8291A

GPIB Talker/Listener

The 8291A is an enhanced version of the 8291 GPIB Talker/Listener designed to interface microprocessors to an IEEE Standard 488 Instrumentation Interface Bus. It implements all of the Standard's interface functions except for the controller. The controller function can be added with the 8292 GPIB Controller, and the 8293 GPIB Transceiver performs the electrical interface for Talker/Listener and Talker/Listener/Controller configurations.

Rochester Electronics Manufactured Components

Rochester branded components are manufactured using either die/wafers purchased from the original suppliers or Rochester wafers recreated from the original IP. All recreations are done with the approval of the OCM.

Parts are tested using original factory test programs or Rochester developed test solutions to guarantee product meets or exceeds the OCM data sheet.

Quality Overview

- ISO-9001
- AS9120 certification
- Qualified Manufacturers List (QML) MIL-PRF-38535
    - Class Q Military
    - Class V Space Level
- Qualified Suppliers List of Distributors (QSLD)
    - Rochester is a critical supplier to DLA and meets all industry and DLA standards.

Rochester Electronics, LLC is committed to supplying products that satisfy customer expectations for quality and are equal to those originally supplied by industry manufacturers.

The original manufacturer's datasheet accompanying this document reflects the performance and specifications of the Rochester manufactured version of this device. Rochester Electronics guarantees the performance of its semiconductor products to the original OEM specifications. 'Typical' values are for reference purposes only. Certain minimum or maximum ratings may be based on product characterization, design, simulation, or sample testing.

intel®

8291A
GPIB TALKER/LISTENER

- Designed to Interface Microprocessors (e.g., 8048/49, 8051, 8080/85, 8086/88) to an IEEE Standard 488 Digital Interface Bus

- Programmable Data Transfer Rate

- Complete Source and Acceptor Handshake

- Complete Talker and Listener Functions with Extended Addressing

- Service Request, Parallel Poll, Device Clear, Device Trigger, Remote/Local Functions

- Selectable Interrupts

- On-Chip Primary and Secondary Address Recognition

- Automatic Handling of Addressing and Handshake Protocol

- Provision for Software Implementation of Additional Features

- 1–8 MHz Clock Range

- 16 Registers (8 Read, 8 Write), 2 for Data Transfer, the Rest for Interface Function Control, Status, etc.

- Directly Interfaces to External Non-Inverting Tranceivers for Connection to the GPIB

- Provides Three Addressing Modes, Allowing the Chip to be Addressed Either as a Major or a Minor Talker/Listener with Primary or Secondary Addressing

- DMA Handshake Provision Allows for Bus Transfers without CPU Intervention

- Trigger Output Pin

- On-Chip EOS (End of Sequence) Message Recognition Facilitates Handling of Multi-Byte Transfers

The 8291A is an enhanced version of the 8291 GPIB Talker/Listener designed to interface microprocessors to an IEEE Standard 488 Instrumentation Interface Bus. It implements all of the Standard's interface functions except for the controller. The controller function can be added with the 8292 GPIB Controller, and the 8293 GPIB Transceiver performs the electrical interface for Talker/Listener and Talker/Listener/Controller configurations.
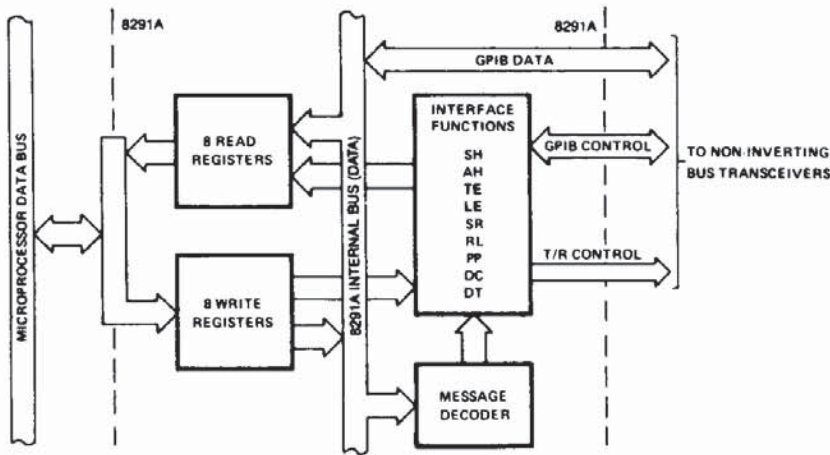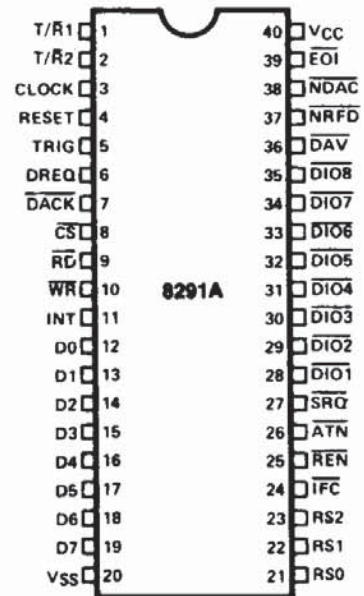

205248–1

Figure 1. Block Diagram

205248–2

Figure 2. Pin Configuration

8291A FEATURES AND IMPROVEMENTS

The 8291A is an improved design of the 8291 GPIB Talker/Listener. Most of the functions are identical to the 8291, and the pin configuration is unchanged.

The 8291A offers the following improvements to the 8291:

1. $\overline{EOI}$ is active with the data as a ninth data bit rather than as a control bit. This is to comply with some additions to the 1975 IEEE-488 Standard incorporated in the 1978 Standard.

2. The BO interrupt is not asserted until RFD is true. If the Controller asserts $\overline{ATN}$ synchronously, the data is guaranteed to be transmitted. If the Controller asserts $\overline{ATN}$ asynchronously, the SH (Source Handshake) will return to SIDS (Source Idle State), and the output data will be cleared. Then, if $\overline{ATN}$ is released while the 8291A is addressed to talk, a new BO interrupt will be generated. This change fixes 8291 problems which caused data to be lost or repeated and a problem with the RQS bit (sometimes cannot be asserted while talking).

3. LLOC and REMC interrupts are setting flipflops rather than toggling flipflops in the interrupt backup register. This ensures that the CPU knows that these state changes have occurred. The actual state can be determined by checking the LLO and REM status bits in the upper nibble of the Interrupt Status 2 Register.

4. DREQ is cleared by $\overline{DACK}$ ($\overline{RD}$ + $\overline{WR}$). DREQ on the 8291 was cleared only by $\overline{DACK}$ which is not compatible with the 8089 I/O Processor.

5. The INT bit in Interrupt Status 2 Register is duplicated in bit 7 of the Address 0 Register. If software polling is used to check for an interrupt, INT in the Address 0 Register should be polled rather than the Interrupt Status 2 Register. This ensures that no interrupts are lost due to asynchronous status reads and interrupts.

6. The 8291A's Send $\overline{EOI}$ Auxiliary Command works on any byte including the first byte of a message. The 8291 did not assert $\overline{EOI}$ after this command for a one byte message nor on two consecutive bytes.

7. To avoid confusion between holdoff on DAV versus RFD if a device is readdressed from a talker to a listener role or vice-versa during a holdoff, the "Holdoff on Source Handshake" has been eliminated. Only "Holdoff on Acceptor Handshake" is available.

8. The rsv local message is cleared automatically upon exit from SPAS if (APRS:STRS:SPAS) occurred. The automatic resetting of the bit after the serial poll is complete simplifies the service request software.

9. The SPASC interrupt on the 8291 has been replaced by the SPC (Serial Poll Complete) interrupt on the 8291A. SPC interrupt is set on exit from SPAS if APRS:STRS:SPAS occurred, indicating that the controller has read the bus status byte after the 8291A requested service. The SPASC interrupt was ambiguous because a controller could enter SPAS and exit SPAS generating two SPASC interrupts without reading the serial poll status byte. The SPC interrupt also simplifies the CPU's software by eliminating the interrupt when the serial poll is half way done.

10. The rtl Auxiliary Command in the 8291 has been replaced by Set and Clear rtl Commands in the 8291A. Using the new commands, the CPU has the flexibility to extend the length of local mode or leave it as a short pulse as in the 8291.

11. A holdoff RFD on GET, SDC, and DCL feature has been added to prevent additional bus activity while the CPU is responding to any of these commands. The feature is enabled by a new bit (B_4) in the Auxiliary Register B.

12. On the 8291, BO could cease to occur upon $\overline{IFC}$ going false if $\overline{IFC}$ occurred asynchronously. On the 8291A, BO continues to occur after $\overline{IFC}$ has gone false even if it arrived asynchronously.

13. User's software can distinguish between the 8291 and the 8291A as follows:

   a) pon (00H to register 5)

   b) RESET (02H to register 5)

   c) Read Interrupt Status 1 Register. If BO interrupt is set, the device is the 8291. If BO is clear, it is the 8291A.

This can be used to set a flag in the user's software which will permit special routines to be executed for each device. It could be included as part of a normal initialization procedure as the first step after a chip reset.

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
D_0–D_7	· 12–19	I/O	**DATA BUS PORT:** To be connected to microprocessor data bus.
RS_0–RS_2	21–23	I	**REGISTER SELECT:** Inputs, to be connected to three nonmultiplexed microprocessor address bus lines. Select which of the 8 internal read (write) registers will be read from (written into) with the execution of $\overline{RD}$ ($\overline{WR}$).
$\overline{CS}$	8	I	**CHIP SELECT:** When low, enables reading from or writing into the register selected by RS_0–RS_2.
$\overline{RD}$	9	I	**READ STROBE:** When low with $\overline{CS}$ or $\overline{DACK}$ low, selected register contents are read.
$\overline{WR}$	10	I	**WRITE STROBE:** When low with $\overline{CS}$ or $\overline{DACK}$ low, data is written into the selected register.
INT ($\overline{INT}$)	11	O	**INTERRUPT REQUEST:** To the microprocessor, set high for request and cleared when the appropriate register is accessed by the CPU. May be software configured to be active low.
DREQ	6	O	**DMA REQUEST:** Normally low, set high to indicate byte output or byte input in DMA mode; reset by $\overline{DACK}$.
$\overline{DACK}$	7	I	**DMA ACKNOWLEDGE:** When low, resets DREQ and selects data in/data out register for DMA data transfer (actual transfer done by $\overline{RD}/\overline{WR}$ pulse). Must be high if DMA is not used.
TRIG	5	O	**TRIGGER OUTPUT:** Normally low; generates a triggering pulse with 1 μsec min. width in response to the GET bus command or Trigger auxiliary command.
CLOCK	3	I	**EXTERNAL CLOCK:** Input, used only for T, delay generator. May be any speed in 1–8 MHz range.
RESET	4	I	**RESET INPUT:** When high, forces the device into an "idle" (initialization) mode. The device will remain at "idle" until released by the microprocessor, with the "Immediate Execute pon" local message.
$\overline{DIO}_1$–$\overline{DIO}_8$	28–35	I/O	**8-BIT GPIB DATA PORT:** Used for bidirectional data byte transfer between 8291A and GPIB via non-inverting external line transceivers.
$\overline{DAV}$	36	I/O	**DATA VALID:** GPIB handshake control line. Indicates the availability and validity of information on the $\overline{DIO}_1$–$\overline{DIO}_8$ and $\overline{EOI}$ lines.
$\overline{NRFD}$	37	I/O	**NOT READY FOR DATA:** GPIB handshake control line. Indicates the condition of readiness of device(s) connected to the bus to accept data.
$\overline{NDAC}$	38	I/O	**NOT DATA ACCEPTED:** GPIB handshake control line. Indicates the condition of acceptance of data by the device(s) connected to the bus.
$\overline{ATN}$	26	I	**ATTENTION:** GPIB command line. Specifies how data on $\overline{DIO}$ lines are to be interpreted.

3

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
$\overline{\text{IFC}}$	24	I	**INTERFACE CLEAR:** GPIB command line. Places the interface functions in a known quiescent state.
$\overline{\text{SRQ}}$	27	O	**SERVICE REQUEST:** GPIB command line. Indicates the need for attention and requests an interruption of the current sequence of events on the GPIB.
$\overline{\text{REN}}$	25	I	**REMOTE ENABLE:** GPIB command line. Selects (in conjunction with other messages) remote or local control of the device.
$\overline{\text{EOI}}$	39	I/O	**END OR IDENTITY:** GPIB command line. Indicates the end of a multiple byte transfer sequence or, in conjunction with $\overline{\text{ATN}}$, addresses the device during a polling sequence.
T/$\overline{\text{R}}$1	1	O	**EXTERNAL TRANSCEIVERS CONTROL LINE:** Set high to indicate output data/signals on the $\overline{\text{DIO}}_1$–$\overline{\text{DIO}}_8$ and $\overline{\text{DAV}}$ lines and input signals on the $\overline{\text{NRFD}}$ and $\overline{\text{NDAC}}$ lines (active source handshake). Set low to indicate input data/signals on the $\overline{\text{DIO}}_1$–$\overline{\text{DIO}}_8$ and $\overline{\text{DAV}}$ lines and output signals on the $\overline{\text{NRFD}}$ and $\overline{\text{NDAC}}$ lines (active acceptor handshake).
T/$\overline{\text{R}}$2	2	O	**EXTERNAL TRANSCEIVERS CONTROL LINE:** Set to indicate output signals on the $\overline{\text{EOI}}$ line. Set low to indicate expected input signal on the $\overline{\text{EOI}}$ line during parallel poll.
V$_{CC}$	40	P.S.	**POSITIVE POWER SUPPLY:** (5V ±10%).
GND	20	P.S.	**CIRCUIT GROUND POTENTIAL.**

NOTE:
All signals on the 8291A pins are specified with positive logic. However, IEEE 488 specifies negative logic on its 16 signal lines. Thus, the data is inverted once from D_0–D_7 to $\overline{\text{DIO}}_0$–$\overline{\text{DIO}}_8$ and non-inverting bus transceivers should be used.
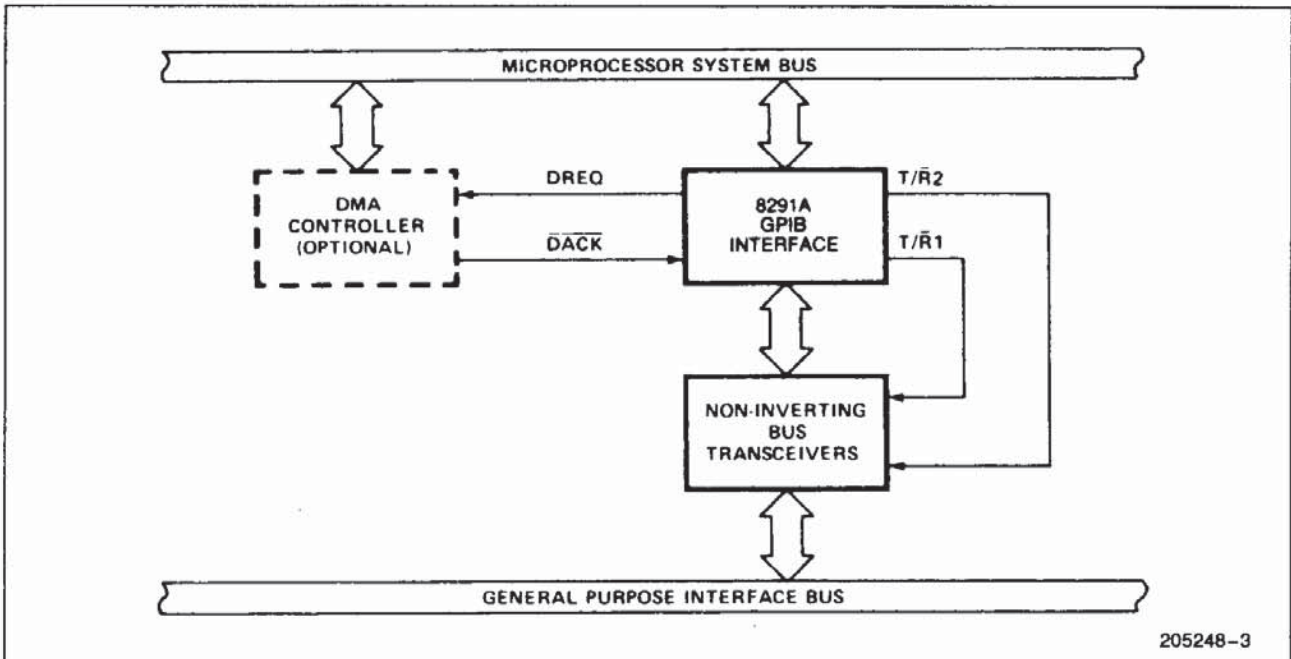

Figure 3. 8291A System Diagram

205248–3

THE GENERAL PURPOSE INTERFACE BUS (GPIB)

The General Purpose Interface Bus (GPIB) is defined in the IEEE Standard 488-1978 "Digital Interface for Programmable Instrumentation." Although a knowledge of this standard is assumed, Figure 4 provides the bus structure for quick reference. Also, Tables 2 and 3 reference the interface state mnemonics and the interface messages respectively. Modified state diagrams for the 8291A are presented in Appendix A.

General Description

The 8291A is a microprocessor-controlled device designed to interface microprocessors, e.g., 8048/49, 8051, 8080/85, 8086/88 to the GPIB. It implements all of the interface functions defined in the IEEE-488 Standard except for the controller function. If an implementation of the Standard's Controller is desired, it can be connected with an Intel® 8292 to form a complete interface.

The 8291A handles communication between a microprocessor-controlled device and the GPIB. Its capabilities include data transfer, handshake protocol, talker/listener addressing procedures, device clearing and triggering, service request, and both serial and parallel polling. In most procedures, it does not disturb the microprocessor unless a byte has arrived (input buffer full) or has to be sent out (output buffer empty).

The 8291A architecture includes 16 registers. Eight of these registers may be written into by the microprocessor. The other eight registers may be read by the microprocessor. One each of these read and write registers is for direct data transfers. The rest of the write registers control the various features of the chip, while the rest of the read registers provide the microprocessor with a monitor of GPIB states, various bus conditions, and device conditions.

GPIB Addressing

Each device connected to the GPIB must have at least one address whereby the controller device in charge of the bus can configure it to talk, listen, or send status. An 8291A implementation of the GPIB offers the user three alternative addressing modes for which the device can be initialized for each application. The first of these modes allows for the device to have two separate primary addresses. The second mode allows the user to implement a single talker/listener with a two byte address (primary address + secondary address). The third mode again allows for two distinct addresses but in this instance, they can each have a ten-bit address (5 low-order bits of each of two bytes). However, this mode requires that the secondary addresses be passed to the microprocessor for verification. These three addressing schemes are described in more detail in the discussion of the Address Registers.

Figure 4. Interface Capabilities and Bus Structure

Table 2. IEEE 488 Interface State Mnemonics

Mnemonic	State Represented
ACDS	Accept Data State
ACRS	Acceptor Ready State
AIDS	Acceptor Idle State
ANRS	Acceptor Not Ready State
APRS	Affirmative Poll Response State
AWNS	Acceptor Wait for New Cycle State
CACS	Controller Active State
CADS	Controller Addressed State
CAWS	Controller Active Wait State
CIDS	Controller Idle State
CPPS	Controller Parallell Poll State
CPWS	Controller Parallel Poll Wait State
CSBS	Controller Standby State
CSNS	Controller Service Not Requested State
CSRS	Controller Service Requested State
CSWS	Controller Synchronous Wait State
CTRS	Controller Transfer State
DCAS	Device Clear Active State
DCIS	Device Clear Idle State
DTAS	Device Trigger Active State
DTIS	Device Trigger Idle State
LACS	Listener Active State
LADS	Listener Addressed State
LIDS	Listener Idle State
LOCS	Local State
LPAS	Listener Primary Addressed State
LPIS	Listener Primary Idle State
LWLS	Local With Lockout State
NPRS	Negative Poll Response State
PACS	Parallel Poll Addressed to Configure State
PPAS	Parallel Poll Active State
PPIS	Parallel Poll Idle State

Mnemonic	State Represented
PPSS	Parallel Poll Standby State
PUCS	Parallel Poll Unaddressed to Configure State
REMS	Remote State
RWLS	Remote With Lockout State
SACS	System Control Active State
SDYS	Source Delay State
SGNS	Source Generate State
SIAS	System Control Interface Clear Active State
SIDS	Source Idle State
SIIS	System Control Interface Clear Idle State
SINS	System Control Interface Clear Not Active State
SIWS	Source Idle Wait State
SNAS	System Control Not Active State
SPAS	Serial Poll Active State
SPIS	Serial Poll Idle State
SPMS	Serial Poll Mode State
SRAS	System Control Remote Enable Active State
SRIS	System Control Remote Enable Idle State
SRNS	System Control Remote Enable Not Active State
SRQS	Service Request State
STRS	Source Transfer State
SWNS	Source Wait for New Cycle State
TACS	Talker Active State
TADS	Talker Addressed State
TIDS	Talker Idle State
TPIS	Talker Primary Idle State

The Controller function is implemented on the Intel® 8292.

Table 3. IEEE 488 Interface Message Reference List

Mnemonic	Message	Interface Function(s)
LOCAL MESSAGES RECEIVED (By Interface Functions)		
gts[1]	go to standby	C
ist	individual status	PP
lon	listen only	L, LE
lpe	local poll enable	PP
nba	new byte available	SH
pon	power on	SH, AH, T, TE, L, LE, SR, RL, PP, C
rdy	ready	AH
rpp[1]	request parallel poll	C
rsc[1]	request system control	C
rsv	request service	SR
rtl	return to local	RL
sic[1]	send interface clear	C
sre[1]	send remote enable	C
tca[1]	take control asynchronously	C

Table 3. IEEE 488 Interface Message Reference List (Continued)

Mnemonic	Message	Interface Function(s)
tcs[1]	take control synchronously	AH, C
ton	talk only	T, TE
REMOTE MESSAGES RECEIVED		
ATN	Attention	SH, AH, T, TE, L, LE, PP, C
DAB	Data Byte	(Via L, LE)
DAC	Data Accepted	SH
DAV	Data Valid	AH
DCL	Device Clear	DC
END	End	(via L, LE)
GET	Group Execute Trigger	DT
GTL	Go to Local	RL
IDY	Identify	L, LE, PP
IFC	Interface Clear	T, TE, L, LE, C
LLO	Local Lockout	RL
MLA	My Listen Address	L, LE, RL, T, TE
MSA	My Secondary Address	TE, LE, RL
MTA	My Talk Address	T, TE, L, LE
OSA	Other Secondary Address	TE
OTA	Other Talk Address	T, TE
PCG	Primary Command Group	TE, LE, PP
PPC[2]	Parallel Poll Configure	PP
[PPD][2]	Parallel Poll Disable	PP
[PPE][2]	Parallel Poll Enable	PP
PPR_N[1]	Parallel Poll Response N	(via C)
PPU[2]	Parallel Poll Unconfigure	PP
REN	Remote Enable	RL
RFD	Ready for Data	SH
RQS	Request Service	(via L, LE)
[SDC]	Select Device Clear	DC
SPD	Serial Poll Disable	T, TE
SPE	Serial Poll Enable	T, TE
SQR[1]	Service Request	(via C)
STB	Status Byte	(via L, LE)
TCT or [TCT][1]	Take Control	C
UNL	Unlisten	L, LE
REMOTE MESSAGES SENT		
ATN	Attentions	C
DAB	Data Byte	(Via T,TE)
DAC	Data Accepted	AH
DAV	Data Valid	SH
DCL	Device Clear	(via C)
END	End	(via T)
GET	Group Execute Trigger	(via C)
GTL	Go to Local	(via C)
IDY	Identify	C
IFC	Interface Clear	C
LLO	Local Lockout	(via C)
MLA or [MLA]	My Listen Address	(via C)
MSA or [MSA]	My Secondary Address	(via C)
MTA or [MTA]	My Talk Address	(via C)
OSA	Other Secondary Address	(via C)

3

Table 3. IEEE 488 Interface Message Reference List (Continued)

Mnemonic	Message	Interface Function(s)[3]
OTA	Other Talk Address	(via C)
PCG	Primary Command Group	(via C)
PPC	Parallel Poll Configure	(via C)
[PPD]	Parallel Poll Disable	(via C)
[PPE]	Parallel Poll Enable	(via C)
PPR_N	Parallel Poll Response N	PP
PPU	Parallel Poll Unconfigure	(via C)
REN	Remote Enable	C
RFD	Ready for Data	AH
RQS	Request Service	T, TE
[SDC]	Selected Device Clear	(via C)
SPD	Serial Poll Disable	(via C)
SPE	Serial Poll Enable	(via C)
SRQ	Service Request	SR
STB	Status Byte	(via T,TE)
TCT	Take Control	(via C)
UNL	Unlisten	(via C)

NOTES:
1. These messages are handled only by Intel's 8292.
2. Undefined commands which may be passed to the microprocessor.
3. All Controller messages must be sent via Intel's 8292.

8291A Registers

A bit-by-bit map of the 16 registers on the 8291A is presented in Figure 5. A more detailed explanation of each of these registers and their functions follows. The access of these registers by the microprocessor is accomplished by using the $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, and RS_0–RS_2 pins.

Register	$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	RS_0–RS_2
All Read Registers	0	0	1	CCC
All Write Registers	0	1	0	CCC
High Impedance	1	d	d	ddd

Data Registers

D17	D16	D15	D14	D13	D12	D11	D10

DATA-IN REGISTER (0R)

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

DATA-OUT REGISTER (0W)

The Data-In Register is used to move data from the GPIB to the microprocessor or to memory when the 8291A is addressed to listen. Incoming information is separately latched by this register, and its contents are not destroyed by a write to the data-out register. The RFD (Ready for Data) message is held false until the byte is removed from the data in register, either by the microprocessor or by DMA. The 8291A then completes the handshake automatically. In RFD holdoff mode (see Auxiliary Register A), the handshake is not finished until a command is sent

telling the 8291A to release the holdoff. In this way, the same byte may be read several times, or an over anxious talker may be held off until all available data has been processed.

When the 8291A is addressed to talk, it uses the data-out register to move data onto the GPIB. After the BO interrupt is received and a byte is written to this register, the 8291A initiates and completes the handshake while sending the byte out over the bus. In the BO interrupt disable mode, the user should wait until BO is active before writing to the register. (In the DMA mode, this will happen automatically.) A read of the Data-In Register does not destroy the information in the Data-Out Register.

Interrupt Registers

CPT	APT	GET	END	DEC	ERR	BO	BI

INTERRUPT STATUS 1 (1R)

INT	SPAS	LLO	REM	SPC	LLOC	REMC	ADSC

INTERRUPT STATUS 2 (2R)

CPT	APT	GET	END	DEC	ERR	BO	BI

INTERRUPT ENABLE 1 (1W)

0	0	DMAO	DMAI	SPC	LLOC	REMC	ADSC

INTERRUPT ENABLE 2 (2W)

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

ADDRESS 0 REGISTER

Figure 5. 8291A Registers

READ REGISTERS								RS2	RS1	RS0	WRITE REGISTERS							
D17	D16	D15	D14	D13	D12	D11	D10	0	0	0	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
DATA IN											DATA OUT							
CPT	APT	GET	END	DEC	ERR	BO	BI	0	0	1	CPT	APT	GET	END	DEC	ERR	BO	BI
INTERRUPT STATUS 1											INTERRUPT ENABLE 1							
INT	SPAS	LLO	REM	SPC	LLOC	REMC	ADSC	0	1	0	0	0	DMAO	DMAI	SPC	LLOC	REMC	ADSC
INTERRUPT STATUS 2											INTERRUPT ENABLE 2							
S8	SEQS	S6	S5	S4	S3	S2	S1	0	1	1	S8	rsv	S6	S5	S4	S3	S2	S1
SERIAL POLL STATUS											SERIAL POLL MODE							
ton	lon	EOI	LPAS	TPAS	LA	TA	MJMN	1	0	0	TO	LO	0	0	0	0	ADM1	ADM0
ADDRESS STATUS											ADDRESS MODE							
CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0	1	0	1	CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0
COMMAND PASS THROUGH											AUX MODE							
INT	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0	1	1	0	ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
ADDRESS 0											ADDRESS 0/1							
X	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1	1	1	1	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0
ADDRESS 1											EOS							

REGISTER SELECT CODE

3

The 8291A can be configured to generate an interrupt to the microprocessor upon the occurrence of any of 12 conditions or events on the GPIB. Upon receipt of an interrupt, the microprocessor must read the Interrupt Status Registers to determine which event has occurred, and then execute the appropriate service routine (if necessary). Each of the 12 interrupt status bits has a matching enable bit in the interrupt enable registers. These enable bits are used to select the events that will cause the INT pin to be asserted. Writing a logic "1" into any of these bits enables the corresponding interrupt status bits to generate an interrupt. Bits in the Interrupt Status Registers are set regardless of the states of the enable bits. The Interrupt Status Registers are then cleared upon being read or when a local pon (power-on) message is executed. If an event occurs while one of the Interrupt Status Registers is being read, the event is held until after its register is cleared and then placed in the register.

NOTE:
Reading the interrupt status registers clears the bits which were set. The software must examine *all* relevant bits in the interrupt status registers before disregarding the value or an important interrupt may be missed.

The mnemonics for each of the bits in these registers and a brief description of their respective functions appears in Table 4. This table also indicates how each of the interrupt bits is set.

NOTE:
The INT bit in the Address 0 Register is a duplicate of the INT bit in the Interrupt Status 2 Register. It is only a status bit. It does not generate interrupts and thus does not have a corresponding enable bit.

The BO and BI interrupts enable the user to perform data transfer cycles. BO indicates that a data byte should be written to the Data Out Register. It is set by TACS • (SWNS + SGNS) • RFD. It is reset when the data byte is written, ATN is asserted, or the 8291A exits TACS. Data should never be written to the Data Out Register before BO is set. Similarly, BI is set when an input byte is accepted into the 8291A and reset when the microprocessor reads the Data In Register. BO and BI are also reset by pon (power-on local message) and by a read of the Interrupt Status 1 Register. However, if it is so desired, data transfer cycles may be performed without reading the Interrupt Status 1 Register if all interrupts except for BO or BI are disabled; BO and BI will automatically reset after each byte is transferred.

Table 4. Interrupt Bits

Indicates Undefined Commands	CPT	An undefined command has been received.
Set by (TPAS + LPAS)•SCG•ACDS•MODE 3	APT	A secondary address must be passed through to the microprocessor for recognition.
Set by DTAS	GET	A group execute trigger has occurred.
Set by (EOS + EOI)•LACS	END	An EOS or EOI message has been received.
Set by DCAS	DEC	Device Clear Active State has occurred.
Set by TACS•nba•DAC•RFD	ERR	Interface error has occurred; no listeners are active.
TACS•(SWNS + SGNS)	BO	A byte should be output.
Set by LACS•ACDS	BI	A byte has been input.

Shows status of the INT pin	INT	
The device has been enabled for a serial poll	SPAS	These are status only. They will not generate interrupts, nor do they have corresponding mask bits.
The device is in local lock out state. (LWLS+RWLS)	LLO	
The device is in a remote state. (REMS+RWLS)	REM	

SPAS →$\overline{SPAS}$ if APRS:STRS:SPAS was true	SPC	Serial Poll Complete interrupt.
LLO ⤸ NO LLO	LLOC	Local lock out change interrupt.
Remote ⤸ Local	REMC	Remote/Local change interrupt.
Addressed ⤸ Unaddressed	ADSC	Address status change interrupt.[1]

205248–24

NOTE:
1. In ton (talk-only) and lon (listen-only) modes, no ADSC interrupt is generated.

If the 8291A is used in the interrupt mode, the INT and DREQ pins can be dedicated to data input and output interrupts respectively by enabling BI and DMAO, provided that no other interrupts are enabled. This eliminates the need to read the interrupt status registers if a byte is received or transmitted.

The ERR bit is set to indicate the bus error condition when the 8291A is an active talker and tries to send a byte to the GPIB, but there are no active listeners (e.g., all devices on the GPIB are in AIDS). The logical equivalent of (nba • TACS • DAC • RFD) will set this bit.

The DEC bit is set whenever DCAS has occurred. The user must define a known state to which all device functions will return in DCAS. Typically this state will be a power-on state. However, the state of the device functions at DCAS is at the designer's discretion. It should be noted that DCAS has no effect on the interface functions which are returned to a known state by the IFC (interface clear) message or the pon local message.

The END interrupt bit may be used by the microprocessor to detect that a multi-byte transfer has been completed. The bit will be set when the 8291A is an active listener (LACS) and either EOS (provided the End on EOS Received feature is enabled in the Auxiliary Register A) or EOI is received. EOS will generate an interrupt when the byte in the Data In Register matches the byte in the EOS register. Otherwise the interrupt will be generated when a true input is detected on EOI.

The GET interrupt bit is used by the microprocessor to detect that DTAS has occurred. It is set by the 8291A when the GET message is received while it is addressed to listen. The TRIG output pin of the 8291A fires when the GET message is received. Thus, the basic operation of device trigger may be started without microprocessor software intervention.

The APT interrupt bit indicates to the processor that a secondary address is available in the CPT register for validation. This interrupt will only occur if Mode 3 addressing is in effect. (Refer to the section on addressing.) In Mode 2, secondary addresses will be recognized automatically on the 8291A. They will be ignored in Mode 1.

The CPT interrupt bit flags the occurrence of an undefined command and of all secondary commands following an undefined command. The Command Pass Through feature is enabled by the B0 bit of Auxiliary Register B. Any message not decoded by the 8291A (not included in the state diagrams in Appendix B) becomes an undefined command. Note that any addressed command is automatically ignored when the 8291A is not addressed.

Undefined commands are read by the CPU from the Command Pass Through register of the 8291A. This register reflects the logic levels present on the data lines at the time it is read. If the CPT feature is enabled, the 8291A will hold off the handshake until this register is read.

An especially useful feature of the 8291A is its ability to generate interrupts from state transitions in the interface functions. In particular, the lower 3 bits of the Interrupt Status 2 Register, if enabled by the corresponding enable bits, will cause an interrupt upon changes in the following states as defined in the IEEE 488 Standard.

Bit 0   ADSC   change in LIDS or TIDS or MJMN
Bit 1   REMC   change in LOCS or REMS
Bit 2   LLOC   change in LWLS or RWLS

The upper 4 bits of the Interrupt Status 2 Register are available to the processor as status bits. Thus, if one of the bits 0–2 generates an interrupt indicating a state change has taken place, the corresponding status bit (bits 3–5) may be read to determine what the new state is. To determine the nature of a change in addressed status (bit 0) the Address Status Register is available to be read. The SPC interrupt (bit 3 in Interrupt Status 2) is set upon exit from SPAS if APRS:STRS:SPAS occurred which indicates that the GPIB controller has read the bus serial poll status byte after the 8291A requested service (asserted SRQ). The SPC interrupt occurs once after the controller reads the status byte if service was requested. The controller may read the status byte later, and the byte will contain the last status the 8291A's CPU wrote to the Serial Poll Mode Register, but the SRQS bit will not be set and no interrupt will be generated. Finally, bit 7 monitors the state of the 8291A INT pin. Logically, it is an OR of all enabled interrupt status bits. One should note that bits 3–6 of the Interrupt Status 2 Register do not generate interrupts, but are available only to be read as status bits by the processor. Bit 7 in Interrupt Status 2 is duplicated in Address 0 Register, and the latter should be used when polling for interrupts to avoid losing one of the interrupts in Interrupt Status 2 Register.

Bits 4 and 5 (DMAI, DMAO) of the Interrupt Mask 2 Register are available to enable direct data transfers between memory and the GPIB; DMAI (DMA in) enables the DREQ (DMA request) pin of the 8291A to be asserted upon the occurrence of BI. Similarly, DMAO (DMA out) enables the DREQ pin to be asserted upon the occurrence of BO. One might note that the DREQ pin may be used as a second interrupt output pin, monitoring BI and/or BO and enabled by DMAI and DMAO. One should note that the DREQ pin is not affected by a read of the Interrupt Status 1 Register. It is reset whenever a byte is written to the Data Out Register or read from the Data In Register.

To ensure that an interrupt status bit will not be cleared without being read, and will not remain uncleared after being read, the 8291A implements a special interrupt handling procedure. When an enabled interrupt bit is set in either of the Interrupt Status Registers, the input of the registers are blocked until the set bit is read and reset by the microprocessor. Thus, potential problems arise when interrupt status changes while the register is being blocked. However, the 8291A stores all new interrupts in a temporary register and transfers them to the appropriate Interrupt Status Register after the interrupt has been reset. This transfer takes place only if the corresponding bits were read as zeroes.

Serial Poll Registers

S8	SRQS	S6	S5	S4	S3	S2	S1

SERIAL POLL STATUS (3R)

S8	rsv	S6	S5	S4	S3	S2	S1
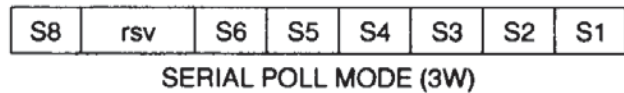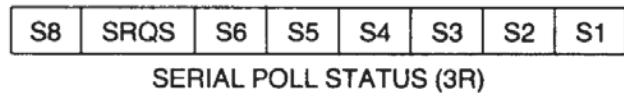
SERIAL POLL MODE (3W)

The Serial Poll Mode Register determines the status byte that the 8291A sends out on the GPIB data lines when it receives the SPE (Serial Poll Enable) message. Bit 6 of this register is reserved for the rsv (request service) local message. Setting this bit to 1 causes the 8291A to assert its $\overline{SRQ}$ line, indicating its need for attention from the controller-in-charge of the GPIB. The other bits of this register are available for sending status information over the GPIB. Sometime after the microprocessor initiates a request for service by setting bit 6, the controller of the GPIB sends the SPE message and then addresses the 8291A to talk. At this point, one byte of status is returned by the 8291A via the Serial Poll Mode Register. After the status byte is read by the controller, rsv is automatically cleared by the 8291A and an SPC interrupt is generated. The CPU may request service again by writing another byte to the Serial Poll Mode Register with the rsv bit set. If the control-

ler performs a serial poll when the rsv bit is clear, the last status byte written will be read, but the $\overline{SRQ}$ line will not be driven by the 8291A and the SRQS bit will be clear in the status byte.

The Serial Poll Status Register is available for reading the status byte in the Serial Poll Mode Register. The processor may check the status of a request for service by polling bit 6 of this register, which corresponds to SRQS (Service Request State). When a Serial Poll is conducted and the controller-in-charge reads the status byte, the SRQS bit is cleared. The $\overline{SRQ}$ line and the rsv bit are tied together.

Address Registers

ton	lon	EOI	LPAS	TPAS	LA	TA	MJMN

ADDRESS STATUS (4R)

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

ADDRESS 0 (6R)

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

ADDRESS 1 (7R)

TO	LO	0	0	0	0	ADM1	ADM0

ADDRESS MODE (4W)

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1

ADDRESS 0/1 (6W)

The Address Mode Register is used to select one of the five modes of addressing available on the 8291A. It determines the way in which the 8291A uses the information in the Address 0 and Address 1 Registers.

—In Mode 1, the contents of the Address 0 Register constitute the "Major" talker/listener address while the Address 1 Register represents the "Minor" talker/listener address. In applications where only one address is needed, the major talker/listener is used, and the minor talker/listener should be disabled. Loading an address via the Address 0/1 Register into Address Registers 0 and 1 enables the major and minor talker/listener functions respectively.

—In Mode 2 the 8291A recognizes two sequential address bytes: a primary followed by a secondary. Both address bytes must be received in order to enable the device to talk or listen. In this manner, Mode 2 addressing implements the extended talker and listener functions as defined in IEEE-488.

To use Mode 2 addressing the primary address must be loaded into the Address 0 Register, and the Secondary Address is placed in the Address 1 Register. With both primary and secondary addresses residing on chip, the 8291A can handle all addressing sequences without processor intervention.

—In Mode 3, the 8291A handles addressing just as it does in Mode 1, except that each Major or Minor primary address must be followed by a secondary address. All secondary addresses must be verified by the microprocessor when Mode 3 is used. When the 8291A is in TPAS or LPAS (talker/listener primary addressed state), and it does not recognize the byte on the DIO lines, an APT interrupt is generated (see section on Interrupt Registers) and the byte is available in the CPT (Command Pass-Through) Register. As part of its interrupt service routine, the microprocessor must read the CPT Register and write one of the following responses to the Auxiliary Mode Register:

1. 07H implies a non-valid secondary address
2. 0FH implies a valid secondary address

Setting the TO bit generates the local ton (talk-only) message and sets the 8291A to a talk-only mode. This mode allows the device to operate as a talker in an interface system without a controller.

Setting the LO bit generates the local lon (listen-only) message and sets the 8291A to a listen-only mode. This mode allows the device to operate as a listener in an interface system without a controller. The above bits may also be used by a controller-in-charge to set itself up for remote command or data communication.

The mode of addressing implemented by the 8291A may be selected by writing one of the following bytes to the Address Mode Register.

Register Contents	Mode
10000000	Enable talk only mode (ton)
01000000	Enable listen only mode (lon)
11000000	The 8291 may talk to itself
00000001	Mode 1, (Primary-Primary)
00000010	Mode 2 (Primary-Secondary)
00000011	Mode 3 (Primary/APT-Primary/APT)

The Address Status Register contains information used by the microprocessor to handle its own addressing. This information includes status bits that monitor the address state of each talker/listener, "ton" and "lon" flags which indicate the talk and listen only states, and an EOI bit which, when set, signifies that the END message came with the last data byte. LPAS and TPAS indicate that the listener

or talker primary address has been received. The microprocessor can use these bits when the secondary address is passed through to determine whether the 8291A is addressed to talk or listen. The LA (listener addressed) bit will be set when the 8219A is in LACS (Listener Active State) or in LADS (Listener Addressed State). Similarly, the TA (Talker Addressed bit) will be set to indicate TACS or TADS, but also to indicate SPAS (Serial Poll Active State). The MJMN bit is used to determine whether the information in the other bits applies to the Major or Minor talker/listener. It is set to "1" when the Minor talker/listener is addressed. It should be noted that only one talker/listener may be active at any one time. Thus, the MJMN bit will indicate which, if either, of the talker/listeners is addressed or active.

The Address 0/1 Register is used for specifying the device's addresses according to the format selected in the Address Mode Register. Five bit addresses may be loaded into the Address 0 and Address 1 Registers by writing into the Address 0/1 Register. The ARS bit is used to select which of these registers the other seven bits will be loaded into. The DT and DL bits may be used to disable the talker or listener function at the address signified by the other five bits. When Mode 1 addressing is used and only one primary address is desired, *both* the talker and the listener should be disabled at the Minor address.

As an example of how the Address 0/1 Register might be used, consider an example where two primary addresses are needed in the device. The Major primary address will be selectable only as a talker and the Minor primary address will be selectable only as a listener. This configuration of the 8291A is formed by the following sequence of writes by the microprocessor.

Operation	CS	RD	WR	Data	RS₂–RS₀
1. Select addressing Mode 1	0	1	0	00000001	100
2. Load major address into Address 0 Register with listener function disabled.	0	1	0	001AAAAA	110
3. Load minor address into Address 1 Register with talker function disabled.	0	1	0	110BBBBB	110

At this point, the addresses AAAAA and BBBBB are stored in the Address 0 and Address 1 Registers respectively, and are available to be read by the microprocessor. Thus, it is not necessary to store any address information elsewhere. Also, with the information stored in the Address 0 and Address 1 Registers, processor intervention is not required to recognize addressing by the controller. Only in Mode 3, where secondary addresses are passed through, must the processor intervene in the addressing sequence.

The Address 0 Register contains a copy of bit 7 of the Interrupt Status 2 Register (INT). This is to be used when polling for interrupts. Software should poll register 6 checking for INT (bit 7) to be set. When INT is set, the Interrupt Status Register should be read to determine which interrupt was received.

Command Pass Through Register

CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0

COMMAND PASS THROUGH (5R)

The Command Pass Through Register is used to transfer undefined 8-bit remote message codes from the GPIB to the microprocessor. When the CPT feature is enabled (bit BO in Auxiliary Register B), any message not decoded by the 8291A becomes an undefined command. When Mode 3 addressing is used secondary addresses are also passed through the CPT Register. In either case, the 8291A will hold-off the handshake until the microprocessor reads this register and issues the VSCMD auxiliary command.

The CPT and APT interrupts flag the availability of undefined commands and secondary addresses in the CPT Register. The details of these interrupts are explained in the section on Interrupt Registers.

An added feature of the 8291A is its ability to handle undefined secondary commands following undefined primaries. Thus, the number of available commands for future IEEE-488 definition is increased; one undefined primary command followed by a sequence of as many as 32 secondary commands can be processed. The IEEE-488 Standard does not permit users to define their own commands, but upgrades of the standard are thus provided for.

The recommended use of the 8291A's undefined command capabilities is for a controller-configured Parallel Poll. The PPC message is an undefined primary command typically followed by PPE, and undefined secondary command. For details on this procedure, refer to the section on Parallel Poll Protocol.

Auxiliary Mode Register

CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0

AUX MODE (5W)
CNT0–2:CONTROL BITS
COM0–4:COMMAND BITS

The Auxiliary Mode Register contains a three-bit control field and a five-bit command field. It is used for several purposes on the 8291A:

1. To load "hidden" auxiliary registers on the 8291A.

2. To issue commands from the microprocessor to the 8291A.

3. To preset an internal counter used to generate T1, delay in the Source Handshake function, as defined in IEEE-488.

Table 5 summarizes how these tasks are performed with the Auxiliary Mode Register. Note that the three control bits determine how the five command bits are interpreted.

Table 5

Code		Command
Control Bits	Command Bits	
000 001	0CCCC 0DDDD	Execute auxiliary command CCCC Preset internal counter to match external clock frequency of DDDD MHz (DDDD binary representation of 1 to 8 MHz)
100	DDDDD	Write DDDDD into auxiliary register A
101	DDDDD	Write DDDDD into auxiliary register B
011	USP$_3$P$_2$P$_1$	Enable/disable parallel poll either in response to remote messages (PPC followed by PPE or PPD) or as a local lpe message. (Enable if U = 0, disable if U = 1.)

AUXILIARY COMMANDS

Auxiliary commands are executed by the 8291A whenever 0000CCCC is written into the Auxiliary Mode Register, where CCCC is the 4-bit command code.

0000—Immediate Execute pon: This command resets the 8291A to a power up state (local pon message as defined in IEEE-488).

The following conditions constitute the power up state:

1. All talkers and listeners are disabled.
2. No interrupt status bits are set.

The 8291A is designed to power up in certain states as specified in the IEEE-488 state diagrams. Thus, the following states are in effect in the power up state: SIDS, AIDS, TIDS, LIDS, NPRS, LOCS, and PPIS.

The "0000" pon is an immediate execute command (a pon pulse). It is also used to release the "initialize" state generated by either an external reset pulse or the "0010" Chip Reset command.

0010—Chip Reset (Initialize): This command has the same effect as a pulse applied to the Reset pin. (Refer to the section on Reset Procedure.)

0011—Finish Handshake: This command finishes a handshake that was stopped because of a holdoff on RFD. (Refer to Auxiliary Register A.)

0100—Trigger: A "Group Execute Trigger" is forced by this command. It has the same effect as a GET command issued by the controller-in-charge of the GPIB, but does not cause a GET interrupt.

0101, 1101—Clear/Set rtl: These commands correspond to the local rtl message as defined by the IEEE-488. The 8291A will go into local mode when a Set rtl Auxiliary Command is received if local lockout is not in effect. The 8291A will exit local mode after receiving a Clear rtl Auxiliary Command if the 8291A is addressed to listen.

0110—Send EOI: The EOI line of the 8291A may be asserted with this command. The command causes EOI to go true with the next byte transmitted. The EOI line is then cleared upon completion of the handshake for that byte.

0111, 1111—Non Valid/Valid Secondary Address or Command (VSCMD): This command informs the 8291A that the secondary address received by the microprocessor was valid or invalid (0111 = invalid, 1111 = valid). If Mode 3 addressing is used, the processor must field each extended address and respond to it, or the GPIB will hang up. Note that the COM3 bit is the invalid/valid flag.

The valid (1111) command is also used to tell the 8291A to continue from the command-pass-through-state, or from RFD holdoff on GET, SDC or DCL.

1000—pon: This command puts the 8291A into the pon (power on) state and holds it there. It is similar to a Chip Reset except none of the Auxiliary Mode Registers are cleared. In this state, the 8291A does not participate in any bus activity. An Immediate Execute pon releases the 8291A from the pon state and permits the device to participate in the bus activity again.

0001, 1001—Parallel Poll Flag (local "ist" message): This command sets (1001) or clears (0001) the parallel poll flag. A "1" is sent over the assigned data line (PRR = Parallel Poll Response true) only if the parallel poll flag matches the sense bit from the lpe local message (or indirectly from the PPE message). For a more complete description of the Parallel Poll features and procedures refer to the section on Parallel Poll Protocol.

INTERNAL COUNTER

The internal counter determines the delay time allowed for the setting of data on the DIO lines. This delay time is defined as T_1 in IEEE-488 and appears in the Source Handshake state diagram between the SDYS and STRS. As such, DAV is asserted T_1 after the DIO lines are driven. Consequently , T_1 is a major factor in determining the data transfer rate of the 8291A over the GPIB (T_1 = TWRDV2-TWRD15).

When open-collector transceivers are used for connection to the GPIB, T_1 is defined by IEEE-488 to be 2 μs. By writing 0010DDDD into the Auxiliary Mode Register, the counter is preset to match a f_C MHz clock input, where DDDD is the binary representation of N_F [$1 \leq N_F \leq 8$, N_F = (DDDD)$_2$]. When N_F = f_C, a 2 μs T_1 delay will be generated before each DAV asserted.

$$T_{1(\mu s)} = \frac{2N_F}{f_c} + t_{SYNC}, 1 \leq N_F \leq 8$$

t_{SYNC} is a synchronization error, greater than zero and smaller than the larger of T clock high and T clock low. (For a 50% duty cycle clock, t_{SYNC} is less than half the clock cycle).

If it is necessary that T_1 be different from 2 μs, N_F may be set to a value other than f_C. In this manner, data transfer rates may be programmed for a given system. In small systems, for example, where transfer rates exceeding GPIB specifications are required, one may set $N_F < f_C$ and decrease T_1.

When tri-state transceivers are used, IEEE-488 allows a higher transfer rate (lower T_1). Use of the 8291A with such transceivers is enabled by setting B_2 in Auxilliary Register B. In this case, setting N_F = f_c causes a T_1 delay of 2μs to be generated for the first byte transmitted—all subsequent bytes will have a delay of 500 ns.

$$T_1 \text{ (High Speed) } \mu s = \frac{N_F}{2f_C} + t_{SYNC}$$

Thus, the shortest T_1 is achieved by setting N_F = 1 using an 8 MHz clock with a 50% duty cycle clock ($t_{SYNC} < 63$ ns):

$$T_{1(HS)} = \frac{1}{2 \times 8} + 0.063 = 125 \text{ ns max.}$$

AUXILIARY REGISTER A

Auxiliary Register A is a "hidden" 5-bit register which is used to enable some of the 8291A features. Whenever a 100 $A_4A_3A_2A_1A_0$ byte is written into the Auxiliary Register, it is loaded with the data $A_4A_3A_2A_1A_0$. Setting the respective bits to "1" enables the following features.

A_0—RFD Holdoff on all Data: If the 8291A is listening, RFD will not be sent true until the "finish handshake" auxiliary command is issued by the microprocessor. The holdoff will be in effect for each data byte.

A_1—RFD Holdoff on End: This feature enables the holdoff on EOI or EOS (if enabled). However, no hold-off will be in effect on any other data bytes.

A_2—End on EOS Received: Whenever the byte in the Data In Register matches the byte in the EOS Register, the END interrupt bit will be set in the Interrupt Status 1 Register.

A_3—Output EOI on EOS Sent: Any occurrence of data in the Data Out Register matching the EOS Register causes the EOI line to be sent true along with the data.

A_4—EOS Binary Compare: Setting this bit causes the EOS Register to function as a full 8-bit word. When it is not set, the EOS Register is a 7-bit word (for ASCII characters).

If A_0 = A_1 = 1, a special "continuous Acceptor Handshake cycling" mode is enabled. This mode should be used only in a controller system configuration, where both the 8291A and the 8292 are used. It provides a continuous cycling through the Acceptor Handshake state diagram, requiring no local messages from the microprocessor; the rdy local message is automatically generated when in ANRS. As such, the 8291A Acceptor Handshake serves as the controller Acceptor Handshake. Thus, the controller cycles through the Acceptor Handshake without delaying the data transfer in progress. When the tcs local message is executed, the 8291A should be taken out of the "continuous AH cycling" mode, the GPIB will hang up in ANRS, and a BI interrupt will be generated to indicate that control may be taken. A

simpler procedure may be used when a "tcs on end of block" is executed; the 8291A may stay in "continuous AH cycling". Upon the end of a block (EOI or EOS received), a holdoff is generated, the GPIB hangs up in ANRS, and control may be taken.

AUXILIARY REGISTER B

Auxiliary Register B is a "hidden" 4-bit register which is used to enable some of the features of the 8291A. Whenever a 101 $B_4B_3B_2B_1B_0$ is written into the Auxiliary Mode Register, it is loaded with the data $B_4B_3B_2B_1B_0$. Setting the respective bits to "1" enables the following features:

B_0—Enable Undefined Command Pass Through: This feature allows any commands not recognized by the 8291A to be handled in software. If enabled, this feature will cause the 8291A to holdoff the handshake when an undefined command is received. The microprocessor must then read the command from the Command Pass Through Register and send the VSCMD auxiliary command. Until the VSCMD command is sent, the handshake holdoff will be in effect.

B_1—Send EOI in SPAS: This bit enables EOI to be sent with the status byte; EOI is sent true in Serial Poll Active State. Otherwise, EOI is sent false in SPAS.

B_2—Enable High Speed Data Transfer: This feature may be enabled when tri-state external transceivers are used. The data transfer rate is limited by T_1 delay time generated in the Source Handshake function, which is defined according to the type of transceivers used. When the "High Speed" feature is enabled, $T_1 = 2$ microseconds is generated for the first byte transmitted after each true to false transition of ATN. For all subsequent bytes, $T_1 = 500$ ns. Refer to the Internal Counter section for an explanation of T_1 duration as a function of B_2 and of clock frequency.

B_3—Enable Active Low Interrupt: Setting this bit causes the polarity of the INT pin to be reversed, providing an output signal compatible with Intel's MCS-48® Family. Interrupt registers are not affected by this bit.

B_4—Enable RFD Holdoff on GET or DEC: Setting this bit causes RFD to be held false until the "VSCMD" auxiliary command is written after GET, SDC, and DCL commands. This allows the device to hold off the bus until it has completed a clear or trigger similar to an unrecognized command.

PARALLEL POLL PROTOCOL

Writing a 011$USP_3P_2P_1$ into the Auxiliary Mode Register will enable (U = 0) or disable (U = 1) the 8291A for a parallel poll. When U = 0, this command is the "lpe" (local poll enable) local message as defined in IEEE-488. The "S" bit is the sense in which the 8291A is enabled; only if the Parallel Poll Flag ("ist" local message) matches this bit will the Parallel Poll Response, PPR_N, be sent true (Response = S + ist). The bits $P_3P_2P_1$ specify which of the eight data lines PPR_N will be sent over. Thus, once the 8291A has been configured for Parallel Poll, whenever it senses both EOI and ATN true, it will automatically compare its PP flag with the sense bit and send PPR_N true or false according to the comparison.

If a PP2* implementation is desired, the "lpe" and "jst" local messages are all that are needed. Typically, the user will configure the 8291A for Parallel Poll immediately after initialization. During normal operation the microprocessor will set or clear the Parallel Poll Flag (ist) according to the device's need for service. Consequently the 8291A will be set up to give the proper response to IDY (EOI • ATN) without directly involving the microprocessor.

If a PP1* implementation is desired, the undefined command features of the 8291A must be used. In PP1, the 8291A is indirectly configured for Parallel Poll by the active controller on the GPIB. The sequence at the 8291A being enabled or disabled remotely is as follows:

1. The PPC message is received and is loaded into the Command Pass Through Register as an undefined command. A CPT Interrupt is sent to the microprocessor; the handshake is automatically held off.

2. The microprocessor reads the CPT Register and sends VSCMD to the 8291A, releasing the handshake.

3. Having recieved an undefined primary command, the 8291A is set up to receive an undefined secondary command (the PPE or PPD message). This message is also received into the CPT Register, the handshake is held off, and the CPT interrupt is generated.

4. The microprocessor reads the PPE or PPD message and writes the command into the Auxiliary Mode Register (bit 7 should be cleared first). Finally, the microprocessor sends VSCMD and the handshake is released.

NOTE:
*As defined in IEEE Standard 488.

End of Sequence (EOS) Register

EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0

EOS REGISTER

The EOS Register and its features offer an alternative to the "Send EOI" auxiliary command. A seven or eight bit byte (ASCII or binary) may be placed in the register to flag the end of a block or read. The type of EOS byte to be used is selected in Auxiliary Register bit A_4.

If the 8291A is a listener, and the "End on EOS Received" is enabled with bit A_2, then an END interrupt is generated in the Interrupt Status 1 Register whenever the byte in the Data-In Register matches the byte in the EOS Register.

If the 8291A is a talker, and the "Output EOI on EOS Sent" is enabled with bit A_3, then the EOI line is sent true with the next byte whenever the contents of the Data Out Register match the EOS register.

Reset Procedure

The 8291A is reset to an initialization state either by a pulse applied to its Reset pin, or by a reset auxiliary command (02H written into the Auxiliary Command Register). The following conditions are caused by a reset pulse (or local reset command):

1. A "pon" local message as defined by IEEE-488 is held true until the initialization state is released.

2. The Interrupt Status Registers are cleared (not Interrupt Enable Registers).

3. Auxiliary Registers A and B are cleared.

4. The Serial Poll Mode Register is cleared.

5. The Parallel Poll Flag is cleared.

6. The EOI bit in the Address Status Register is cleared.

7. N_F in the Internal Counter is set to 8 MHz. This setting causes the longest possible T_1 delay to be generated in the Source Handshake (16 μs for 1 MHz clock).

8. The rdy local message is sent.

The initialization state is released by an "immediate execute pon" command (00H written into the Auxiliary Command Register).

The suggested initialization sequence is:

1. Apply a reset pulse or send the reset auxiliary command.

2. Set the desired initial conditions by writing into the Interrupt Enable, Serial Poll Mode, Address Mode, Address 0/1, and EOS Registers. Auxiliary Registers A and B, and the internal counter should also be initialized.

3. Send the "immediate execute pon" auxiliary command to release the initialization state.

4. If a PP2 Parallel Poll implementation is to be used the "lpe" local message may be sent, enabling the 8291A for a Parallel Poll Response on an assigned line. (Refer to the section on Parallel Poll Protocol.)

Using DMA

The 8291A may be connected to the Intel® 8237 or 8257 DMA Controllers or the 8089 I/O Processor for DMA operation. The 8237 will be used to refer to any DMA controller. The DREQ pin of the 8291A requests a DMA byte transfer from the 8237. It is set by BO or BI flip flops, enabled by the DMAO and DMAI bits in the Interrupt Enable 2 Register. (After reading , the INT1 register BO and BI interrupts will be cleared but not BO and BI in DREQ equation.)

The $\overline{DACK}$ pin is driven by the 8237 in response to the DMA request. When $\overline{DACK}$ is true (active low) it sets $\overline{CS}$ = RS0 = RS1 = RS2 = 0 such that the $\overline{RD}$ and $\overline{WR}$ signals sent by the 8237 refer to the Data In and Data Out Registers. Also, the DMA request line is reset by $\overline{DACK}$ ($\overline{RD}$ + $\overline{WR}$).

DMA input sequence:

1. A data byte is accepted from the GPIB by the 8291A.

2. A BI interrupt is generated and DREQ is set.

3. $\overline{DACK}$ and $\overline{RD}$ are driven by the 8237, the contents of the Data In Register are transferred to the system bus, and DREQ is reset.

4. The 8291A sends RFD true on the GPIB and proceeds with the Acceptor Handshake protocol.

DMA output sequence:

1. A BO interrupt is generated (indicating that a byte should be output) and DREQ is asserted.

2. $\overline{DACK}$ and $\overline{WR}$ are driven by the 8237, a byte is transferred from the MCS bus into the Data Out Register, and DREQ is reset.

3. The 8291A sends DAV true on the GPIB and proceeds with the Source Handshake protocol.

It should be noted that each time the device is addressed (MTA + MLA + ton + lon), the Address Status Register should be read, and the 8237 should be initialized accordingly. (Refer to the 8237 or 8257 Data Sheets.)

Polling the 8291A

If polling is used to determine the 8291A's service needs, the CPU must poll the INT bit in the address 0 register. All relevant interrupt status bits must be enabled during initialization for them to affect the INT status bit. The following flow chart illustrates the recommended polling algorithm.

205248–25
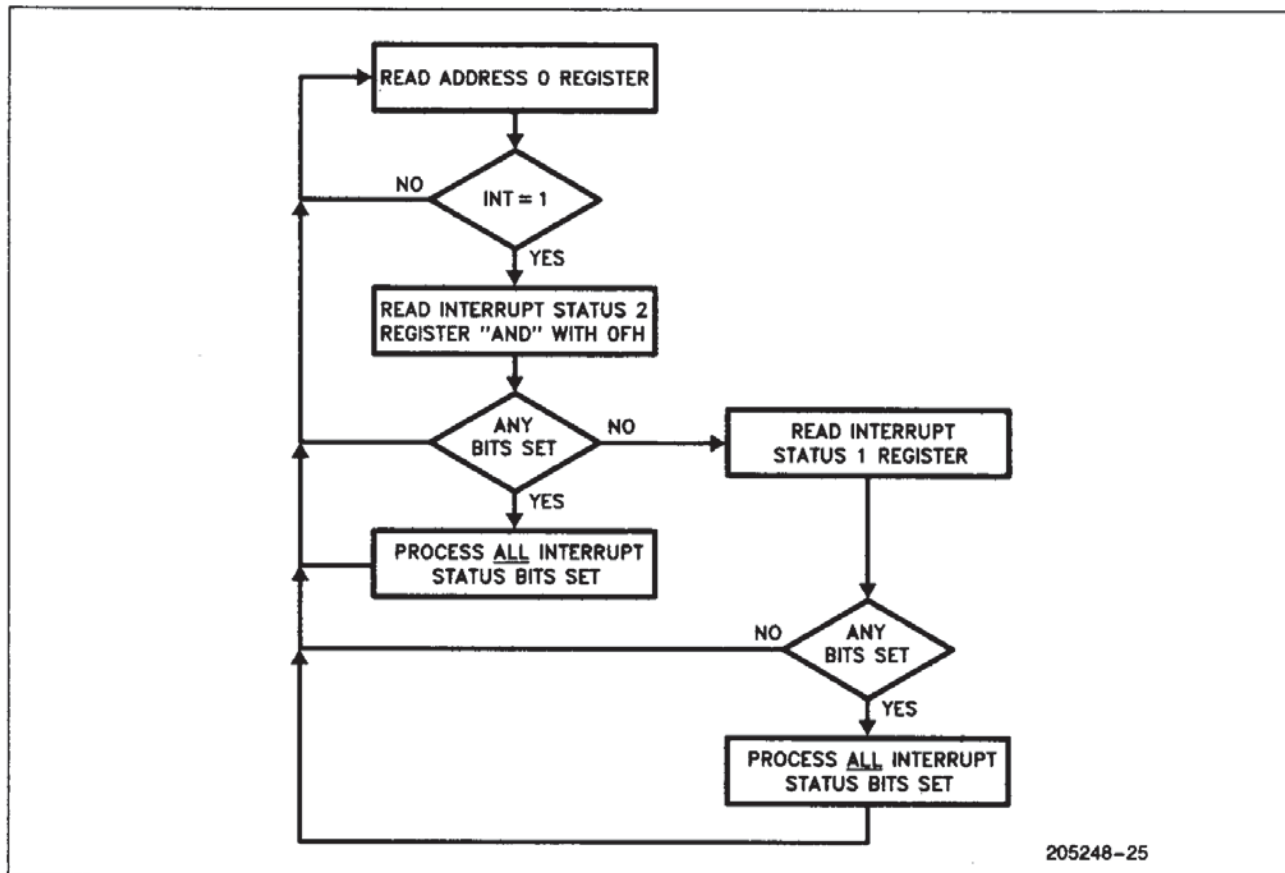
APPLICATION BRIEF

System Configuration

MICROPROCESSOR BUS CONNECTION

The 8291A is 8048/49, 8051, 8080/85, and 8086/88 compatible. The three address pins (RS_0, RS_1, and RS_2) should be connected to the non-multiplexed address bus (for example: A_8, A_9, A_{10}). In case of 8080, any address lines may be used. If the three lowest address bits are used (A_0, A_1, A_2), then they must be demultiplexed first.

EXTERNAL TRANSCEIVERS CONNECTION

The 8293 GPIB Transceiver interfaces the 8291A directly to the IEEE-488 bus. The 8291A and two 8293's can be configured as a talker/listener (see Figure 6) or with the 8292 as a talker/listener/controller (see Figure 7). Absolutely no active or passive external components are required to comply with the complete IEEE-488 electrical specification.

* = GPIB Bus Transceiver

205248-5
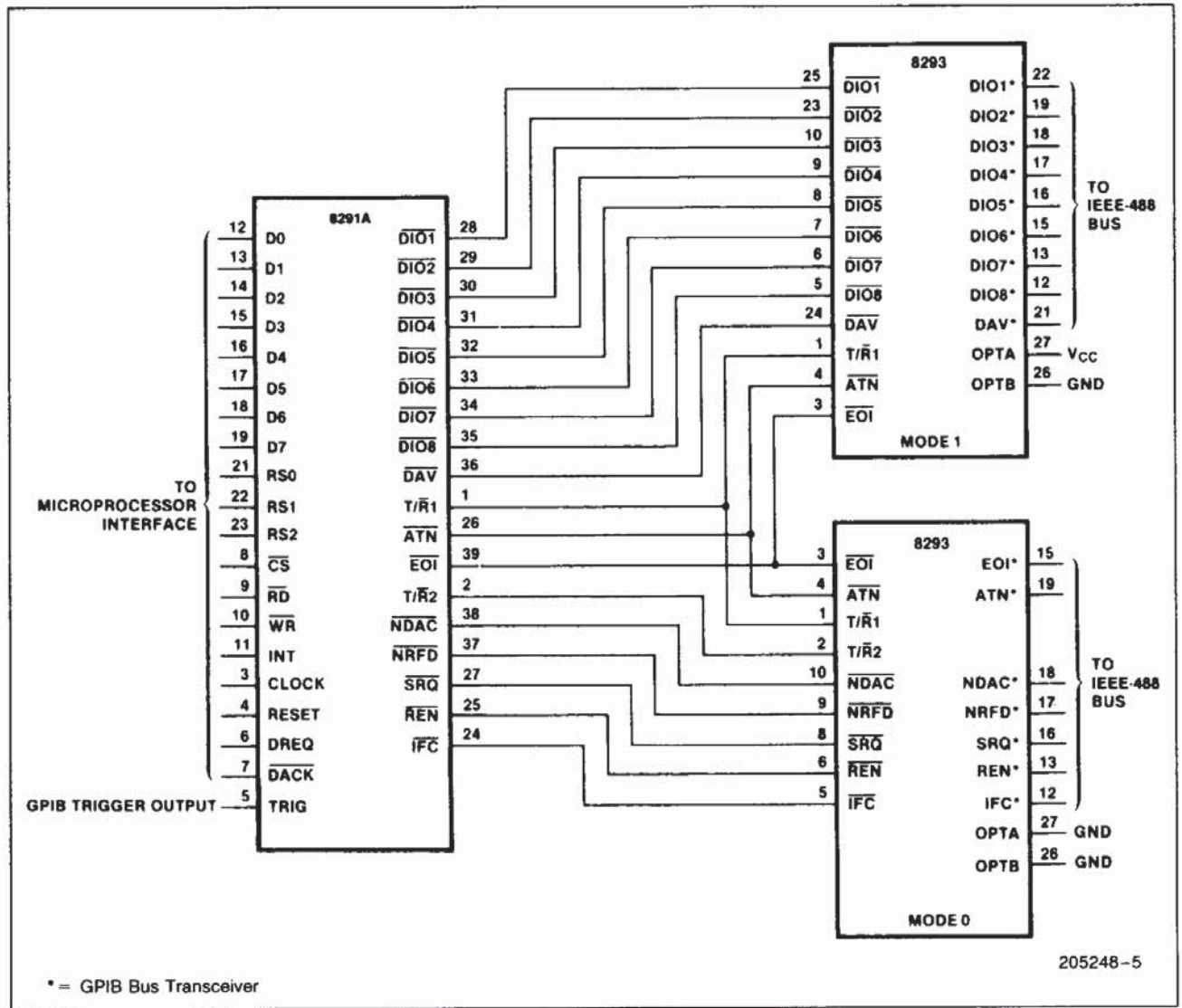
Figure 6. 8291A and 8293 System Configuration
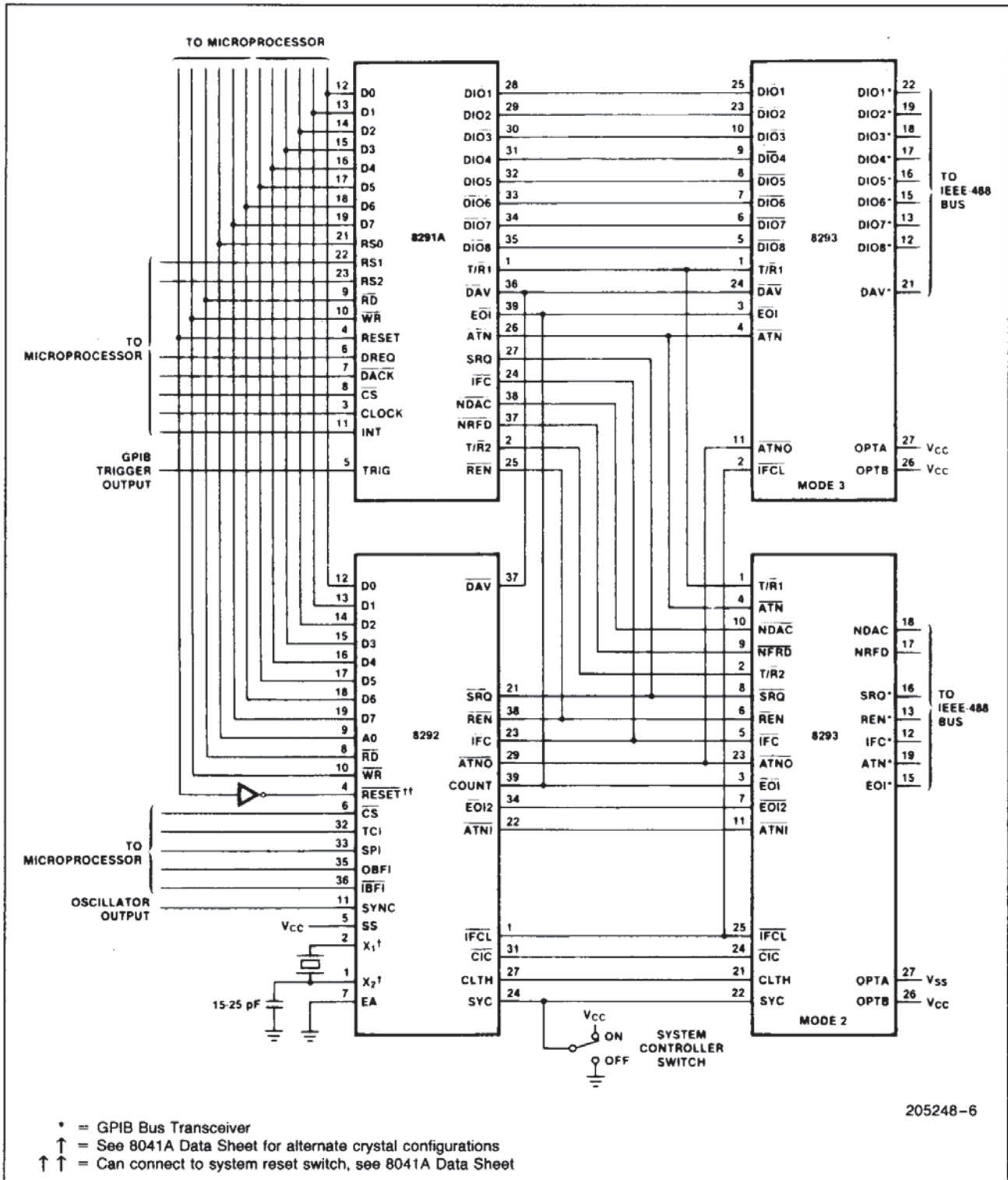
Figure 7. 8291A, 8292, and 8293 System Configuration

* = GPIB Bus Transceiver
↑ = See 8041A Data Sheet for alternate crystal configurations
↑↑ = Can connect to system reset switch, see 8041A Data Sheet

205248–6

Start-Up Procedures

The following section describes the steps needed to initialize a typical 8291A system implementing a talker/listener interface and an 8291A/8292 system implementing a talker/listener/controller interface.

TALKER/LISTENER SYSTEM

Assume a general system configuration with the following features: (i) Polled system interface; (ii) Mode 1 addressing; (iii) same address for talker and listener; (iv) ASCII carriage return as the end-of-sequence (EOS) character; (v) EOI sent true with the last byte; and, (vi) 8 MHz clock.

Initialization. Initialization is accomplished with the following steps:

1. Pulse the RESET input or write 02H to the Auxiliary Mode Register.

2. Write 00H to the Interrupt Enable Registers 1 and 2. This disables interrupt and DMA.

3. Write 01H to the Address Mode Register to select Mode 1 addressing.

4. Write 28H to the Auxiliary Mode Register. This loads 8H to the Auxiliary Register A matching the 8 MHz clock input to the internal T1 delay counter to generate the delay meeting the IEEE spec.

5. Write the talker/listener address to the Address 0/1 register. The three most significant bits are zero.

6. Write an ASCII carriage return (0DH) to the EOS register.

7. Write 84H to the Auxiliary Mode Register to allow EOI to be sent true when the EOS character is sent.

8. Write 00H to the Auxiliary Mode Register. This writes the "Immediate Execute pon" message and takes the 8291A from the initialization state into the idle state. The 8291A will remain idle until the controller initiates some activity by driving ATN true.

Communication. The local CPU now polls the 8291A to determine which controller command has been received.

The controller addresses the 8291A by driving ATN, placing MLA (My Listen Address) on the bus and driving DAV. If the lower five bits of the MLA message match the address programmed into the Address 0/1 register, the 8291A is addressed to listen. It would be addressed to talk if the controller sent the MTA message instead of MLA.

The ADSC bit in the Interrupt Status 2 Register indicates that the 8291A has been addressed or unaddressed. The TA and LA bits in the Address Status Register indicate whether the 8291A is talker (TA = 1), listener (LA = 1), both (TA = LA = 1) or unaddressed (TA = LA = 0).

If the 8291A is addressed to listen, the local CPU can read the Data-In Register whenever the BI (Byte In) interrupt occurs in the Interrupt Status 1 Register. If the END bit in the same register is also set, either EOI or a data byte matching the pattern in the EOS register has been received.

In the talker mode, the CPU writes data into the Byte-Out Register on BO (Byte Out) true.

TALKER/LISTENER/CONTROLLER SYSTEM

Combined with the Intel 8292, the 8291A executes a complete IEEE-488-1978 controller function. The 8291A talks and listens via the data and handshake lines (NRFD, NDAC and DAV). The 8292 controls four of the five bus management lines (IFC, SRQ, ATN and REN). EOI, the fifth line, is shared. The 8291A drives and receives EOI when EOI is used as an end-of-block indicator. The 8292 drives EOI along with ATN during a parallel poll command.

Once again, assume a general system configuration with the following features: (i) Polled system interface; (ii) 8292 as the system controller and controller-in-charge; (iii) ASCII carriage return (0DH) as the EOS identifier; (iv) EOI sent with the last character; and, (v) an external buffer (8282) used to monitor the TCI line.

Initialization. In order to send a command across the GPIB, the 8292 has to drive ATN, and the 8291A has to drive the data lines. Both devices therefore need initialization.

To initialize the 8292:

1. Pulse the RESET input. The 8292 will initially drive all outputs high. TCI, SPI, OBFI, IBFI and CLTH will then go low. The Interrupt Status, Interrupt Mask, Error Flag, Error Mask and Timeout registers will be cleared. The interrupt counter will be disabled and loaded with 255. The 8292 will then monitor the status of the SYC pin. If high, the 8292 will pulse IFC true for at least 100 μs in compliance with the IEEE-488-1978 standard. It will then take control by asserting ATN.

To initialize the 8291A, the following is necessary:

1. Write 00H to Interrupt Enable registers 1 and 2. This disables interrupt and DMA.

2. With the 8292 as the controller-in-charge, it is impossible to address the 8292 via the GPIB. Therefore, the ton or lon modes of the 8291A must be used. To send commands, set the 8291A in the ton mode by writing 80H to the Address Mode Register.

3. Write 26H to the Auxiliary Mode Register to match the T1 data settling time to the 6 MHz clock input.

4. Write an ASCII carriage return (0DH) to the EOS Register.

5. Write 84H to the Auxiliary Mode Register in order to enable "Output EOI on EOS sent" and thus send EOI with the last character.

6. Write 00H—Immediate Execute pon—to the Auxiliary Mode Register to put the 8291A in the idle state.

Communication. Since the 8291A is in the ton mode, a BO interrupt is generated as soon as the immediate Execute pon command is written. The CPU writes the command into the Data Out Register, and repeats it on BO becoming true for as many commands as necessary. $\overline{ATN}$ remains continuously true unless the GTSB (Go To Standby) command is sent to the 8292.

$\overline{ATN}$ has to be false in order to send data rather than commands from the controller. To do this, the following steps are needed:

1. Enable the TCI interrupt if not already enabled.

2. Wait for IBF (Input Buffer Full) in the 8292 Interrupt Status Register to be reset.

3. Write the GTSB (F6H) command to the 8292 Command Field Register.

4. Read the 8282 and wait for TCI to be true.

5. Write the ton (80H) and pon (00H) command to the 8291A Address Mode Register and Auxiliary Mode Registers respectively.

6. Wait for the BO interrupt to be set in the 8291A.

7. Write the data to the 8291A Data-Out Register.

Identically, the user could command the controller to listen rather than talk. To do that, write lon (40H) instead of ton into the Address Mode Register. Then wait for BI rather than BO to go true. Read the data Register.

ABSOLUTE MAXIMUM RATINGS

Ambient Temperature Under Bias ......0°C to 70°C

Storage Temperature .... −65°C to +150°C

Voltage on Any Pin
  With Respect to Ground .... −0.5V to +7V

Power Dissipation......................0.65 Watts

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

D.C. CHARACTERISTICS $V_{CC} = 5V \pm 10\%$, $T_A = 0°C$ to 70°C (Commercial)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2	$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage		0.45	V	$I_{OL} = 2$ mA (4 mA for TR1 pin)
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -400 \mu A$ ($-150 \mu A$ for SRQ pin)
V_{OH-INT}	Interrupt Output High Voltage	2.4 3.5		V V	$I_{OH} = -400 \mu A$ $I_{OH} = -50 \mu A$
I_{IL}	Input Leakage		10	μA	$V_{IN} = 0V$ to V_{CC}
I_{OFL}	Output Leakage Current		±10	μA	$V_{OUT} = 0.45V$, V_{CC}
I_{CC}	V_{CC} Supply Current		120	mA	$T_A = 0°C$

A.C. CHARACTERISTICS $V_{CC} = 5V \pm 10\%$, $T_A = 0°C$ to 70°C (Commercial)

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{AR}	Address Stable Before $\overline{READ}$	0		ns	
t_{RA}	Address Hold After $\overline{READ}$	0		ns	
t_{RR}	$\overline{READ}$ Width	140		ns	
t_{AD}	Address Stable to Data Valid		250	ns	
t_{RD}	$\overline{READ}$ to Data Valid		100	ns	
t_{RDF}	Data Float After $\overline{READ}$	0	60	ns	
t_{AW}	Address Stable Before $\overline{WRITE}$	0		ns	
t_{WA}	Address Hold After $\overline{WRITE}$	0			
t_{WW}	$\overline{WRITE}$ Width	170		ns	
t_{DW}	Data Set Up Time to the Trailing Edge of WRITE	130		ns	
t_{WD}	Data Hold Time After $\overline{WRITE}$	0		ns	
t_{DKDR4}	$\overline{RD} \downarrow$ or $\overline{WR} \downarrow$ to DREQ $\downarrow$		130	ns	
t_{DKDA6}	$\overline{RD} \downarrow$ to Valid Data (D_0–D_7)		200	ns	$\overline{DACK} \downarrow$ to $\overline{RD} \downarrow$ $0 \le t \le 50$ ns

3

WAVEFORMS

READ

205248–7

WRITE

205248–8

DMA

205248–9
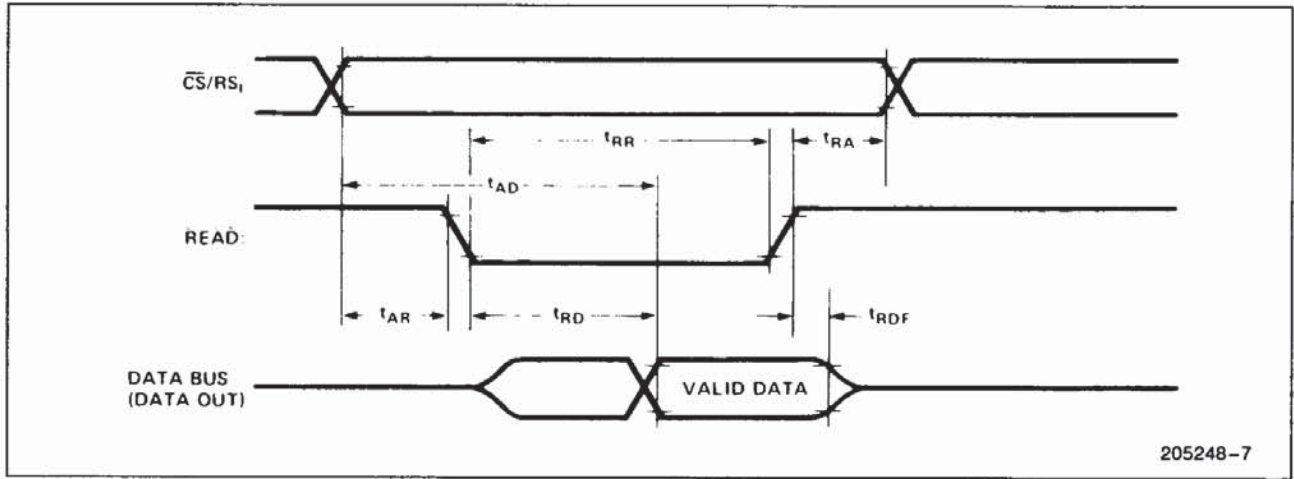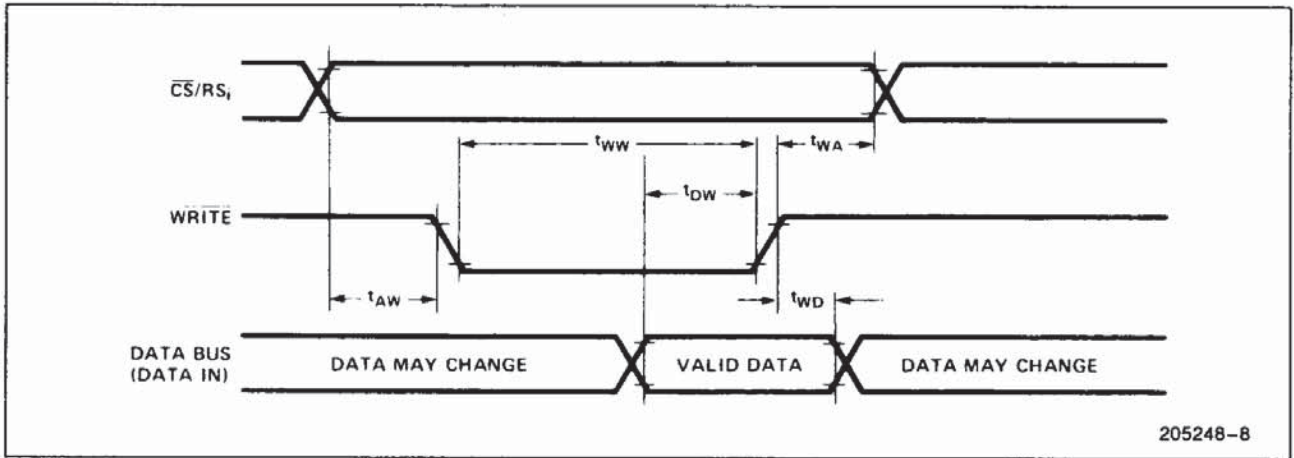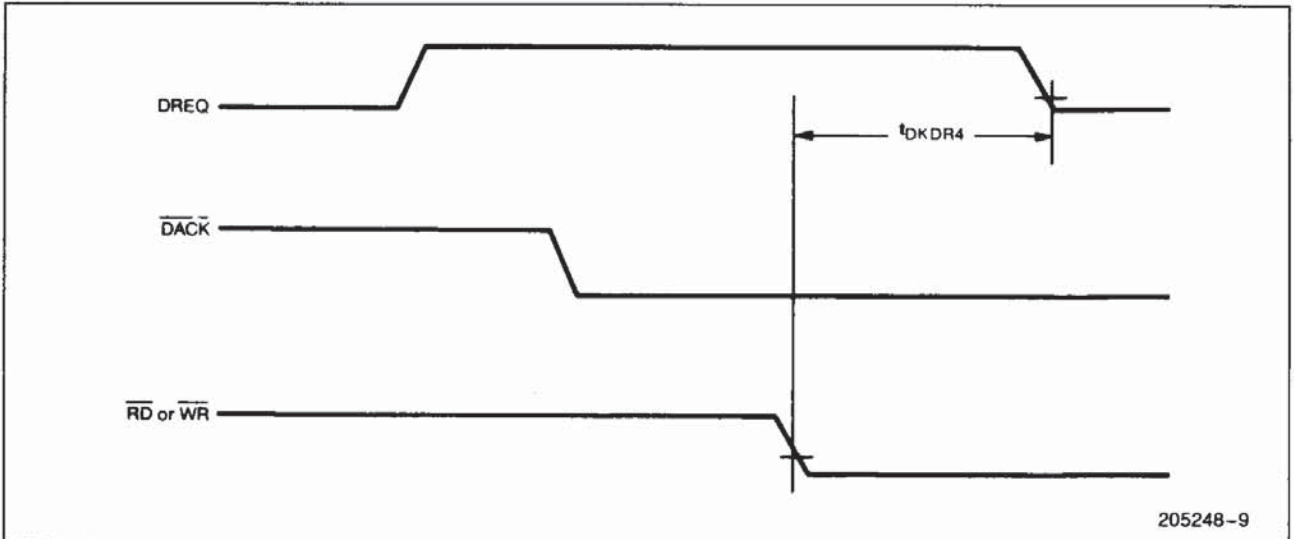
A.C. TIMING MEASUREMENT POINTS AND LOAD CONDITIONS

INPUT/OUTPUT

2.4

2.0 — TEST POINTS — 2.0

0.8 — 0.8

0.45

205248–10

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

DEVICE UNDER TEST

$C_L = 150$ pF

205248–11

3
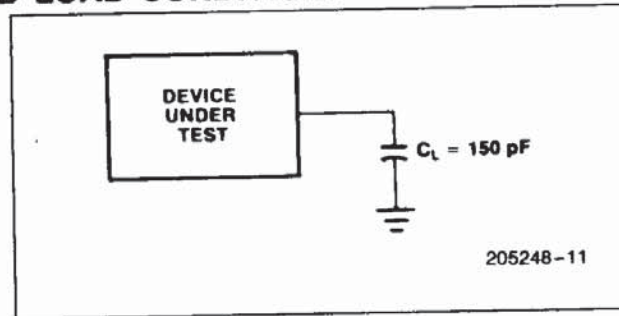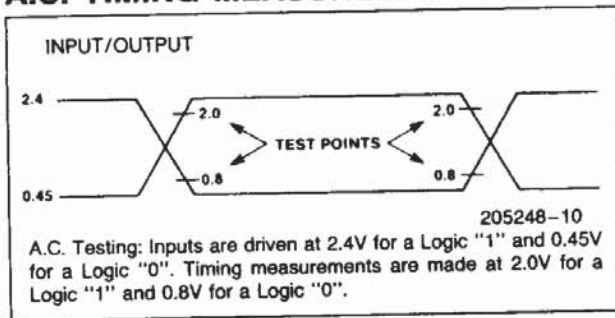
GPIB TIMINGS(1)

Symbol	Parameter	Max	Units	Test Conditions
TEOT13[2]	$\overline{EOI} \downarrow$ to TR1 $\uparrow$	135	ns	PPSS, ATN = 0.45V
TEOD16	$\overline{EOI} \downarrow$ to $\overline{DIO}$ Valid	155	ns	PPSS, ATN = 0.45V
TEOT12	$\overline{EOI} \uparrow$ to TR1 $\downarrow$	155	ns	PPSS, ATN = 0.45V
TATND4	$\overline{ATN} \downarrow$ to $\overline{NDAC} \downarrow$	155	ns	TACS, AIDS
TATT14	$\overline{ATN} \downarrow$ to TR1 $\downarrow$	155	ns	TACS, AIDS
TATT24	$\overline{ATN} \downarrow$ to TR2 $\downarrow$	155	ns	TACS, AIDS
TDVND3-C	$\overline{DAV} \downarrow$ to $\overline{NDAC} \uparrow$	650	ns	AH, CACS
TNDDV1	$\overline{NDAC} \uparrow$ to $\overline{DAV} \uparrow$	350	ns	SH, STRS
TNRDR1	$\overline{NRFD} \uparrow$ to $\overline{DREQ} \uparrow$	400	ns	SH
TDVDR3	$\overline{DAV} \downarrow$ to $\overline{DREQ} \uparrow$	600	ns	AH, LACS, ATN = 2.4V
TDVND2-C	$\overline{DAV} \uparrow$ to $\overline{NDAC} \downarrow$	350	ns	AH, LACS
TDVNR1-C	$\overline{DAV} \uparrow$ to $\overline{NRFD} \uparrow$	350	ns	AH, LACS, rdy = True
TRDNR3	$\overline{RD} \downarrow$ to $\overline{NRFD} \uparrow$	500	ns	AH, LACS
TWRD15	$\overline{WR} \uparrow$ to $\overline{DIO}$ Valid	280	ns	SH, TACS, RS = 0.4V
TWREO5	$\overline{WR} \uparrow$ to $\overline{EOI}$ Valid	350	ns	SH, TACS
TWRDV2	$\overline{WR} \uparrow$ to $\overline{DAV} \downarrow$	830 + t_{SYNC}	ns	High Speed Transfers Enabled, $N_F = f_C$, $t_{SYNC} = \frac{1}{2} \bullet f_C$

NOTES:
1. All GPIB timings are at the pins of the 8291A.
2. The last number in the symbol for any GPIB timing parameter is chosen according to the transition directions of the reference signals. The following table describes the numbering scheme.

$\uparrow$ to $\uparrow$	1
$\uparrow$ to $\downarrow$	2
$\downarrow$ to $\uparrow$	3
$\downarrow$ to $\downarrow$	4
$\uparrow$ to VALID	5
$\downarrow$ to VALID	6

APPENDIX A

MODIFIED STATE DIAGRAMS

Figure A-1 presents the interface function state diagrams. It is derived from IEEE Std. state diagrams, with the following changes:

A. The 8291A supports the complete set of IEEE-488 interface functions except for the controller. These include: SH1, AH1, T5, TE5, L3, LE3, SR1, RL1, PP1, DC1, DT1, and C0.

B. Addressing modes included in T, L state diagrams.

Note that in Mode 3, MSA, OSA are generated only after secondary address validity check by the microprocessor (APT interrupt).

C. In these modified state diagrams, the IEEE-488-1978 convention of negative (low true) logic is followed. This should not be confused with the Intel pin- and signal-naming convention based on positive logic. Thus, while the state diagrams below carry low true logic, the signals described elsewhere in this data sheet are consistent with Intel notation and are based on positive logic.

Consider the condition when the Not-Ready-For-Data signal (pin 37) is active. Intel indicates this active low signal with the symbol $\overline{NRFD}$ ($V_{OUT} \leq V_{OL}$ for AH; $V_{IN} \leq V_{IL}$ for SH). The IEEE-488-1978 Standard, in its state diagrams, indicates the active state of this signal (True condition) with NRFD.

D. All remote multiline messages decoded are conditioned by ACDS. The multiplication by ACDS is not drawn to simplify the diagrams.

E. The symbol

205248-12
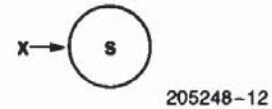
indicates:

1. When event X occurs, the function returns to state S.

2. X overrides any other transition condition in the function.

Statement 2 simplifies the diagram, avoiding the explicit use of X to condition all transitions from S to other states.
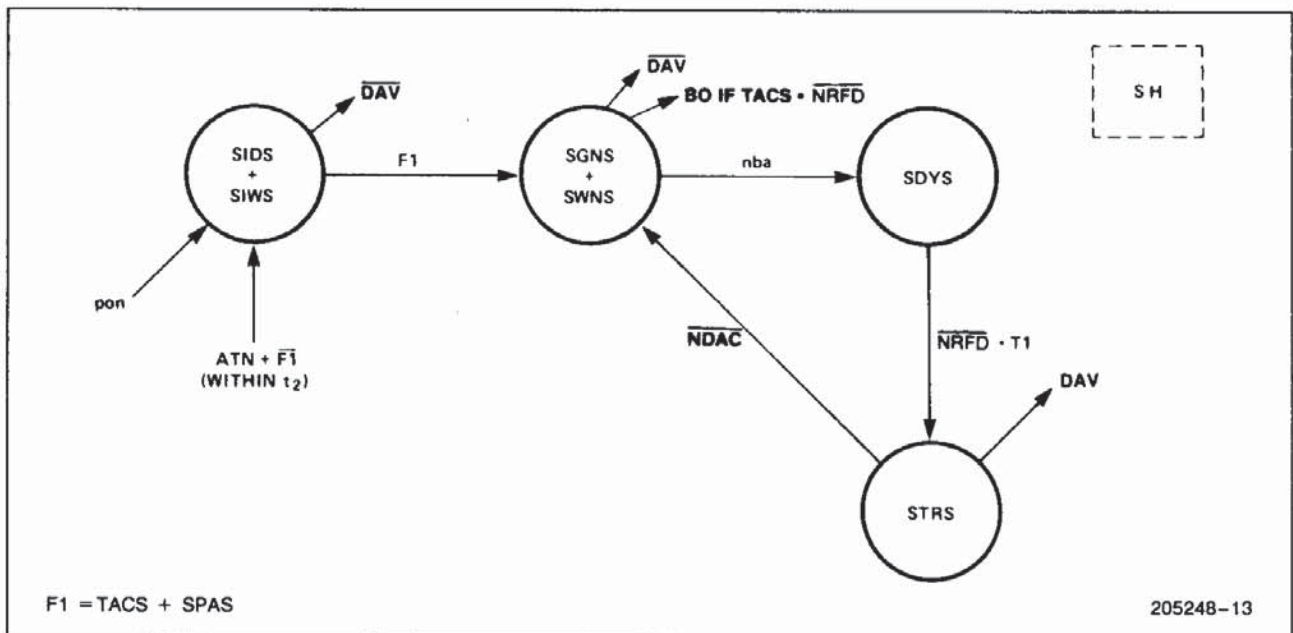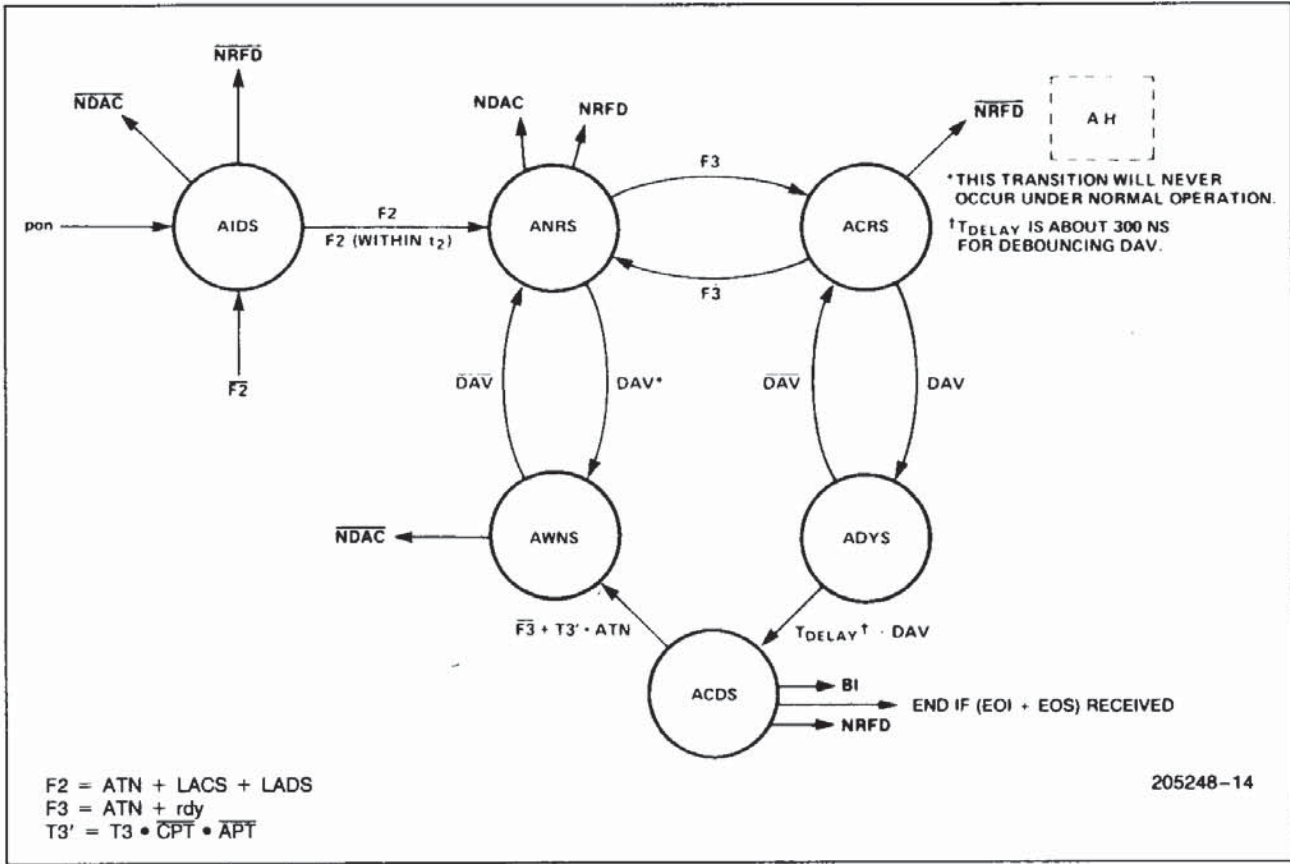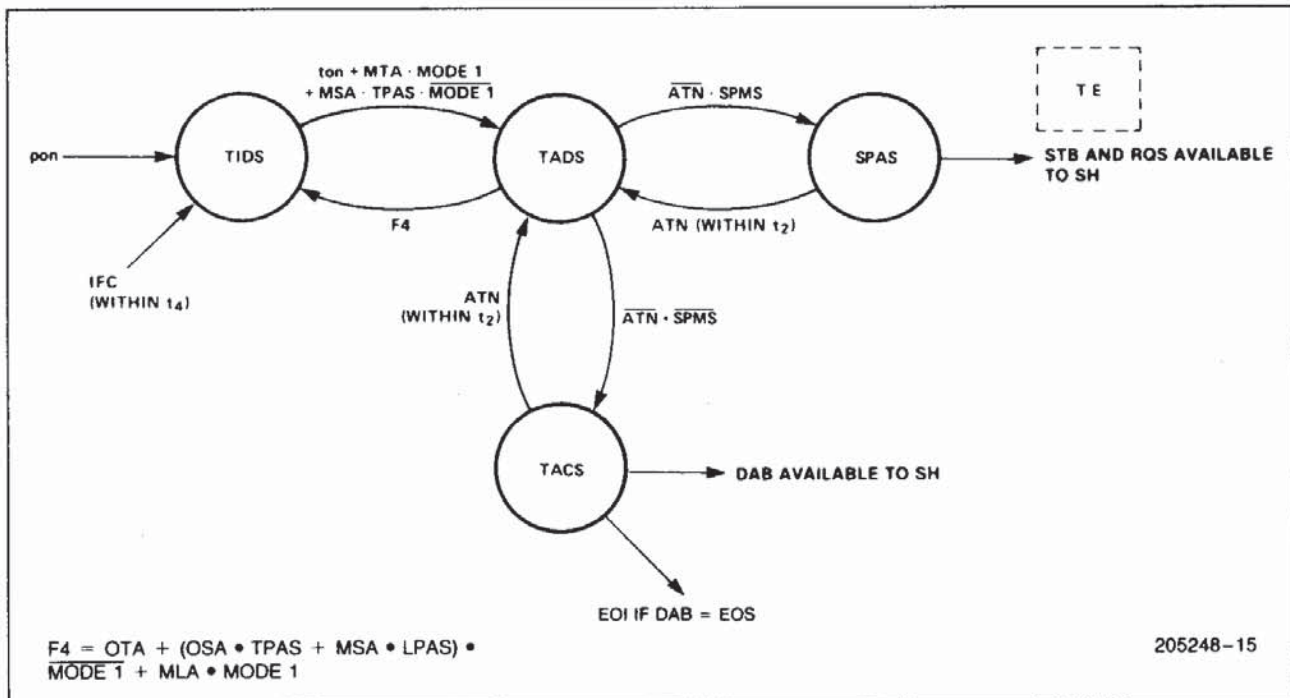
Level	Logic	Convention	
		IEEE-488	Intel
0	T	DAV	$\overline{DAV}$
1	F	$\overline{DAV}$	DAV
0	T	NDAC	$\overline{NDAC}$
1	F	$\overline{NDAC}$	NDAC
0	T	NRFD	$\overline{NRFD}$
1	F	$\overline{NRFD}$	NRFD

$F1 = TACS + SPAS$

205248-13

Figure A-1. 8291A State Diagrams

$F2 = ATN + LACS + LADS$
$F3 = ATN + rdy$
$T3' = T3 \cdot \overline{CPT} \cdot \overline{APT}$

205248–14

$F4 = \overline{OTA} + (OSA \cdot TPAS + MSA \cdot LPAS) \cdot$
$\overline{MODE\ 1} + MLA \cdot MODE\ 1$

205248–15

Figure A-1. 8291A State Diagrams (Continued)

205248-16

205248-17

Figure A-1. 8291A State Diagrams (Continued)

205248-18

F5 = (MLA • MODE 1 + LPAS • MSA • $\overline{MODE\ 1}$)

205248-19

Figure A-1. 8291A State Diagrams (Continued)

PP 2

*IDY = ATN • EOI

205248–20

D C
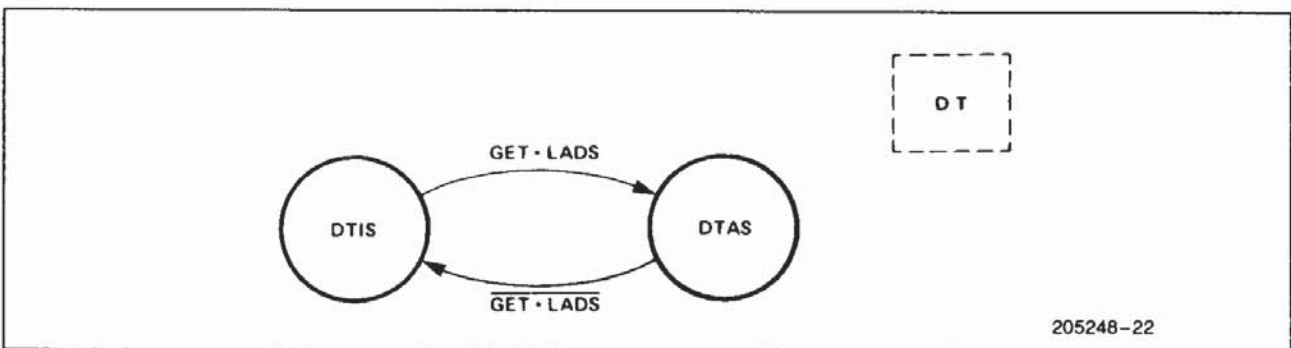
F6 = DCL + SDC • LADS
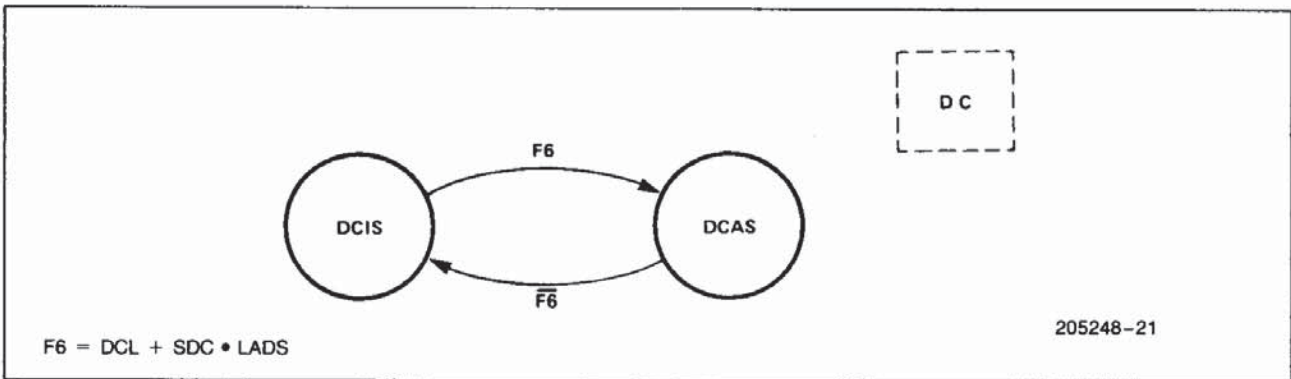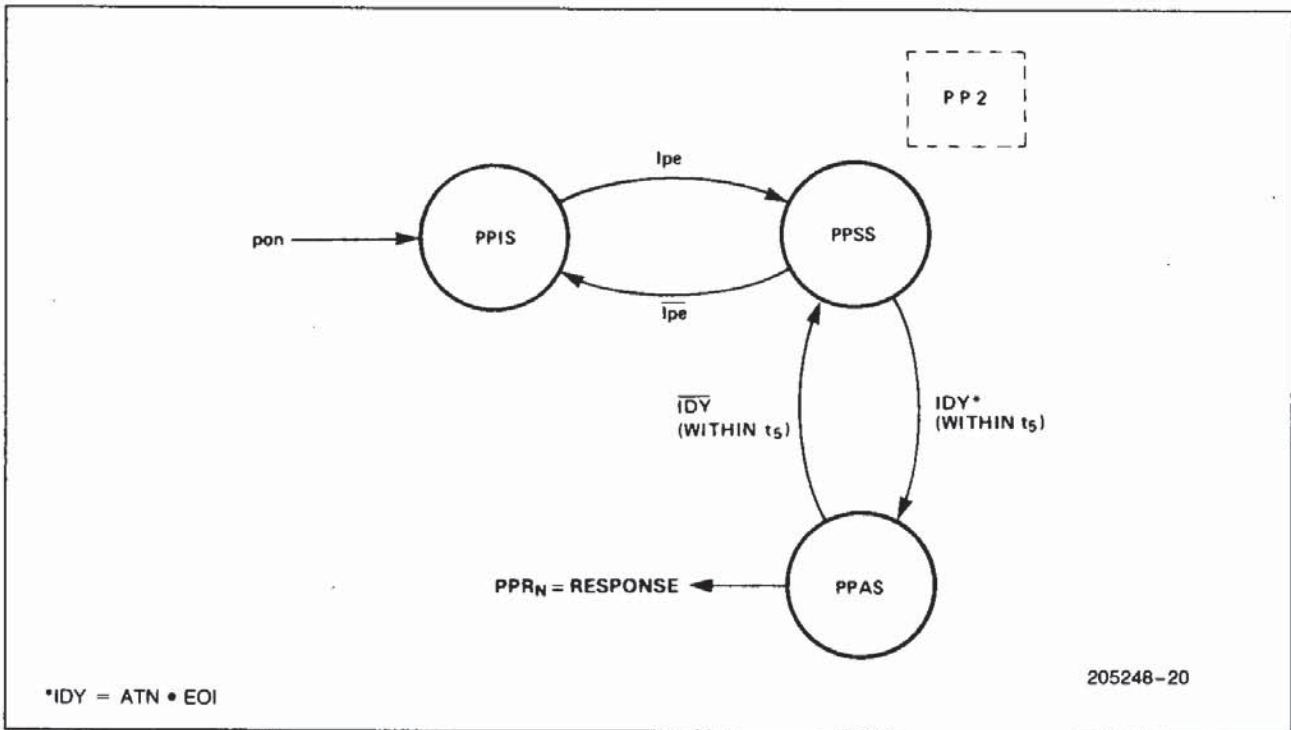
205248–21

D T

205248–22
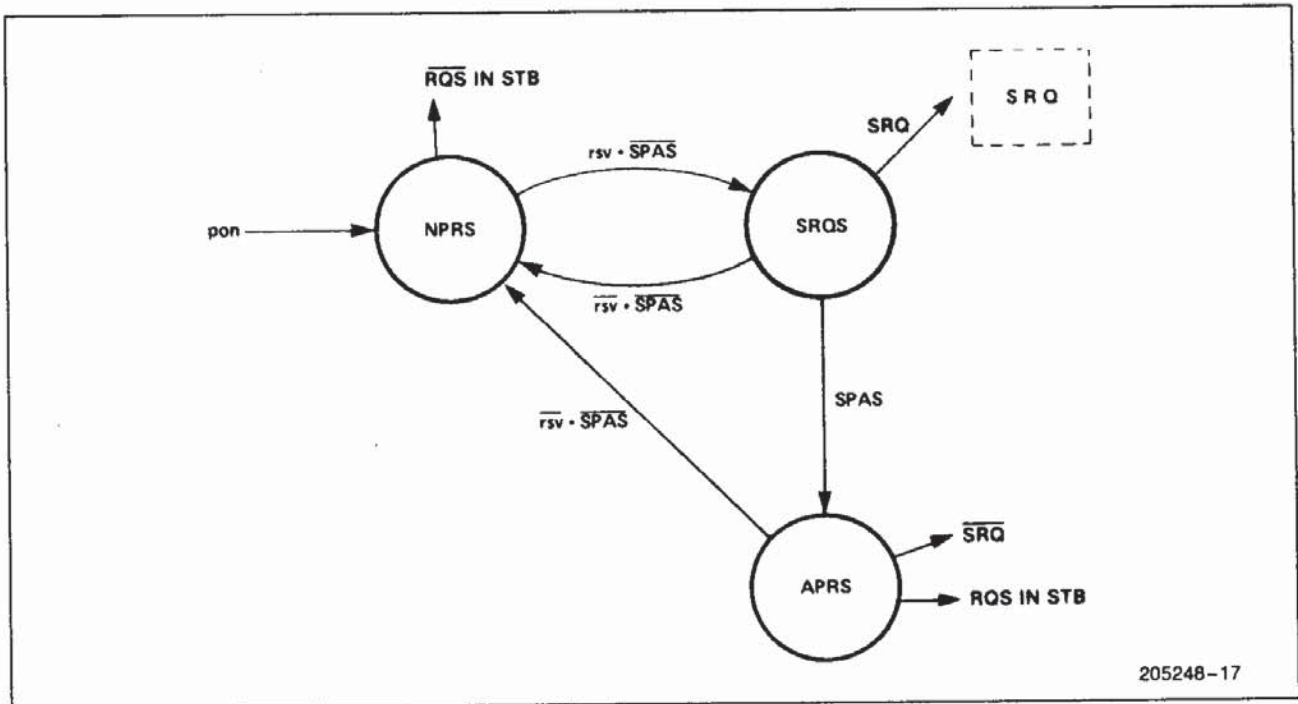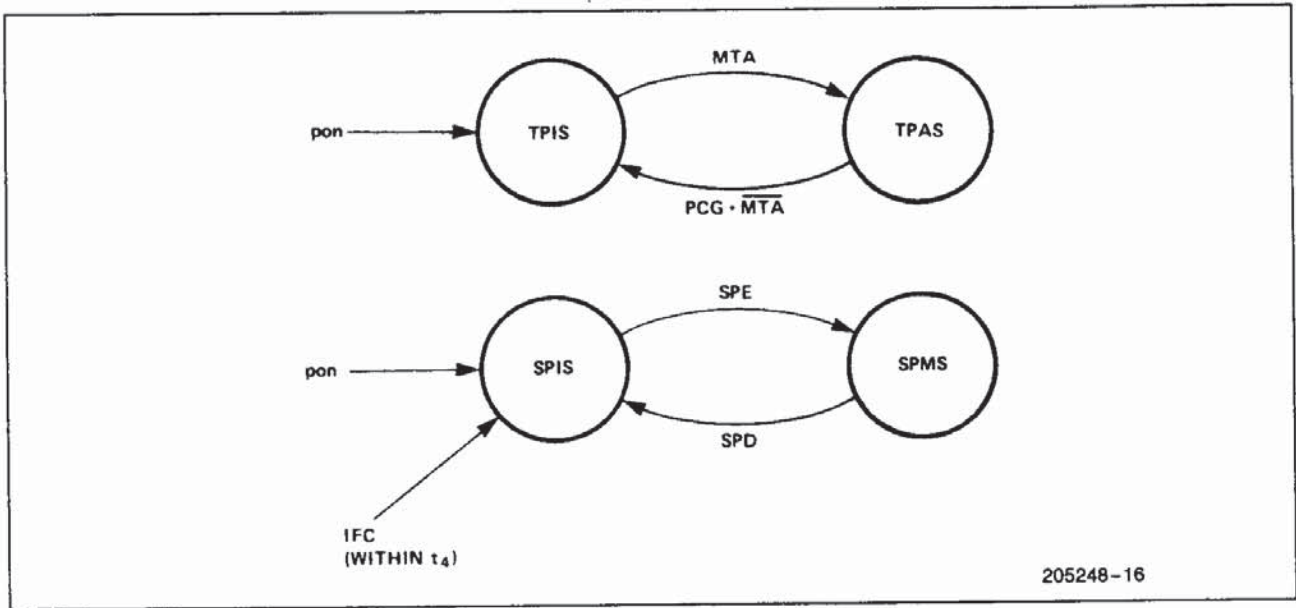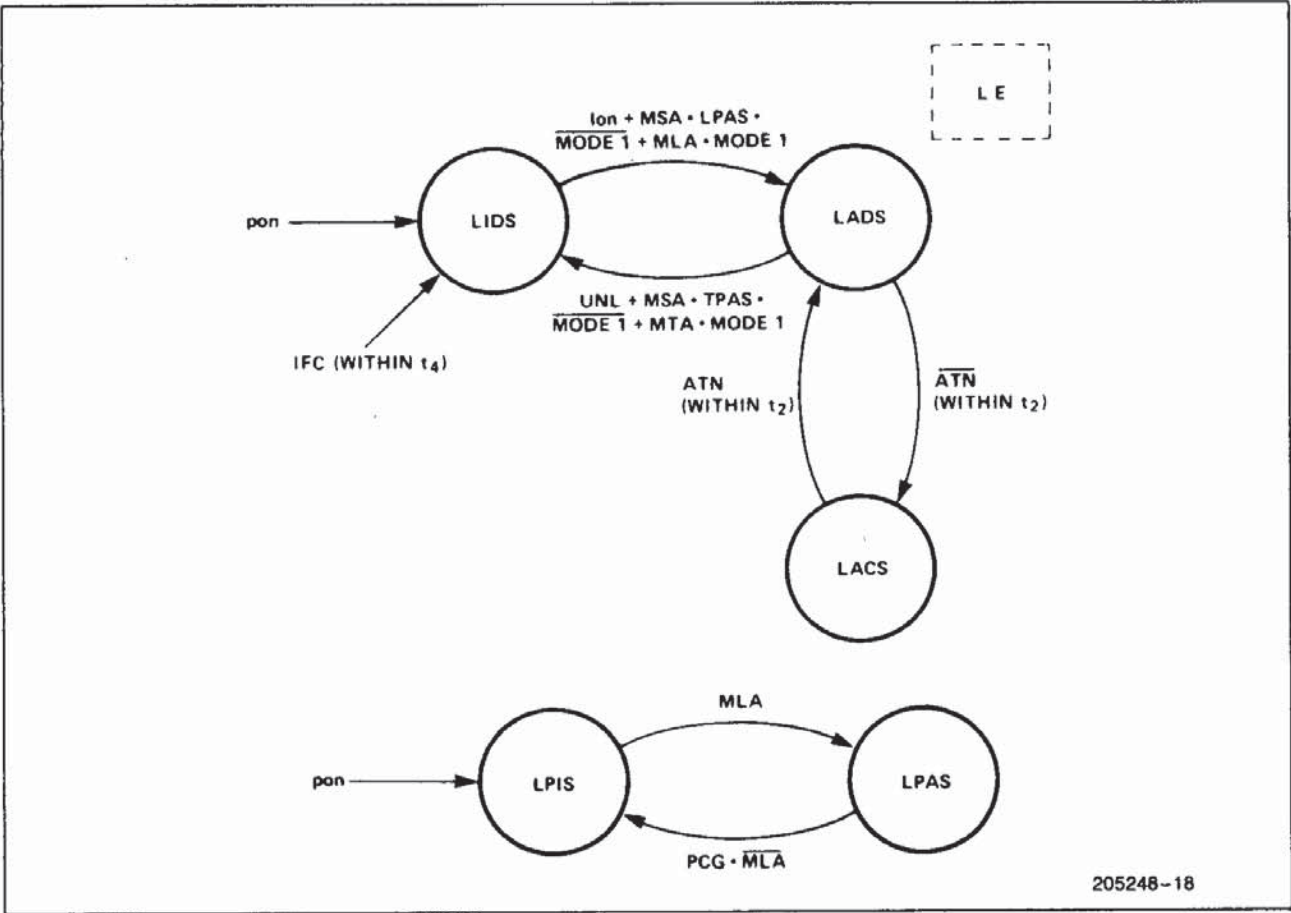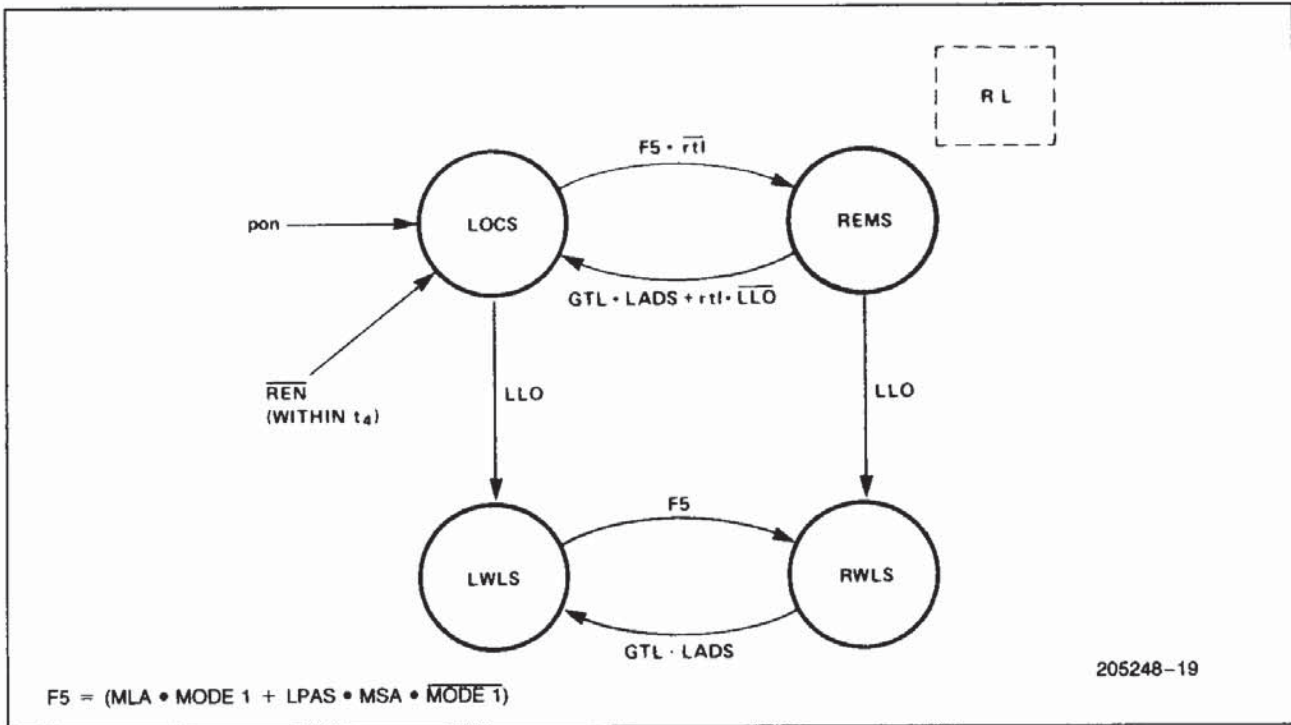
Figure A-1. 8291A State Diagrams (Continued)

APPENDIX B

Table B-1. IEEE 488 Time Values

Time Value Identifier[1]	Function (Applies to)	Description	Value
T_1	SH	Settling Time for Multiline Messages	$\geq 2\ \mu s$[2]
t_2	LC, $\overline{\text{IC}}$, SH, AH, T, L	Response to ATN	≤ 200 ns
T_3	AH	Interface Message Accept Time[3]	> 0[4]
t_4	T, TE, L, LE, C, CE	Response to IFC or REN False	$< 100\ \mu s$
t_5	PP	Response to ATN + EOI	≤ 200 ns
T_6	C	Parallel Poll Execution Time	$\geq 2\ \mu s$
T_7	C	Controller Delay to Allow Current Talker to see ATN Message	≥ 500 ns
T_8	C	Length of IFC or REN False	$> 100\ \mu s$
T_9	C	Delay for EOI[5]	$\geq 1.5\ \mu s$[6]

NOTES:

1. 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 must remain in a state before exiting.

2. If three-state drivers are used on the $\overline{\text{DIO}}$, $\overline{\text{DAV}}$, and $\overline{\text{EOI}}$ lines, T_1 may be:

    1. $\geq$ 1100 ns.

    2. Or $\geq$ 700 ns if it is known that within the controller ATN is driven by a three-state driver.

    3. Or $\geq$ 500 ns for all subsequent bytes following the first sent after each false transition of ATN (the first byte must be sent in accordance with (1) or (2).

    4. Or $\geq$ 350 ns for all subsequent bytes following the first sent after each false transition of ATN under conditions specified in Section 5.2.3 and warning note. See IEEE Standard 488.

3. Time required for interface functions to accept, not necessarily respond to interface messages.

4. Implementation dependent.

5. Delay required for $\overline{\text{EOI}}$, $\overline{\text{NDAC}}$, and $\overline{\text{NRFD}}$ signal lines to indicate valid states.

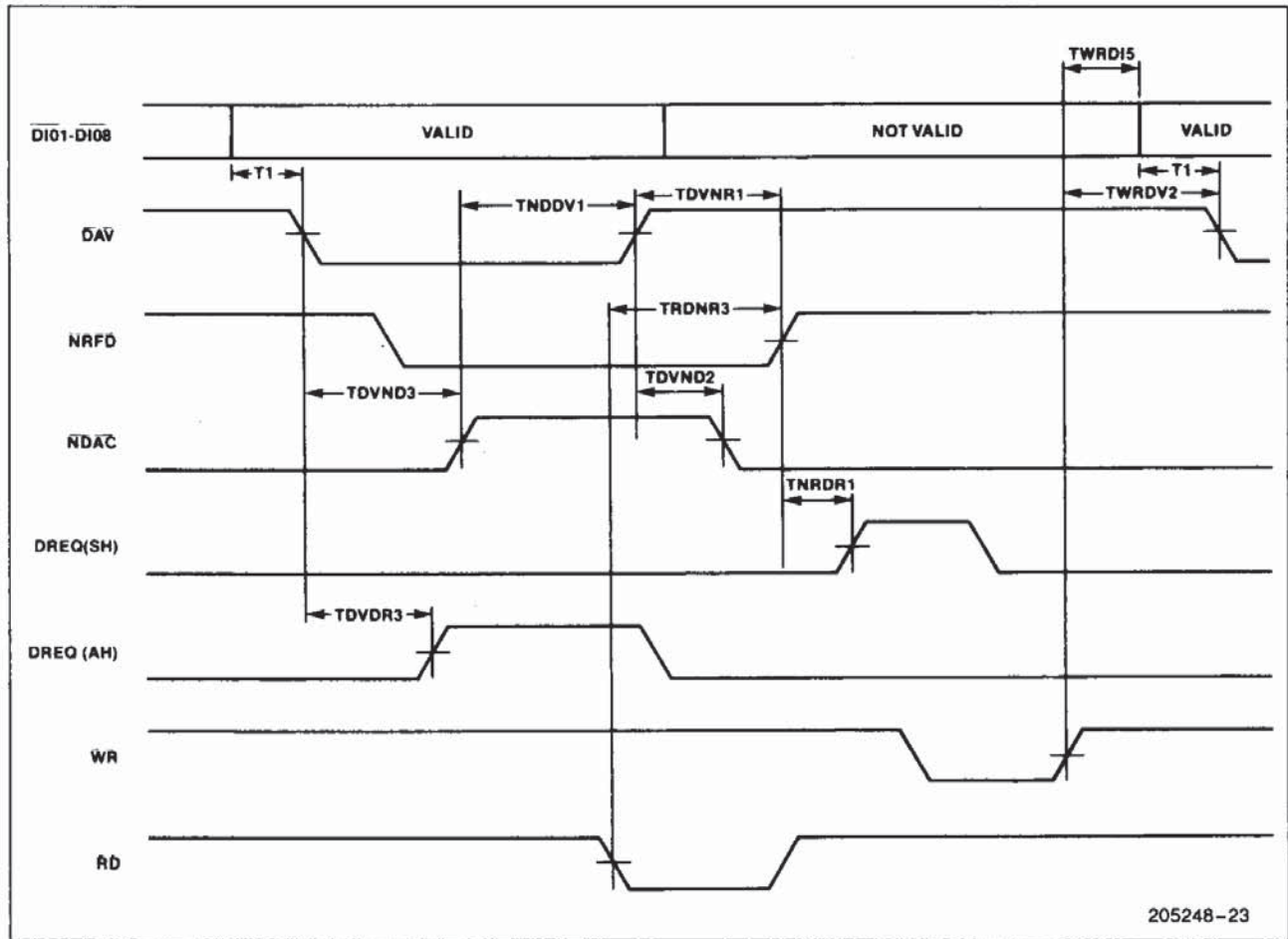6. $\geq$ 600 ns for three-state drivers.

3

APPENDIX C
THE THREE-WIRE HANDSHAKE

Figure C-1. 3-Wire Handshake Timing at 8291A