfer

The Controller-In-Charge uses three auxiliary commands to stop data transfer. Use these commands when the TLC is in the Controller Standby state (CSBS).

- `tca` (take control asynchronously)
- `tcs` (take control synchronously)
- `tcse` (take control synchronously on end)

`tca` (take control asynchronously)

When you issue tca, ATN is asserted and data transfer stops immediately. Data corruption or loss may occur if a talker/listener is in the process of transferring a data byte.

`tcs` (take control synchronously)

When you issue tcs, ATN is asserted and data transfer stops at the end of the handshake — Accept Not Ready state (ANRS). This command ensures that no data is lost or corrupted.

`tcse` (take control synchronously on end)

When you issue tcse, ATN is sent *true* at the end of the current data block and data transfer stops.

Specifying the controller as listener

To specify the system controller as listener, issue the auxiliary command `ltn` (Listen) or `ltnc` (Listen with Continuous mode) before issuing the auxiliary command `gts` (Go to Standby).

- When you issue the ltn, the listener operates in the mode specified by the A_0 and A_1 bits.
- When you issue ltnc, the listener operates in continuous mode regardless of the A_0 and A_1 bit values.

Measurement Computing Corporation
10 Commerce Way
Suite 1008
Norton, Massachusetts 02766
(508) 946-5100
Fax: (508) 946-9500
E-mail: info@mccdaq.com
www.mccdaq.com