WD9914 General Purpose Interface Bus (GPIB) Controller

WESTERN DIGITAL

WD9914
General Purpose Interface Bus (GPIB) Controller

TABLE OF CONTENTS

PAGE

Features 1
Description 1
Pin Designation 1
Pin Description 2
Architecture 3
Registers
- Interrupt Mask and Status Register 0 5
- Interrupt Mask and Status Register 1 6
- Address Status Register 7
- Address Register 7
- Auxiliary Command Register 7
  - Description of Auxiliary Commands
    - Software Reset (swrst) 8
    - Release DAC Holdoff (dacr) 9
    - Release RFD Holdoff (rhdf) 9
    - Holdoff On All Data (hdfa) 9
    - Holdoff On End (hdfe) 9
    - Set New Byte Available False (nbaf) 9
    - Force Group Execute Trigger (fget) 9
    - Return To Local (rtl) 9
    - Force End Or Identify (feoi) 9
    - Listen Only (lon) 9
    - Talk Only (ton) 9
    - Go to Standby (gts) 9
    - Take Control Synchronously (tcs) 9
    - Request Parallel Poll (rpp) 10
    - Take Control Asynchronously (tca) 10
    - Send Interface Clear (sic) 10
    - Send Remote Enable (sre) 10
    - Request Control (rqc) 10
    - Release Control (ric) 10
    - Disable All Interrupts (dai) 10
    - Pass Through Next Secondary (pts) 10
    - Set T1 Delay (std1) 10
    - Shadow Handshake (shdw) 10
    - Very Short T1 Delay (vstd1) 10
    - Request Service Bit 2 (rsv2) 10
- Bus Status Register 11
- Serial Poll Register 11
- Command Pass Through Register 11
- Parallel Poll Register 11
- Data In Register 12
- Data Out Register 12

Direct Memory Access 12
Terminal Assignments and Functions 12
Transceiver Connections 13
State Diagram Implementation 15
Auxiliary Commands 15
Acceptor Handshake 16
Source Handshake 18

Talker And Listener Functions 19
Service Request Function 21
Remote/local Function 23
Parallel Poll Function 24
Remotely Configured Parallel Poll 24
Controller Function 25
Controller Self Addressing 25
Passing Control 25
System Controller 28
Electrical Specifications 34
Timing Characteristics and Requirements 35
Package Diagrams 44
Ordering Information 45

TABLES

1. WD9914 Read Register 4
2. WD9914 Write Registers 5
3. Auxiliary Commands 8
4. Software Reset Conditions 9
5. Auxiliary Command State Diagram Mnemonics 16
6. Acceptor Handshake Mnemnonics 17
7. Acceptor Handshake Message Outputs 17
8. Source Handshake Mnemonics 18
9. Source Handshake Message Outputs 19
10. Talker and Listener Mnemonics 21
11. Talker Function Message Outputs 21
12. Service Request Mnemonics 22
13. Service Request Message Outputs 23
14. Remote/Local Mnemonics 23
15. Parallel Poll Mnemonics 24
16. Parallel Poll Message Outputs 25
17. Controller Function Mnemonics 26
18. Controller Function Message Outputs 27
19. Multiline Interface Messages 30

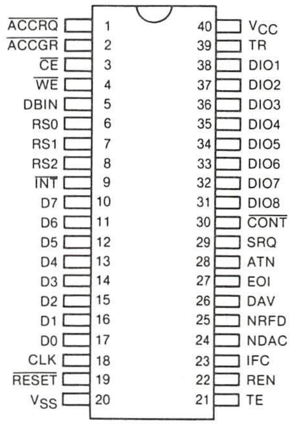
FIGURES

1. Simplified Block Diagram 4
2. DMA Configuration 13
3. Transceiver Connections 14
4. WD9914 Auxiliary Command State Diagram 15
5. WD9914 Acceptor Handshake State Diagram 16
6. WD9914 Source Handshake State Diagram 18
7. WD9914 Listener State Diagram 19
8. WD9914 Talker State Diagram 20
9. Service Request Diagram 22
10. WD9914 Remote Local State Diagram 23
11. WD9914 Parallel Poll State Diagram 24
12. WD9914 Controller State Diagrams 26
13. Passing Control Between WD9914 28
14. IFC and REN Pins 29
15. Controller Taking Control 31
16. Controller as a Listener (Going to Standby) 32
17. Controller as a Talker (Going to Standby) 33
18. Controller Parallel Polling 34
19. WD9914 Clock Cycle Timing 37
20. WD9914 Read Cycle Timing 38
21. WD9914 Write Cycle Timing 38
22. WD9914 DMA Read Operation 39
23. WD9914 DMA Write Operation 39
24. WD9914 Source and Acceptor Handshake Timing 40
25. WD9914 Acceptor Handshake Timing "ATN" True 41
26. WD9914 Response to "ATN" and "EOI" 42
27. WD9914 Controller Timing 43

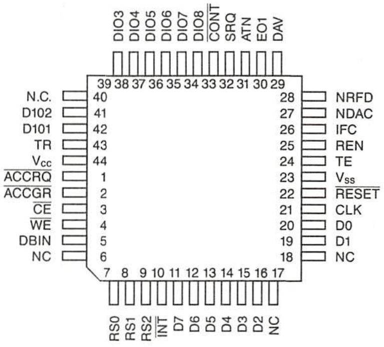
WD9914
General Purpose Interface Bus (GPIB) Controller

FEATURES

- Handles all IEEE-488 1975/78 functions
- Compatible with IEEE-488A 1980 supplement
- Talker and listener function (T, TE, L, LE)
- Automatic source and acceptor handshakes (SH, AH)
- Controller with pass control
- System controller capabilities
- Device trigger and device clear capabilities (DT, DC)
- Optional automatically cleared 'request service bit'
- Parallel and serial poll facilities (PP)
- Remote/local function with local lockout (RL)
- Single or dual primary addressing
- Secondary address capabilities
- Direct interface to SN75160/161/162 bus transceivers with no additional logic
- Compatible with most microprocessors
- Direct memory access facilities
- Memory-mapped microprocessor interface
- Available in 40-pin DIP or 44-pin QUAD package
- Single +5V supply
- Compatible with TMS9914A features



DIP PIN DESIGNATION



QUAD PIN DESIGNATION

DESCRIPTION

The WD9914 provides an interface between a microprocessor system and the General Purpose Interface Bus (GPIB) specified in the IEEE-488 1975/78 standards and the IEEE-488A 1980 supplement. The device is controlled and configured through 8-bit memory mapped registers and enables all aspects of the standards to be implemented, including talker, listener and controller.

PIN DESCRIPTION

PIN NUMBER			I/O	
DIP	QUAD	SIGNAL NAME	(TYPE)	DESCRIPTION
1	1	$\overline{\text{ACCRQ}}$	O(p/p)	ACCESS REQUEST: This pin becomes active (low) to request a direct memory access.
2	2	$\overline{\text{ACCGR}}$	I	ACCESS GRANTED: When received from the direct memory access control logic, this enables the byte onto the data bus. ACCGR must be high when not participating in DMA transfer.
3	3	$\overline{\text{CE}}$	I	CHIP ENABLE: $\overline{\text{CE}}$ low allows access of read and write registers. If $\overline{\text{CE}}$ is high, D0-D7 are in high impedance unless $\overline{\text{ACCGR}}$ is low.
4	4	$\overline{\text{WE}}$	I	WRITE ENABLE: When active (low), indicates to the WD9914 that data are being written to one of its registers.
5	5	DBIN	I	DATA BUS IN: An active (high) state indicates to the WD9914 that a read is about to be carried out by the MPU.
6	7	RS0	I	REGISTER SELECT LINES: Determine which register is addressed by the MPU during a read or write operation.
7	8	RS1	I	
8	9	RS2	I	
9	10	$\overline{\text{INT}}$	O(o/d) (no pullup)	INTERRUPT: Sent to the MPU to cause a branch to a service routine.
10-17	11-16 19-20	D7-D0	I/O(p/p)	Data transfer lines on the MPU side of the device.
18	21	CLK	I	CLOCK Input: 500 kHz to 5 MHz. Need not be synchronous to system clock.
19	22	$\overline{\text{RESET}}$*	I	INITIALIZES the WD9914 at power-on.
20	23	V_{SS}		Ground reference voltage.
21	24	TE	O(p/p)	TALK ENABLE: Controls the direction of the transfer of the line transceivers. Logically, it is: (CACS + TACS + EIO.ATN.(CIDS + CADS). $\overline{\text{SWRST}}$).
22	25	REN	I/O(o/d)	REMOTE ENABLE: Sent by system controller to select control either from the front panel or from the IEEE bus.
23	26	IFC	I/O(o/d)	INTERFACE CLEAR: Sent by the system controller to set the interface system into a known quiescent state. The system controller becomes the controller in charge.
24	27	NDAC	I/O(p/p)	NOT DATA ACCEPTED: handshake line. Acceptor sets this false (high) when it has latched the data from the I/O lines.
25	28	NRFD	I/O(p/p)	NOT READY FOR DATA: handshake line. Sent by acceptor to indicate readiness for the next byte.
26	29	DAV	I/O(p/p)	DATA VALID: Handshake line controlled by source to show acceptors when valid data is present to the bus.
27	30	EOI	I/O(p/p)	END OR IDENTIFY: If ATN is false (high), this indicates the end of a message block. If ATN is true (low), the controller is requesting a parallel poll.

PIN DESCRIPTION (Continued)

PIN NUMBER			I/O	
DIP	QUAD	SIGNAL NAME	(TYPE)	DESCRIPTION
28	31	ATN	I/O(p/p)	ATTENTION: Sent by controller in charge. When true (low), interface commands are being sent over the DIO lines. When false (high), these lines carry data.
29	32	SRQ	I/O(p/p)	SERVICE REQUEST: Set true (low) by a device to indicate a need for service.
30	33	$\overline{\text{CONT}}$	O(p/p)	Indicates if a device is controller in charge. It is used to control direction of SRQ and ATN in pass control systems. Logically, it is (CIDS + CADS).
31 32 33 34 35 36 37 38	34 35 36 37 38 39 41 42	DIO8 DIO7 DIO6 DIO5 DIO4 DIO3 DIO2 DIO1	I/O(p/p) I/O(p/p) I/O(p/p) I/O(p/p) I/O(p/p) I/O(p/p) I/O(p/p) I/O(p/p)	DIO8 through DIO1 are the data input/output lines on the GPIB side. These pins connect to the IEEE-488 bus via non-inverting transceivers.
39	43	TR	O(p/p)	TRIGGER: Activated when the GET command is received over the interface or the fget command is given by the MPU.
40	44	V_{CC}		Supply voltage (+5 V nominal).

(p/p) = push/pull output.
(o/d) = open drain output with internal pull up.
*The hardware RESET pin has the following effect on the WD9914:
— Serial and Parallel Poll registers cleared.
—All clear/set auxiliary commands cleared except "swrst."
—"swrst" auxiliary command set. This holds the WD9914 in known states.

ARCHITECTURE

The block diagram of the internal architecture of the WD9914 is given in Figure 1. As previously stated, there are 13 MPU accessible registers of which six are read and seven are write. These registers handle all communication between the IEEE-488 1975/78 bus and microprocessor.

Each register is accessed by putting the relevant address on lines RS0, RS1 and RS2 and performing a memory read ($\overline{\text{WE}}$ = 1, DBIN = 1) or memory write ($\overline{\text{WE}}$ = 0, DBIN = 0) operation. The register addresses and use of each bit is shown in Table 1 for the read registers and Table 2 for the write registers. A full description of each register is given in the following paragraphs.

Implementation of the functions described by the state diagrams of the IEEE-488 standard is carried out in the IEEE-488 state diagram block. Information is received from the IEEE bus and from the internal registers and is combined with the current status of the device (for example, Talker Active State, TACS) to produce the control signals to load registers or handle the handshake or bus management lines.

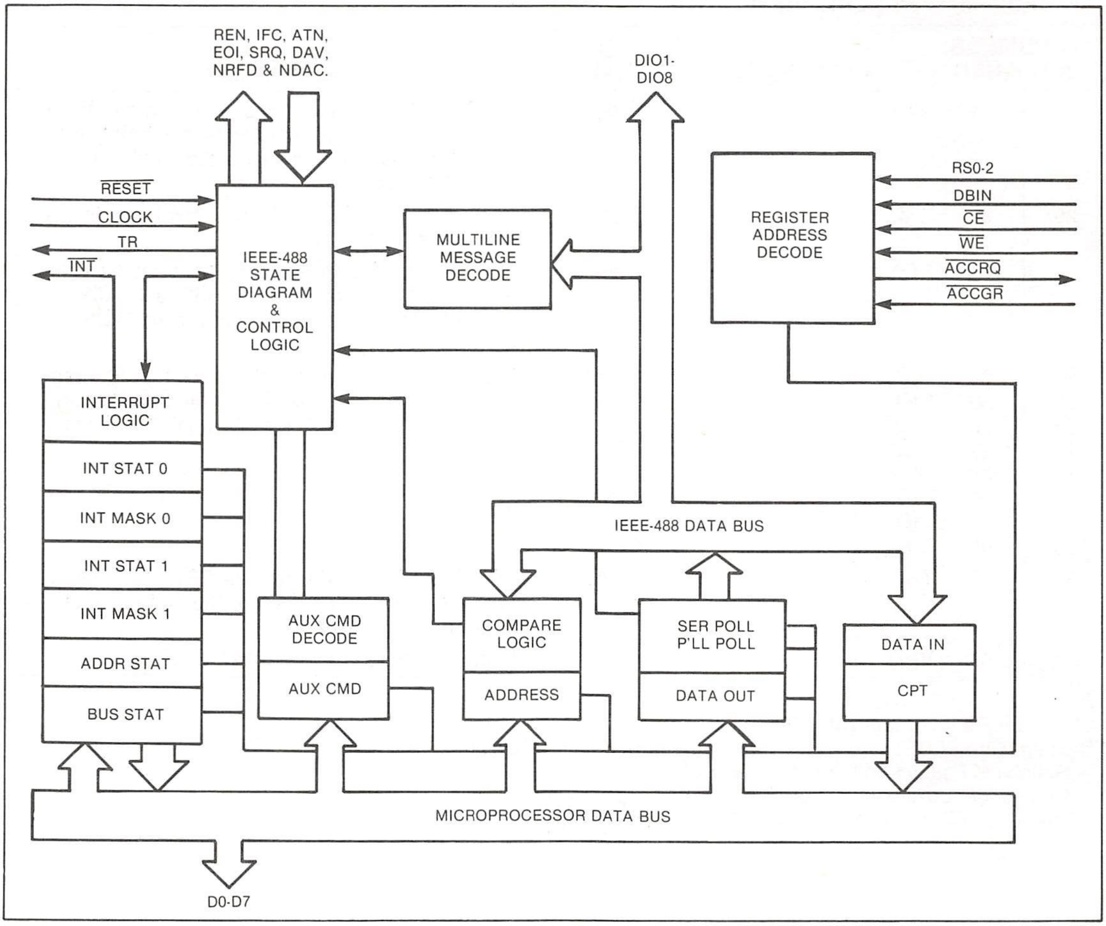


Figure 1. SIMPLIFIED BLOCK DIAGRAM

Table 1. WD9914 READ REGISTERS

ADDRESS				BIT ASSIGNMENT							
RS2	RS1	RS0	REGISTER NAME	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	Int Status 0	INT0	INT1	BI	BO	END	SPAS	RLC	MAC
0	0	1	Int Status 1	GET	ERR	UNC	APT	DCAS	MA	SRQ	IFC
0	1	0	Address Status	REM	LLO	ATN	LPAS	TPAS	LADS	TADS	ulpa
0	1	1	Bus Status	ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN
1	0	0	*								
1	0	1	*								
1	1	0	Cmd Pass Thru	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1
1	1	1	Data In	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

*The WD9914 host interface data lines will remain in the high impedance state when these register locations are addressed. An Address Switch register may therefore be included in the address space of the device at these locations.

Table 2. WD9914 WRITE REGISTERS

ADDRESS RS2	 RS1	 RS0	REGISTER NAME	 D0	 D1	 D2	BIT ASSIGNMENT D3	 D4	 D5	 D6	 D7
0	0	0	Int Mask 0			BI	BO	END	SPAS	RLC	MAC
0	0	1	Int Mask 1	GET	ERR	UNC	APT	DCAS	MA	SRQ	IFC
0	1	0	*	xx	xx	xx	xx	xx	xx	xx	xx
0	1	1	Auxiliary Cmd	cs	xx	xx	f4	f3	f2	f1	f0
1	0	0	Address	edpa	dal	dat	A5	A4	A3	A2	A1
1	0	1	Serial Poll	S8	rsvl	S6	S5	S4	S3	S2	S1
1	1	0	Parallel Poll	PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1
1	1	1	Data Out	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

*This address is not decoded by the WD9914. A write to this location will have no effect on the device, as if a write had not occurred.

REGISTERS

Interrupt Mask and Status Registers 0

The Interrupt Mask and Interrupt Status registers operate independently of each other. The status bits will always be set when the appropriate events occur, regardless of the state of the corresponding mask bit.

All interrupt bits (with the exception of INT0 and INT1, which are not storage bits) are edge triggered and are set when the appropriate condition becomes true. The storage bits are cleared immediately after the corresponding Interrupt Status register is read by the host MPU. If an interrupt condition becomes true during this read operation, then the event is stored. The corresponding bit is set when the read operation ends; hence no interrupts are lost. In addition to being cleared by a read operation, the BO interrupt is also cleared by writing to the Data Out register, and the BI interrupt is cleared by reading the Data In register.

The interrupt status bits are cleared and held in the O condition while Software Reset ("swrst") is set.

The corresponding bit of the Interrupt Mask register must be set to a 1 if an interrupt status bit is to cause an external interrupt ($\overline{\text{INT}}$ Low) when it is set (i.e., $\overline{\text{INT}}$ = $\overline{\text{INT STATUS.INT MASK}}$). The mask register is not cleared by "swrst" or the Hardware Reset pin ($\overline{\text{RESET}}$) and will power on in a random state. It must, therefore, be written to by the host MPU before "swrst" is cleared to avoid extraneous interrupts.

The INT0 and INT1 bits of the Interrupt Status register are not true status bits. Int1 will be true if there are any unmasked interrupt status bits set to a 1 in Interrupt Status register 1. INT0 will be true if any of bits 2-7 of Interupt Status Register 0 are unmasked and set to a 1. If either INT1 or INT0 is true, then the external interrupt pin ($\overline{\text{INT}}$) will be pulled low, provided that the Disable All Interrupts feature (dai) has not been set.

The individual bits of Interrupt Status and Interrupt Mask Register 0 are described in the following paragraphs. The conditions which set these bits, shown in parentheses, are given in terms of the state diagrams. Each bit is set on the rising edge of the condition shown.

INTERRUPT MASK/STATUS REGISTER 0

xx	xx	BI	BO	END	SPAS	RLC	MAC	INT	MASK 0
INT0	INT1	BI	BO	END	SPAS	RLC	MAC	INT	STATUS 0
D0	D1	D2	D3	D4	D5	D6	D7		MPU BUS

NOTE: A 0 masks and a 1 unmasks the bits in the interrupt mask registers.

INT1 This will be a 1 when an unmasked status bit in Interrupt Status Register 1 is set to a 1.

INT0 This will be a 1 when any of bits 2-7 of Interrupt Status Register 0 is unmasked and set to a 1.

BI Byte In. A data byte has been received in the Data In register. If the mask bit is not set, then no interrupt is generated but a RFD holdoff will still occur before the next data byte is accepted. If the Shadow Handshake feature is used, then this status bit will not be set. This bit is cleared by reading the Data In register as well as after Interrupt Status Register 0 has been read. (Set On: ACDS1.LACS)

BO Byte Out. This is set when the Data Out register is available to send a byte over the GPIB. This byte may be either a command if the device is a controller or data if the device is a talker. It is set when the device becomes an active talker or controller, but will not occur if the Data Out register has been loaded with a byte which has not been sent. Subsequently, it will occur after each byte has been sent and the WD9914 returns to SGNS. This bit is cleared by writing to the Data Out register as well as by reading Interrupt Status Register 0. (Set On: SGNS.CACS + SGNS.TASC.$\overline{SHFS}$)

NOTE:

When a controller addresses itself as a talker and then goes to standby, there will be a momentary transition of the source handshake into SIDS before TACS becomes true and it reenters SGNS. Under these circumstances, the WD9914 is guaranteed to give a BO interrupt on reentering SGNS.

END This indicates that a byte just received by a listener was the last byte in a string; that is, it was received with the EOI line true. It is set at the same time as the BI interrupt. (Set On: (ACDS1.LACS.EOI)

SPAS This indicates that the WD9914 has requested service via rsv1 or rsv2 (in the Serial Poll register or Auxiliary Command register) and has been polled in a serial poll. It is set on the false transition of STRS when the serial poll status byte is sent. (Set On: STRS.SPAS.(APRS1 + APRS2)

RLC Remote/Local Change. This is set by any transition between local and remote states in the Remote/Local function. (Set On: (LOCS-REMS) + (REMS-LOCS) + (LWLS-RWLS) + (RWLS-LWLS)

MAC My Address Change. This indicates that a command has been received from the GPIB which has resulted in the addressed state of the WD9914 changing. It will not occur if secondary addressing is being used, nor indicate that the WD9914 has been readdressed on its other primary address. (Set On: ACDS1.(MTA.TADSUNT + OTA.$\overline{TADS}$ + MLA.LADS + UN.$\overline{LADS}$)

Interrupt Mask and Status Registers 1

The operation of Interrupt Mask and Status Register 1 is similar to that of Interrupt Mask and Status Register 0 except that all bits are true storage bits. The status bits are cleared only following the register being read and by "swrst."

There is one distinct group of interrupts in this register: GET, UNC, APT, DCAS, MA. These are all set in response to commands received over the bus and if unmasked, a Data Accepted (DAC) holdoff will occur when the interrupt in question is set, it may be released with a "dacr" auxiliary command. This is further discussed in the Acceptor Handshake discussions.

The mask bit of the APT Interrupt is further used in the talker and listener functions. When the interrupt is unmasked, the talker and listener functions of the WD9914 implement the extended talker and extended listener functions of IEEE-488. Otherwise these functions implement the talker and listener functions of IEEE-488.

The individual bits of Interrupt Status and Interrupt Mask Register 1 are described below. The conditions which set these bits, shown in parentheses, are given in terms of the state diagrams.

INTERRUPT MASK/STATUS REGISTER 1

GET	ERR	UNC	APT	DCAS	MA	SRQ	IFC	INT	MASK 1
GET	ERR	UNC	APT	DCAS	MA	SRQ	IFC	INT	STATUS 1
D0	D1	D2	D3	D4	D5	D6	D7	MPU	BUS

GET This is set if a Group Execute Trigger command is received. A DAC holdoff occurs if the interrupt is unmasked. The TR pin becomes high when this command is received and persists high for the duration of a DAC holdoff if one occurs. If the interrupt is masked, the TR pin becomes high for approximately five clock cycles. (Set On: GET.LADS.ACDS1)

ERR Error. This is set if the source handshake becomes active and finds that the NDAC and NRFD lines are both high. This indicates that, for whatever reason, there are no acceptors on the bus. (Set On: SERS)

UNC Unrecognized Command. This is set if a command has been received which has no meaning to the WD9914. Unrecognized addressed commands will only cause this interrupt if the device is LADS, except for TCT, which will only interrupt in TADS. Secondary commands will only cause this interrupt if the "pts" auxiliary command has been set previously. A DAC holdoff will occur if this interrupt is unmasked which effectively enables the command pass through feature. Unrecognized commands may be inspected in the Command Pass Through register before this holdoff is released. (Set On: ACDS1. (UCG.$\overline{LLO. SPE}$.

$\overline{\text{SPD.DCL}}$ + ACG.$\overline{\text{GET.GTL.SCD.TCT.}}$ LADS + TCT.TADS + SCG.pts)

APT Address Pass Through. Unmasking this interrupt enables secondary addressing. It is set if a secondary command is received, provided that the last primary command received was a primary talk or listen address of the WD9914. A DAC holdoff will occur and the secondary address may be read from the Command Pass Through Register. The holdoff may be released by a "dacr" auxiliary command and the "cs" bit of the Auxiliary Command Register is used to indicate that a valid (cs = 1) or an invalid (cs = 0) secondary has been identified by the host MPU. (Set On: ACDS1. SCG.(LPAS + TPAS)

DCAS Device Clear Active State. This is set when a device clear command (DCL) is received or when a selected device clear (SDC) is received with the WD9914 in LADS. This will cause a DAC holdoff if unmasked. (Set On: ACDS1.(DCL + SDC.LADS)

SRQ Service Request. This is provided for the benefit of the controller, which should execute a serial poll in response to this interrupt. It is set when the SRQ line becomes true. (Set On: SRQ.$\overline{\text{(CIDS + CADS}}$)

MA My Address. This is set when the WD9914 recognizes its primary talk or listen address. A DAC holdoff will occur if this is unmasked. (Set On: (MLA + MTA).$\overline{\text{SPMS}}$. $\overline{\text{aptmk}}$)

IFC Interface Clear. This is provided for the benefit of devices which are not the system controller. It is set when the IFC line becomes true and indicates that the WD9914 has been returned to an idle state. If the device is the system controller, then the IFC interrupt is not set. (Set On: IFCIN)

Address Status Register

REM	LLO	ATN	LPAS	TPAS	LADS	TADS	ulpa		
D0	D1	D2	D3	D4	D5	D6	D7	MPU	BUS

REM	The device is in the remote state.
LLO	Local lockout is in operation.
ATN	The attention line is low (true) on the bus.
LPAS	WD9914 is in the listener primary addressed state.
TPAS	WD9914 is the talker primary addressed state.
LADS (or LACS)	The device is addressed to listen.
TADS (or TACS)	The device is addressed to talk.
ulpa	This bit shows the LSB of the last address recognized by the WD9914.

Address Register

edpa	dal	dat	A5	A4	A3	A2	A1
D0	D1	D2	D3	D4	D5	D6	D7

edpa	Enable dual primary addressing mode.
dal	Disable listener function.
dat	Disable Talker function.
A5-A1	Primary address of the WD9914.

Bits A5-A1 of this register contain the primary address of the device (denoted AAAAA in Table 19). IEEE-488 1975/78 does not allow a device to be assigned the value 11111 for bits A5-A1. When "swrst" is true at power-up or if set by the host MPU, the WD9914 is held in an idle state. During this time the host MPU may load the primary address of the device into these bits. Often this will be read from an Address Switch Register.

The "edpa" bit is used to enable the dual addressing mode of the WD9914. It causes the LSB of the address to be ignored by the address comparator giving two consecutive primary addresses for the device. The address by which the WD9914 was selected is indicated by the "ulpa" bit of the Address Status Register.

The Address Register is not cleared "swrst" or hardware reset.

Auxiliary Command Register

cs	xx	xx	F4	F3	F2	F1	F0
D0	D1	D2	D3	D4	D5	D6	D7

f4-f0	Auxiliary command select (see Table 3).
cs	Clear or set the feature (where applicable).

Auxiliary commands are used to enable and disable most of the selectable features of the WD9914 and to initiate many of the actions of the device. The desired feature is selected by writing a byte to this register with the appropriate value in bits f4-f0. These values are given in Table 3.

The c/s bit is used in most cases when the feature selected by f4-f0 is of the clear/set type. The feature is enabled if c/s = '1' and disabled if c/s = '0'. The holdoff on all data (hdfa) feature is an example of such a feature. Other auxiliary commands initiate an action of the WD9914, such as release RFD Holdoff (rhdf). In most cases, the c/s bit is unused and ignored by these commands.

All the clear/set auxiliary commands are cleared by the hardware RESET pin except "swrst," which is set true by RESET.

The force group execute trigger (fget) and return to local (rtl) auxiliary commands have a clear/set mode of operation and a pulsed mode of operation. They behave as normal clear/set features, but if they are written with c/s = '0' when they have not been previously set, then they will pulse true. Using the "fget" command in this manner will produce a pulse of approximately 1 μs at the TR pin (with a 5 MHz clock). The "rtl" command used in this way will cause a return to one of the local states (assuming local lockout is not in force), but the WD9914 may reenter the remote state next time the listen address occurs.

Table 3. AUXILIARY COMMANDS

c/s	f4	f3	f2	f1	f0	MNEMONIC	FEATURES
0/1	0	0	0	0	0	swrst	Software reset
0/1	0	0	0	0	1	dacr	Release DAC holdoff
na	0	0	0	1	0	rhdf	Release RFD holdoff
0/1	0	0	0	1	1	hdfa	Holdoff on all data
0/1	0	0	1	0	0	hdfe	Holdoff on end
na	0	0	1	0	1	nbaf	New byte available false
0/1	0	0	1	1	0	fget	Force group execute trigger
0/1	0	0	1	1	1	rti	Return to local
na	0	1	0	0	0	feoi	Send EOI with next byte
0/1	0	1	0	0	1	lon	Listen only
0/1	0	1	0	1	0	ton	Talk only
na	0	1	0	1	1	gts	Go to standby
na	0	1	1	0	0	tca	Take control asynchronously
na	0	1	1	0	1	tcs	Take control synchronously
0/1	0	1	1	1	0	rpp	Request parallel poll
0/1	0	1	1	1	1	sic	Send interface clear
0/1	1	0	0	0	0	sre	Send remote enable
na	1	0	0	0	1	rqc	Request control
na	1	0	0	1	0	rlc	Release control
0/1	1	0	0	1	1	dai	Disable all interrupts
na	1	0	1	0	0	pts	Pass through next secondary
0/1	1	0	1	0	1	std1	Set T1 delay
0/1	1	0	1	1	0	shdw	Shadow handshake
0/1	1	0	1	1	1	vstd1	Very short T1 delay
0/1	1	1	0	0	0	rsv2	Request Service Bit 2

DESCRIPTION OF AUXILIARY COMMANDS

Software Reset (swrst) 0/1xx00000

Setting this command causes the WD9914 to be returned to a known idle state during which it will not take part in any activity on the GPIB. This auxiliary command is set by the power-on RESET and the chip should be configured while "swrst" is set. Configuration should include writing the address of the device into the Address register, writing mask values into the Interrupt Mask registers and selecting the desired features in the Auxiliary Command register and Address register. After this, "swrst" may be cleared at which point the device becomes logically existent on the GPIB. The Serial Poll register and Parallel Poll registers may also be written in this period, but this is not necessary if there is no status to report as both of these are cleared by the power-on RESET pin. Table 4 lists the various states and other conditions forced by "swrst."

Table 4. SOFTWARE RESET CONDITIONS

MNEMONIC	DESCRIPTION
SIDS	Source idle state
AIDS	Acceptor idle state
TIDS	Talker idle state
TPAS	Talker primary idle state
LIDS	Listener idle state
LPAS	Listener primary state
NPRS	Negative poll response state
LOCS	Local state
CIDS	Controller idle state
SPIS	Serial poll idle state
PPSS	Parallel poll standby state
ADHS	DAC holdoff state
AEHS	RFD holdoff on end state
SHFS	Source holdoff state
ENIS	END idle state

NOTES: 1. See Section 3 for definition of above.
2. All interrupt status bits are held in a 0 state, but interrupt mask bits are not affected.

Release DAC Holdoff (dacr)0/1xx00001

The Data Accepted (DAC) holdoff allows time for the host microprocessor to respond to unrecognized commands, secondary addresses, and device trigger or device clear commands. The holdoff is released by the MPU when the required action has been taken. Normally the command is loaded with the clear/set bit at zero; however, when used with the address pass through feature c/s is set to one if the secondard address was valid, or to zero if invalid see APT interrupt.

Release RFD Holdoff (rhdf)naxx00010

Any Ready For Data (RFD) holdoff caused by a 'hdfa' or 'hdfe' is released.

Holdoff on All Data (hdfa)0/1xx00011

A Ready For Data (RFD) holdoff is caused on every data byte until the command is loaded with c/s set to zero. The handshake must be completed after each byte has been received by the MPU using the "rhdf" command.

Holdoff on End (hdfe)0/1xx00100.

An RFD holdoff will occur when an end of data string message (EOI true with ATN false) is received over the interface. This holdoff must be released using "rhdf."

Set New Byte Available False (nbaf)naxx00101

If a talker is interrupted before the byte just stored in the Data Out register is sent over the interface, this byte will normally be transmitted as soon as the ATN line returns to the false state. If, as a result of the interrupt, this byte is no longer required, its transmission may be suppressed using the "nbaf" command.

Force Group Execute Trigger (fget)0/1xx00110

The state of the TR output from the WD9914 is affected when this command is executed. If the c/s bit is zero, the line is pulsed high for approximately 5 clock cycles (1 μs at 5 MHz). If c/s is one, the TR line goes high until "fget" is sent with c/s equal to zero. No interrupts or handshakes are initiated.

Return to Local (rtl)0/1xx00111

Provided the local lockout (LLO) has not been enabled, the remote/local status bit is reset, and an interrupt is generated (if enabled to inform the host microprocessor that it should respond to the front panel controls). If the c/s bit is set to one, the "rtl" command must be cleared (c/s = 0) before the device is able to return to remote control. If c/s is set to zero, the device may return to remote without first clearing "rtl."

Force End or Identify (feoi)naxx01000

This command causes the EOI message to be sent with the next data byte. The EOI line is then reset.

Listen Only (lon)0/1xx01001

The listener state is activated until the command is set with c/s set to 0, or until deactivated by a bus command.

Talk Only (ton)0/1xx01010

The talker state is activated until the command is set with c/s set to 0, or until deactivated by a bus command.

NOTE:
"ton" and "lon" are included for use in systems without a controller. However, where the WD9914 is being used as a controller, it utilizes the "lon" and "ton" functions to set itself up as a listener or talker, respectively. Care must therefore be taken to ensure these functions are reset if sending UNL or OTA.

Go to Standby (gts)naxx01011

Issued by the controller in charge to set the ATN line false.

Take Control Synchronously (tcs)naxx01101

Control is again taken by the controller in charge, and ATN is asserted. If the controller is not a true listener, the shadow handshake command must be used to monitor the handshake lines so that the WD9914 is synchronous with the talker/listeners and only sends ATN true at the end of byte transfer. This ensures that no data is lost or corrupted.

Request Parallel Poll (rpp)0/1xx01110

This is executed by the controller in charge to send the parallel poll command over the interface. (The WD9914 must be in the Controller Active State so that the Attention line is asserted). The poll is completed by reading the Command Pass Through register to obtain the status bits, then sending 'rpp' with the cs bit at zero.

Take Control Asynchronously (tca)naxx01100

This command is used by the controller in charge to set the attention line true and to gain control of the interface. The command is executed immediately and data corruption or loss may occur if a talker/listener is in the process of transferring a data byte.

Send Interface clear (sic)0/1xx01111

The IFC line is set true when this command is sent with cs set to one. This must only be sent by the system controller and should be reset (cs = 0) after the IEEE minimum time for IFC has elapsed (100 μs). The system controller is put into the controller active state.

Send Remote Enable (sre)0/1xx10000

Issued by the system controller to set the REN line true and send the remote enable message over the interface, REN is set false by sending "sre" with cs at zero.

Request Control (rqc)naxx10001

When the TCT command has been recognized via the unidentified command pass through, this command is sent by the MPU. The WD9914 waits for the ATN line to go false and then enters the controller active state (CACS).

Release Control (ric)naxx10010

This command is used after TCT has been sent and handshake completed to release the ATN line and pass control to another device.

Disable All Interrupts (dai)0/1xx10011

The $\overline{\text{INT}}$ line is disabled, but the interrupt registers and any holdoffs selected are not affected.

Pass Through Next Secondary (pts)naxx10100

This feature may be used to carry out a remote configuration of a parallel poll. The parallel poll configure command (PPC) is passed through the WD9914 as an unrecognized addressed command and is identified by the MPU. The "pts" command is loaded, and the next byte received by the WD9914 is passed through via the Command Pass Through register. This would be the parallel poll enable (PPE), which is read by the microprocessor.

Set T1 Delay (std1)1xx10101

The T1 delay time can be set to six clock cycles (1.2 μs at 5 MHz) if this command is sent with the cs bit at one. The T1 delay time is 11 clock cycles (2.2 μs at 5 MHz) following a power-on reset, or if the command is sent with cs set to zero.

Shadow Handshake (shdw)0/1xx10110

This feature enables the controller in charge to carry out the listener handshake without participating in a data transfer. The Data Accepted line (DAC) is pulled true a maximum of three clock cycles after Data Valid (DAV) is received, and Not Ready For Data (NRFD) is allowed to go false as soon as DAV is removed.

The shadow handshake function allows the "tcs" command to be synchronized with the Acceptor Not Ready State (ANRS) so that ATN can be re-asserted without causing the loss or corruption of data byte. The END interrupt can also be received and causes a RFD holdoff to be generated.

Very Short T1 Delay (vstd1)0/1xx10111

If this feature is enabled, the GPIB setting time (T1) will be reduced to three clock cycles (600 ns at 5 MHz) on the second and subsequent data bytes when ATN is false. Otherwise, the GPIB setting time is determined by the "std1" feature.

Request Service Bit 2 (rsv2)0/1xx11000

The "rsv2" bit performs the same function as the "rsv1" bit, but provides a means of requesting service which is independent of the Serial Poll register.

This allows minor updates to be made to the Serial Poll register without affecting the state of the request service.

In addition, "rsv2" is cleared when the serial poll status byte is sent to the controller during a serial poll. It is therefore used in situations where a service request is simply a request from an instrument for the controller to poll its status. As soon as this happens, "rsv2" is cleared since the reason for requesting service has been satisfied. This eliminates the burden of clearing the bit from the host MPU, but also guarantees that "rsv2" is cleared before another serial poll can occur. If this were not so, there would be a possibility of a second status byte being sent with the RQS message true, which could result in confusion for the controller. ("rsv2" is cleared on: SPAS.(APRS1 + APRS2).STRS).

Bus Status Register

ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN	
D0	D1	D2	D3	D4	D5	D6	D7	MPU BUS

The host MPU may examine the status of the GPIB management lines at the time of reading.

The IFC bit of this register does not indicate a true value if the device is a system controller using the 'sic' auxiliary command.

Serial Poll Register

S8	rsv1	S6	S5	S4	S3	S2	S1	
DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	GPIB
D0	D1	D2	D3	D4	D5	D6	D7	MPU BUS

S8, S6-S0 Device status
rsv1 Request service bit 1

Bits S8, S6-S1 of this register are sent out over the GPIB when the device is addressed during a serial poll. They are cleared by a hardware reset but not by "swrst" and may, therefore, be set up during configuration of the chip. These bits are fully double buffered; and if the register is written to while the device is addressed during a serial poll (serial poll active state, SPAS), the value written is saved, and these bits are updated when SPAS is terminated.

The "rsv1" provides an input to the service request function of the WD9914, and is used to instruct this request that the controller service the device. When "rsv1" is set true, the SRQ line is pulled true on the GPIB, and the controller typically responds by setting up a serial poll to obtain the status of all instruments on the bus that may require service. When the WD9914 is addressed to send its status byte, SRQ is set false, and the status byte is sent with the RQS message true on DIO7. the "rsv1" bit must then be cleared and set true again if service is to be requested a second time. The SPAS interrupt is set immediately following the status byte being sent.

The "rev1" bit is also cleared by the hardware reset pin but not by "swrst." It is not double-buffered, but the service request function comprehends changes in the state of "rsv1" while the device is in SPAS. The Serial Poll register may therefore be written to any time.

Command Pass Through Register

DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	GPIB
D0	D1	D2	D3	D4	D5	D6	D7	MPU BUS

This provides a means of directly inspecting the GPIB data lines (DIO(8-1)). It has no storage and should only be used when the data lines are known to be in a steady state, such as during a DAC holdoff or in CPWS during a parallel poll. It is used to read unrecognized commands and secondaries following a UNC interrupt or to read secondary addresses following an APT interrupt. In addition, an active controller uses this register to read the results of a parallel poll at least $2\mu s$ after setting the "rpp" auxiliary command.

Parallel Poll Register

PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1	
DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	GPIB
D0	D1	D2	D3	D4	D5	D6	D7	MPU BUS

When a controller initiates a parallel poll, the contents of this register are presented to the GPIB data lines. If all bits of the register are cleared, then none of the lines (DIO(8-1) will be pulled low during a parallel poll, which corresponds to the Parallel Poll Idle State (PPIS) or IEEE-488. If it is desired to participate in a parallel poll, then the bit corresponding to the desired parallel poll response is set to a 1.

The Parallel Poll Register is fully double buffered. If it is written to during a parallel poll, the new value is held until the parallel poll ends, at which point the register is updated. This permits the host MPU to update the parallel poll response completely asynchronously to the GPIB.

If this register is cleared by the hardware RESET pin but not by "swrst," it may be loaded while the chip is being configured with "swrst" set.

Data In Register

DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	GPIB
D0	D1	D2	D3	D4	D5	D6	D7	MPU BUS

This register is used to hold data received by the WD9914 when it is a listener. It is loaded during Accept Data State (ACDS1) and, following this, an RFD holdoff will occur. This will normally be released when the byte is read by the host MPU; but if the Holdoff On All Data (hdfa) feature is selected, this holdoff must be released by the "rhdf" auxiliary command.

If the Holdoff On End (hdfe) feature is selected, the RFD holdoff will be released by reading the Data In register. But if the EOI line is true when the byte is received, reading the data byte will not release the holdoff and "rhdf" must be used.

As the Data In Register is loaded, the BI interrupt is set. The END interrupt is set simultaneously if the byte is accompanied by a true EOI line.

Data Out Register

DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	GPIB
D0	D1	D2	D3	D4	D5	D6	D7	MPU BUS

The Data Out register is used by a controller or talker for sending interface messages and device dependent messages. When the WD9914 enters the Talker Active State (TACS) or the Controller Active State (CASC), the contents of the Data Out register are presented to the GPIB data lines (DIO(8-1)), and the byte is sent over the bus under the control of the Source Handshake. Each time a byte is written, the source handshake is enabled, and the byte is sent. If the handshake is interrupted before the type can be sent, then it will be sent next time the Source Handshake becomes active unless a new byte available false (nbaf) auxiliary command is written. This has the effect of clearing an unsent byte from the Data Out register, and although the register itself is not cleared, the WD9914 behaves as if it had not been loaded.

Each time the source handshake becomes active and there is no unsent byte in the Data Out register, a BO interrupt will occur informing the host MPU that the Data Out register is available for use.

The Data In register and Data Out register operate independently. The Data Out register is not double buffered, and its contents are output directly to the data lines of the GPIB.

DIRECT MEMORY ACCESS

The WD9914 can operate in DMA using the $\overline{\text{ACCRQ}}$ (DMA request) and $\overline{\text{ACCGR}}$ (DMA grant) DMA handshake lines. The operation is automatic within the WD9914 and needs no "mpu" configuration.

The $\overline{\text{ACCRQ}}$ signal is set by (BO.$\overline{\text{CACS}}$ + B1); and can, therefore, not be used by a controller while ATN is asserted. It is reset by "swrst," reading data in register, writing to the data out register and $\overline{\text{ACCGR}}$. It is not cleared by reading interrupt status register 0.

If using DMA, the internal CE and addressing is disabled by the $\overline{\text{ACCGR}}$ signal going low and $\overline{\text{ACCGR}}$ will automatically address either the Data In register (DBIN = 0) or the Data Out register (DBIN = 1).

NOTE:
The sense of DBIN is inverted for DMA operation.

At the end of a DMA read from memory sequence, the $\overline{\text{ACCRQ}}$ will be left low (also BO bit set). It may be necessary for the "mpu" to clear this in some circumstances; e.g., starting DMA write to memory sequence.

In DMA it is recommended that the MA interrupt be unmasked to prevent errors due to interrupted data streams.

If DMA is not being utilized, the $\overline{\text{ACCGR}}$ signal must be held high. In this case, the ACCRQ signal can be used as a separate interrupt line for BO and BI. This allows faster "mpu" transfers to take place as it is not necessary to read the interrupt register to find the cause of the interrupt. Figure 2 shows a typical DMA configuration.

TERMINAL ASSIGNMENTS AND FUNCTIONS

The IEEE-488 standard uses the negative logic convention for the GPIB lines. The FALSE state (0) is represented by a high voltage (>2.0 V); the TRUE state (1) is represented by a low voltage (>0.8 V). The GPIB terminations of the WD9914 are in agreement with this convention. For example, if Data Valid is true (1), the DAV line is pulled low by the device. These terminations are connected to the bus via noninverting buffers to obtain the correct signal polarity.

Note that the terminations on the microprocessor side of the device are in positive logic (true state (1) = high voltage : false state (0) = low voltage). This is in agreement with the logic convention used by most microprocessors. Thus if:

DO(MSB) D7(LSB)

0	1	1	0	1	0	0	1

is written into the Data Out register, it will appear as:

DIO8(MSB) DIO1(LSB)

HIGH	LOW	LOW	HIGH	LOW	HIGH	HIGH	LOW

on the IEEE-488 D10 lines.

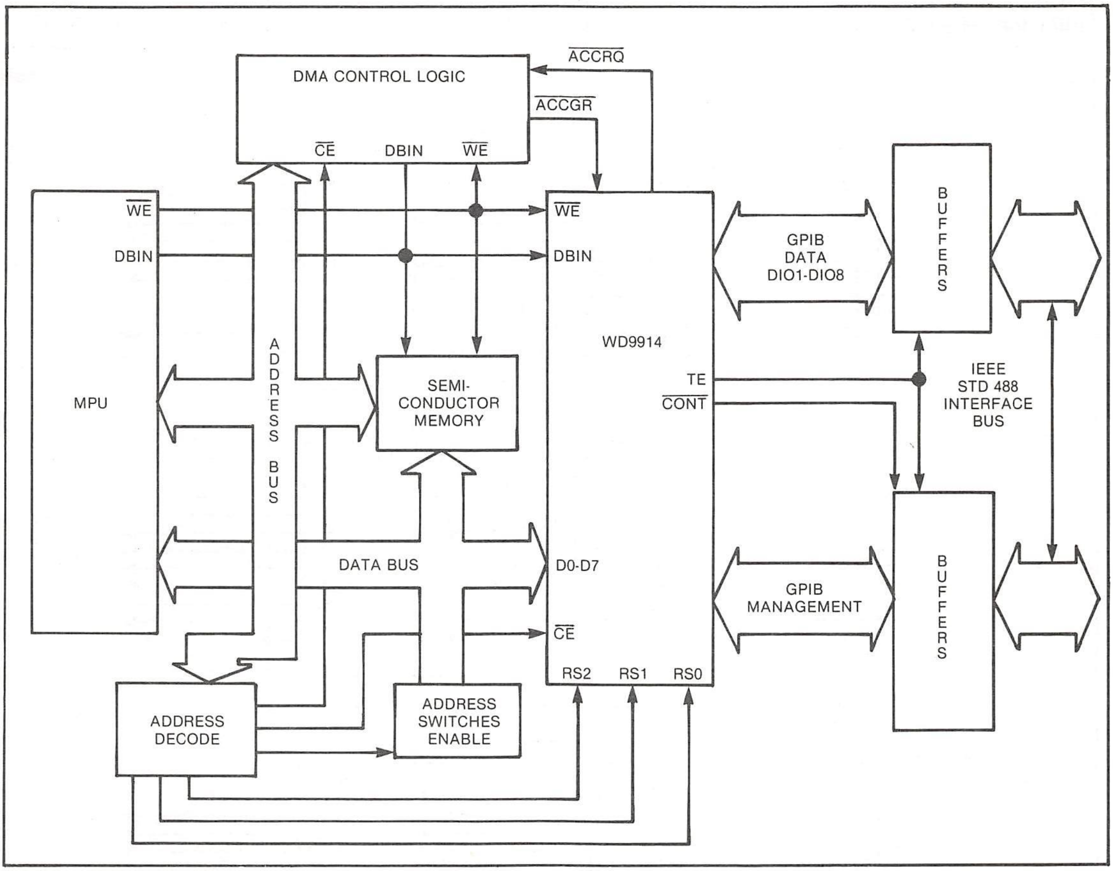


Figure 2. DMA CONFIGURATION

TRANSCEIVER CONNECTIONS

There are three linear transceivers designed to work with the WD9914: The SN75160, SN75161, and SN75162. Figure 3 shows the possible transceiver connections. Note that there is a corresponding pin-out between the WD9914 and the transceivers. This allows the whole GPIB interface to be laid out in a very small area of printed circuit board.

The SN75160 is a 20-pin device used to buffer the IEEE-488 data lines (DIO(8-1)) in all applications. The direction of the buffers is controlled by the Talk Enable (TE) output of WD9914. This active high signal becomes true whenever there is an interface function of the WD9914 not sending the NUL message on DIO(8-1); that is, when the device is in TACS, CACS, SPAS, or PPAS. The Pull-Up Enable (PE) input of the SN75160 is an active high input which selects whether the 'DIO(8-1)' lines are driven by open collector or push/pull buffers. A push/pull buffer is required if faster data rates are required and the "stdl" and/or the "vstdl" features are used. Open collectors must be used if parallel polling is being used in a particular GPIB environment. If only one of these features is desired, the PE input may be hardwired; otherwise it must be derived from ATN and EIO, as shown in Figure 3.

The SN75161 is a 20-pin device used to buffer the IEEE-488 management lines. It may be used for a talker/listener device or for a controller which does not pass control. The direction of the handshake line buffers NRFD, NDAC, DAV are again controlled by the TE signal. However, the SRQ, ATN, REN, and IFC buffers are controlled by the DC input of the SN75161, which connects to the Controller Active (CONT) output of the WD9914. CONT becomes low whenever the WD9914 is an active controller; that is, when it is not in CIDS or CADS. The SN75161 also includes the logic necessary to control the direction of the EOI buffer. This is dependent on the TE signal when ATN is false (high), and the DC signal when ATN is true (low).

The SN75162 is a 22-pin device which may be used to buffer the IEEE-488 management lines in all applications including devices which pass control. The SN75162 has a separate pin to control the direction

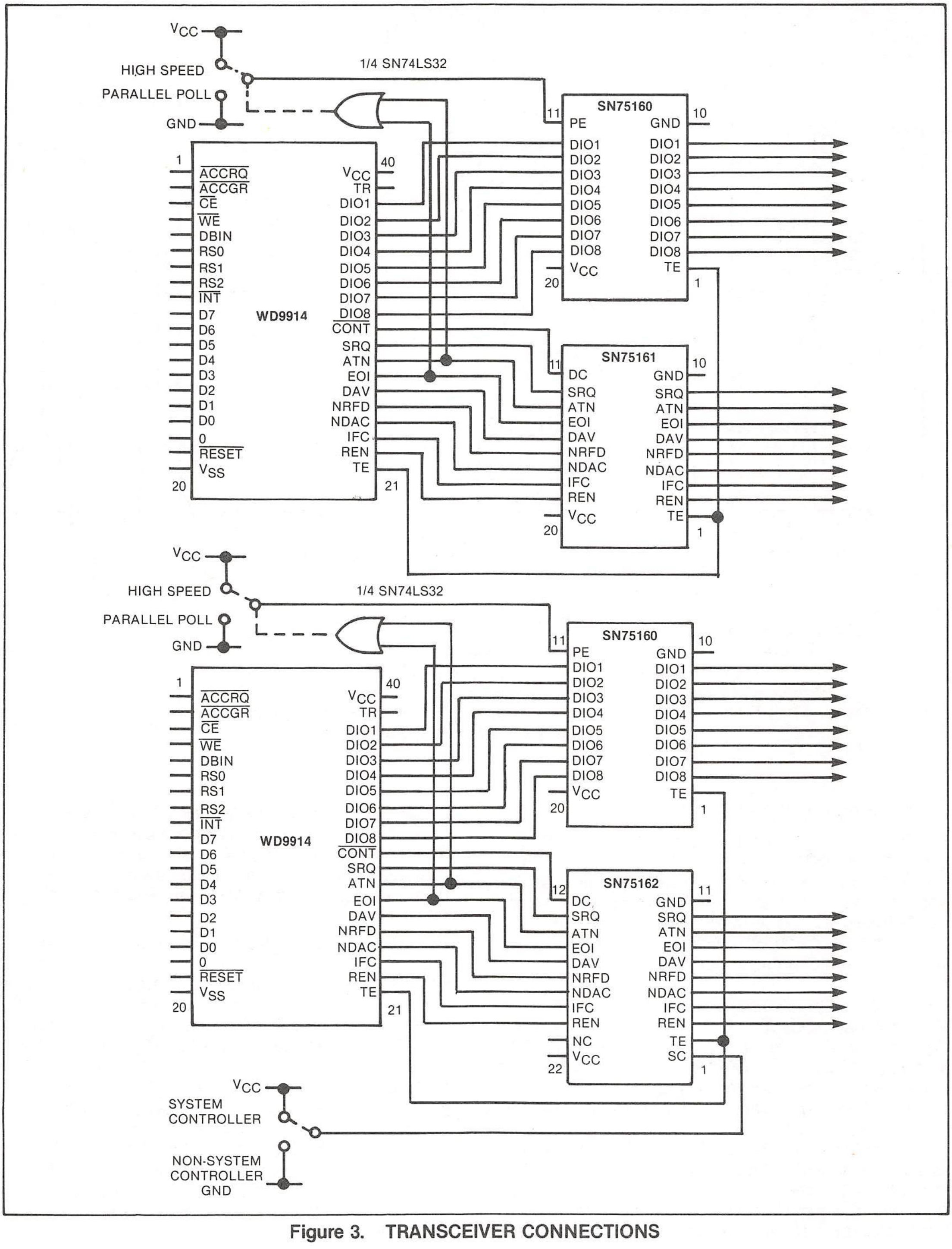


Figure 3. TRANSCEIVER CONNECTIONS

of the REN and IFC buffers, but is otherwise identical to the SN75161 in all other respects. This input is the System Controller input (SC) which may be hardwired or switchable to determine whether or not the instrument in question is a system controller or not. Note that a device which has its buffers configured as a non-system controller should never use the "sic" and "sre" auxiliary commands.

STATE DIAGRAM IMPLEMENTATION

This section presents the state diagrams for the WD9914.

Where equivalent, the names of WD9914 states are the same as those of IEEE-488. In some cases, IEEE-488 states have been divided, for example, ACDS of the IEEE-488 has been split into ACDS1 and ACDS2. The convention of lower case characters for local messages and upper case for remote messages and interface states is retained.

State diagrams with remote message outputs are supplemented with tables. T is used to represent a true output and F a false output. Parentheses denote a passive output; otherwise, it is active. The outputs shown are the values presented to the bus and assume the use of the SN75160 and SN75161 or SN75162 transceivers or their logical equivalents. The symbol (NUL) associated with DIO(1-8) indicates that each of these lines is sent passive false by the function in question.

NOTE:
An arrow into a state with no state as its origin represents a transition from every other state on the diagram. Note, however, that this does not imply that all exit conditions from the destination state are overridden. If such an entry condition is true and, simultaneously, an exit condition is true then this represents an illegal situation and should be avoided. Such situations will not occur in normal operation of the device.

No maximum timings are discussed. The WD9914 with its recommended transceivers meets all IEE-488 maximum timing requirements. If the WD9914 is used with other transceivers, then it must be ensured that these requirements are still met.

AUXILIARY COMMANDS

There are two basic types of commands implemented in the auxiliary command register: immediate execute and clear/set.

The clear/set commands are used to enable and disable the various features of the WD9914. The particular feature is selected by the code on f0-f4, and it is set or cleared according to the value on the cs bit. For the purposes of the state diagrams, the mnemonic of a clear/set command simply represents its current state.

The immediate execute auxiliary commands remain active for the duration of a strobe signal after the Auxiliary Command register has been written to. This is represented in the form of a state diagram in Figure 4. Note that writes to the Auxiliary Command register must be spaced by at least five clock cycles. For the purposes of the remaining state diagrams, the immediate execute commands are represented as the mnemonic gated by the auxiliary command strobe state (AXSS).

The clear/set bit of the Auxiliary Command register is used by several of the immediate execute commands. For example, "dacr" uses it to differentiate between valid and not valid secondary addresses when releasing a DAC holdoff on a secondary address. The "lon" and "ton" auxiliary commands are also considered immediate execute.

The "fget" and "rtl" auxiliary commands are both immediate execute and clear/set. They may be cleared or set in the normal way; but if they are cleared when they are already in the false state, they will pulse true for the duration of AXSS. In the following state diagrams, however, these are simply included in their clear/set form.

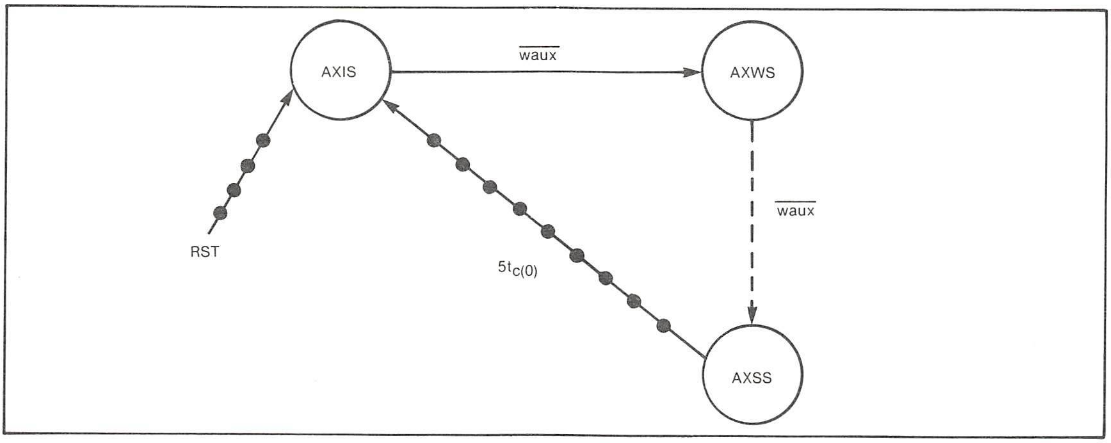


Figure 4. WD9914 AUXILIARY COMMAND STATE DIAGRAM

Table 5. AUXILIARY COMMAND STATE DIAGRAM MNEMONICS

MESSAGES			STATES		
waux	=	write to auxiliary command register	AXIS	=	auxiliary command register idle state
$t_{c(0)}$	=	clock cycle time	AXWS	=	auxiliary command write state
			AXSS	=	auxiliary command strobe state

ACCEPTOR HANDSHAKE

The WD9914 acceptor handshake is shown in Figure 5. The main variation from IEEE-488 to note is that the device remains in AIDS while the controller function is in CACS. The WD9914, therefore, does not monitor the commands which it sends over the bus and this places some restrictions on the user.

The accept data state of IEEE-488 (ACDS) is divided into two states. The first, (ACDS1) is used to strobe data into the Data In register or to sequence the decoding of commands from the bus. All interrupts generated by the acceptor handshake (GET, MA, MAC, DCAS, APT, UCG, BI, and END) are generated by this state. The second (ACDS2) is used as a holding state where the device will remain in the event of a DAC holdoff.

Certain of the commands will cause interrupts in ACDS1, and if the interrupts are unmasked, a DAC holdoff will occur. The interrupts concerned are GET, MA, DCAS, UCG, and APT. This is represented in the state diagram by the signal SAHF, which becomes true when one of the above interrupts is set if it is unmasked. It persists for the duration of ACDS1. This event is stored by causing the ADHS to become active, which inhibits the transition from ACDS2 to AWNS. ADHS is cleared by "dacr." Table 19 shows the response of the WD9914 to the various bus commands.

If a GET command is received in ACDS1, then the TR pin will be set high. This high condition persists throughout ACDS1 and ACDS2, which means that if a DAC holdoff occurs, the TR pin will remain high until the holdoff is released by a "dacr" auxiliary command.

Two additional state diagrams are included to record

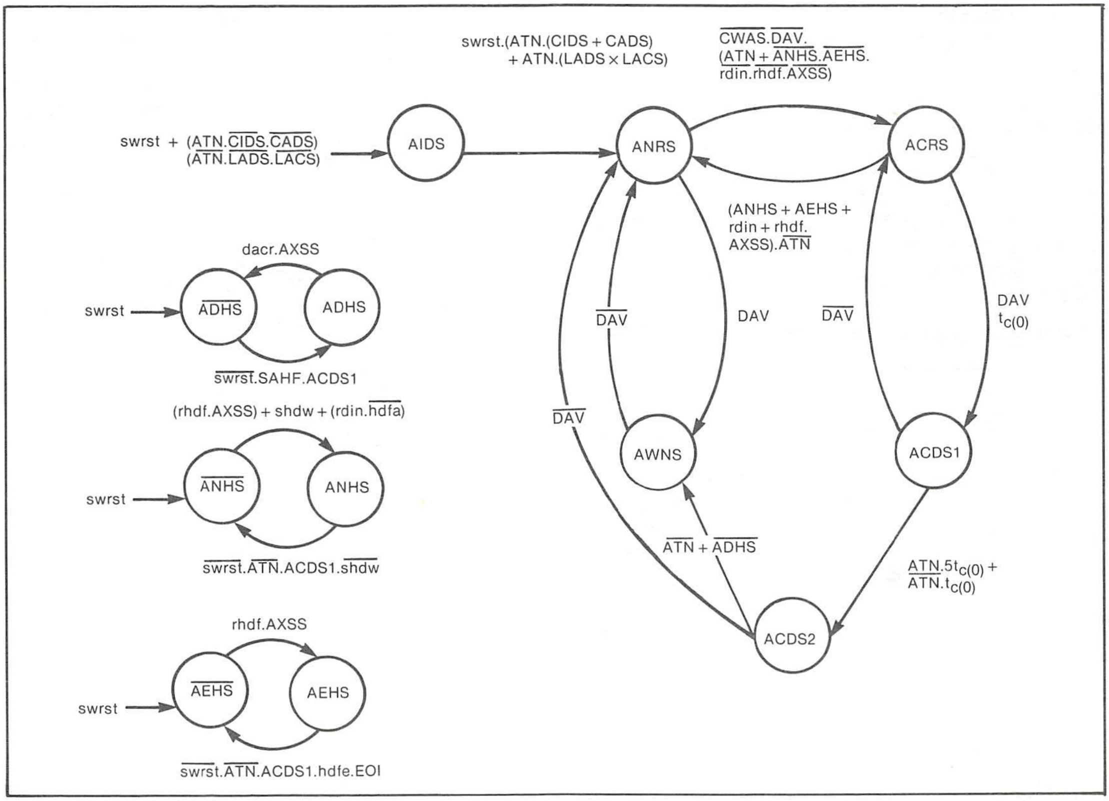

swrst.(ATN.(CIDS + CADS)
+ ATN.(LADS × LACS)
$\overline{CWAS}.\overline{DAV}.$
$(ATN + \overline{ANHS}.\overline{AEHS}.$
$\overline{rdin}.\overline{rhdf}.\overline{AXSS})$
swrst + $(ATN.\overline{CIDS}.\overline{CADS})$
$(ATN.\overline{LADS}.\overline{LACS})$
AIDS
ANRS
ACRS
(ANHS + AEHS +
rdin + rhdf.
$AXSS).\overline{ATN}$
dacr.AXSS
swrst
$\overline{ADHS}$
ADHS
$\overline{swrst}$.SAHF.ACDS1
$\overline{DAV}$
DAV
$\overline{DAV}$
DAV
$t_{c(0)}$
$\overline{DAV}$
AWNS
ACDS1
(rhdf.AXSS) + shdw + $(rdin.\overline{hdfa})$
swrst
$\overline{ANHS}$
ANHS
$\overline{ATN} + \overline{ADHS}$
$\overline{swrst}.\overline{ATN}.ACDS1.\overline{shdw}$
$\overline{ATN}.5t_{c(0)}$ +
$ATN.t_{c(0)}$
ACDS2
rhdf.AXSS
swrst
$\overline{AEHS}$
AEHS
$\overline{swrst}.\overline{ATN}$.ACDS1.hdfe.EOI

Figure 5. WD9914 ACCEPTOR HANDSHAKE STATE DIAGRAM

the type of data received in ACDS1 when ATN is false. ANHS indicates that a data byte has been received and that an RFD holdoff should be caused before the next data byte is accepted. The holdoff may be released by reading the Data In register unless the "hdfa" feature is enabled, in which case "rhdf" must be used. AEHS shows that the last data byte was accepted with the EOI message true and the "hdfe" feature set. This will cause an RFD holdoff which can only be released by "rhdf."

Table 6. ACCEPTOR HANDSHAKE MNEMONICS

MESSAGES			STATES		
swrst	=	software reset	AIDS	=	acceptor idle state
dacr	=	DAC release	ANRS	=	acceptor not ready state
rhdf	=	release RFD holdoff	ACRS	=	acceptor ready state
shdw	=	shadow handshake	ACDS1	=	accept data state 1
rdin	=	read data in register	ACDS2	=	accept data state 2
hdfe	=	enable RFD holdoff after END messages received	AWNS	=	acceptor wait for new cycle state
hdfa	=	enable RFD holdoff on all data	ADHS	=	accept data holdoff state
ATN	=	attention	ANHS	=	acceptor not ready holdoff state
DAV	=	data valid	AEHS	=	acceptor not ready holdoff after 'END'
EOI	=	end or identify state	CWAS	=	controller wait for ANRS state (controller function)
RFD	=	ready for data	AXSS	=	auxiliary command strobe state (auxiliary command register)
DAC	=	data accepted	LADS	=	listener addressed state (listener function)
SAHF	=	set accept data holdoff state	LACS	=	listener active state (listener function)
$t_{c(0)}$	=	clock cycle time	CIDS	=	controller idle state (controller function)
			CADS	=	controller addressed state (controller function)

Table 7. ACCEPTOR HANDSHAKE MESSAGE OUTPUTS

STATE	REMOTE MESSAGES SENT		OTHER ACTIONS
	RFD	DAC	
AIDS	(T)	(T)	
ANRS	F	F	
ACRS	(T)	F	
ACDS1	F	F	ATN False: — data entered into Data In register — BI interrupt generated — end interrupt generated if EOI is true. ATN true: — commands decoded — command related interrupts set — "sahf" set if command requires a DAC holdoff — TR pin set true if GET message is received — "pts" feature cleared after UNC interrupt set
ACDS2	F	F	TR — pin set true if GET command was received in ACDS1
AWNS	F	(T)	

SOURCE HANDSHAKE

The WD9914 source handshake state diagram is shown in Figure 6. IEEE-488 states SIWS and SWNS have been removed. These record the false then true transition of "nba" (new byte available) as the old data byte is removed and a new data byte is made ready. Instead the WD9914 uses a separate state (SHFS) to record the availability of a data byte in the Data Out register. This state is exited when a byte is written to the Data Out register, which enables the transition from SGNS to SDYS and the subsequent transmission of the byte. The SHFS is reentered as the byte is sent in STRS; but if the handshake is interrupted before this, then the fact that the byte has not been sent is recorded until the source handshake again becomes active. If, however, the byte in the Data Out register is to be disregarded, then "nbaf" may be used to return the device to SHFS.

The status byte in the Serial Poll register is continually available. The transition from SGNS to SDYS is not dependent on SHFS during a serial poll, that is, while SPAS is active. By separately recording the availability of a byte in the Data Out register, a talker sending data may be interrupted for a serial poll without risk of a byte being lost.

The additional state SERS is included to detect an error condition on the bus. This will be entered when the source handshake tries to send a byte, but finds both the NRFD and NDAC lines false at the same time. This condition will normally indicate for a controller that there are no devices powered up on the bus, or for a talker that there are no devices addressed to listen on the bus.

The state VSTS will be entered after the first data byte of a talker has been sent if the "vstdl" feature is enabled. This enables a very short bus settling time $(4t_{c(0)})$ for all subsequent bytes until ATN next becomes true. The WD9914 will not use the short bus settling time when it is an active controller.

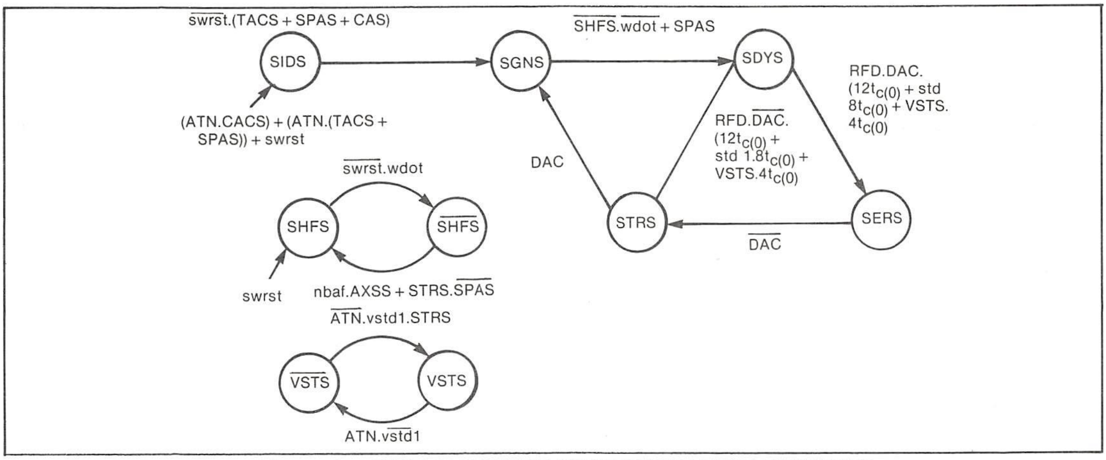


Figure 6. WD9914 SOURCE HANDSHAKE STATE DIAGRAM

Table 8. SOURCE HANDSHAKE MNEMONICS

MESSAGES			STATES		
swrst	=	software reset	SIDS	=	source idle state
nbaf	=	new byte available false	SGNS	=	source generate state
wdot	=	write to the data out register	SDYS	=	source delay state
stdl	=	enable short bus settling time	SERS	=	source error state
vstdl	=	enable very short bus settling time	STRS	=	source transfer state
ATN	=	attention	SHFS	=	source holdoff state
RFD	=	ready for data	VSTS	=	very short bus settling time state
DAC	=	data accepted	TACS	=	talker active state (talker function)
$t_{c(0)}$	=	clock cycle time	CACS	=	controller active state (controller function)
			SPAS	=	serial poll active state (talker function)
			AXSS	=	auxiliary command strobe state (auxiliary command register)

Table 9. SOURCE HANDSHAKE MESSAGE OUTPUTS

STATE	REMOTE MESSAGES SENT	OTHER ACTIONS
	DAV	
SIDS	(F)	BO interrupt and ACCRQ set true if SHFS is false and SPAS is not true
SGNS	F	
SDYS	F	
SERS	F	ERR interrupt set true
STRS	T	

TALKER AND LISTENER FUNCTIONS

Figures 7 and 8 show the WD9914 listener and talker state diagrams, which serve the purpose of the listener and talker or extended listener and extended talker functions of IEEE-488, depending on the state of the APT interrupt mask bit.

The WD9914 does not recognize secondary addresses on-chip and these must be passed through to the host MPU for verification. Secondary addressing is enabled by unmasking the APT interrupt. A secondary address will cause this interrupt if the last primary command received was a primary address of the device; that is, it is in TPAS or LPAS. A DAC holdoff will also occur. The host MPU must respond to the interrupt by reading the secondary from the Command Pass Through register and identifying it as being valid or not valid. The holdoff may then be released with a "dacr" auxiliary command, the sense of the cs bit being used to indicate a valid (cs = 1) or not valid (cs = 0) secondary. If a valid secondary address is indicated, then the WD9914 will enter TADS or LADS, depending on whether it is in TPAS or LPAS.

The "lon" and "ton" auxiliary commands together with the clear/set bit (cs) have a direct influence on the appropriate state diagrams. Therefore, although they appear as ordinary clear/set auxiliary commands, they can be effectively cleared by other bus events. For example, if a WD9914 addresses itself as a listener via the "lon" command, it may be returned to LIDS by an UNL command from the bus at a later time.

The "lon" and "ton" auxiliary commands are used to implement two features of IEEE-488. First, talk only and listen only are used in situations where there is no active controller on the bus. Note that the "lon" and "ton" commands are linked with these features to indicate to the user that these commands are not enabled by CAS, as are "ltn" and "lun" of IEEE-488.

Second, the "lon" and "ton" auxiliary commands are used by an active controller to address itself. IEEE-488 provides for a controller to address itself to listen via the "ltn" and "lun" message, but there is no corresponding message for the talker. Hence, when a controller addresses itself to talk via "ton," it must send its talk address over the bus and; similarly, if it sends another talk address over the bus then it must unaddress itself by writing "ton" false.

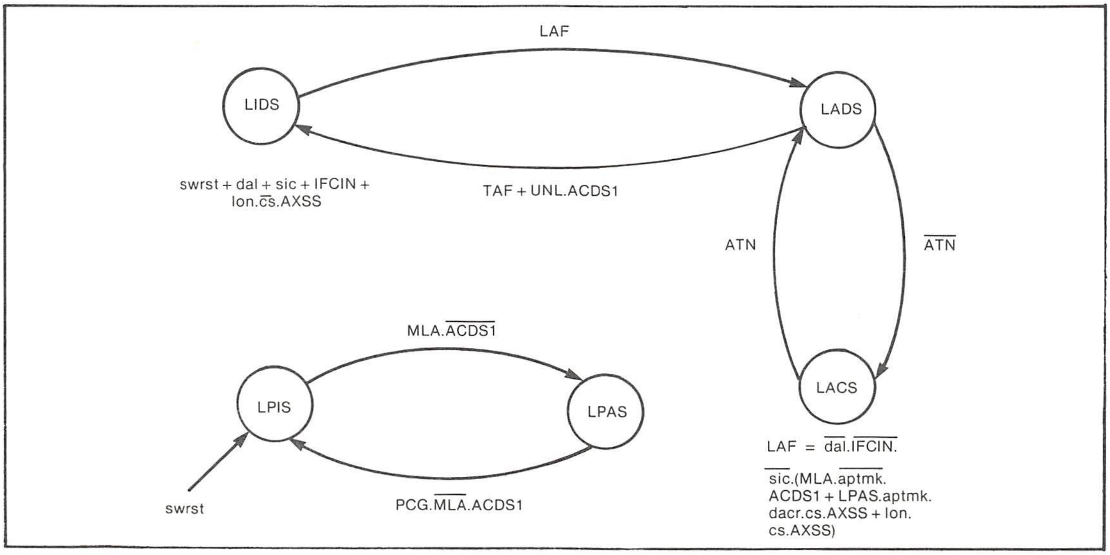
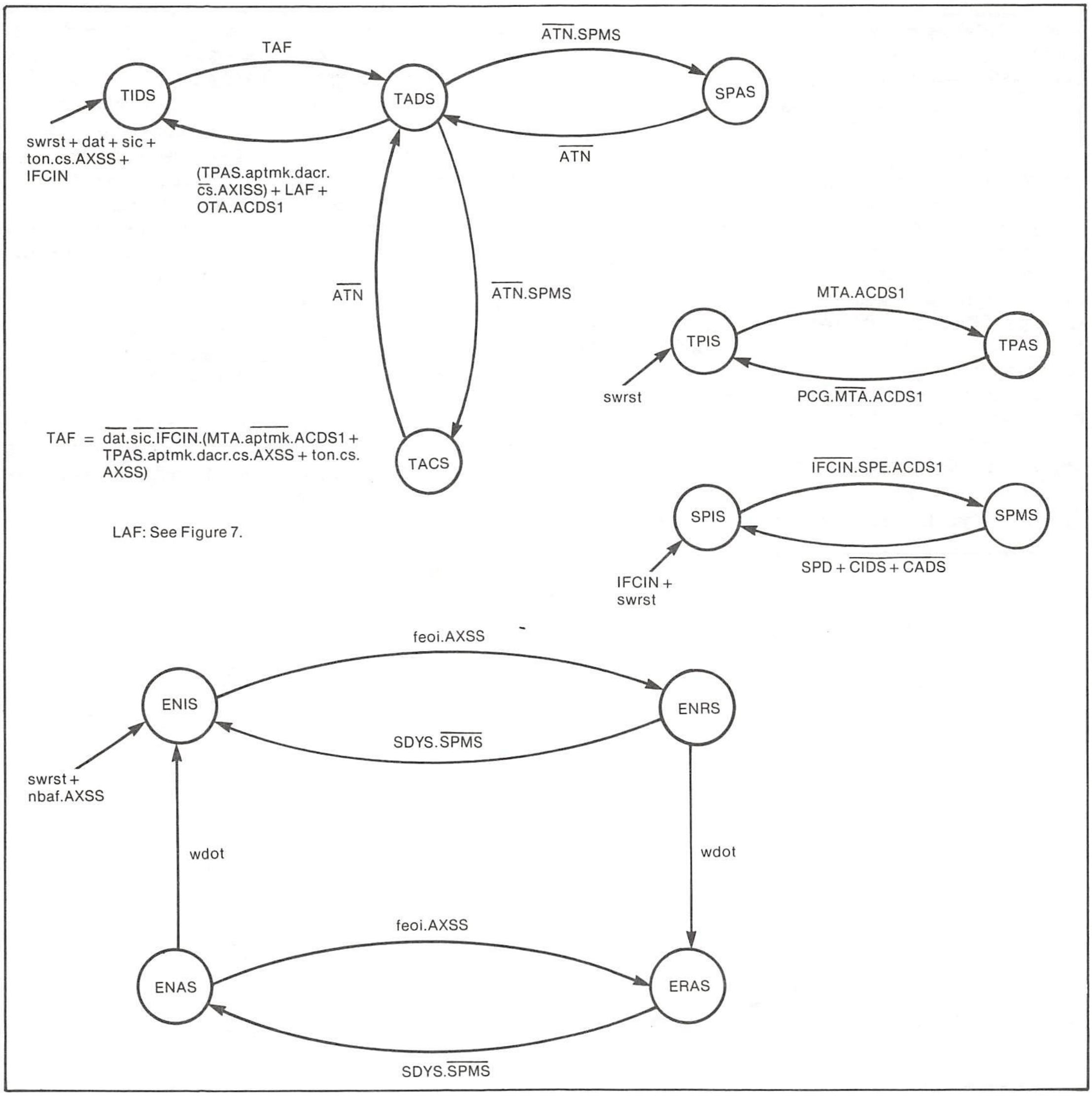

LAF
LIDS
LADS
swrst + dal + sic + IFCIN + lon.$\overline{cs}$.AXSS
TAF + UNL.ACDS1
ATN
$\overline{ATN}$
MLA.$\overline{ACDS1}$
LPIS
LPAS
LACS
swrst
PCG.$\overline{MLA}$.ACDS1
LAF = $\overline{dal}$.$\overline{IFCIN}$.
$\overline{sic}$.(MLA.$\overline{aptmk}$.
ACDS1 + LPAS.aptmk.
dacr.cs.AXSS + lon.
cs.AXSS)

Figure 7. WD9914 LISTENER STATE DIAGRAM

When the WD9914 enters SPAS, the contents of the Serial Poll register are sampled, and presented on DIO(8-1). These will remain unchanged until SPAS is exited. The source handshake will, however, send this status byte as many times as the controller will accept it.

The internal IFC signal of the WD9914 (IFCIN) is suppressed when the device itself is sending IFC in order to simplify implementation of the controller function. Therefore, the send interface clear (sic) auxiliary command is included with IFCIN to return the talker and listener functions to their idle states and allow a system controller to clear its own interface.

A separate state diagram is included to control the sending of the END message of IEEE-488. If the "feoi" auxiliary command is written followed by loading a byte into the Data Out register, the WD9914 will enter ERAS, and the EOI line will be asserted as 'DIO(8-1)' begin to change. The function will enter ENAS as soon as the source handshake begins to send this byte, and EOI will be released when the Data Out register is next loaded. If it is desired to send EOI true with the next byte as well, then "feoi" may be written before the Data Out register returns the device to ERAS.


TAF
TIDS
TADS
$\overline{\text{ATN}}$.SPMS
SPAS
swrst + dat + sic +
ton.cs.AXSS +
IFCIN
(TPAS.aptmk.dacr.
$\overline{\text{cs}}$.AXISS) + LAF +
OTA.ACDS1
ATN
$\overline{\text{ATN}}$
$\overline{\text{ATN}}$.SPMS
TACS
MTA.ACDS1
TPIS
TPAS
swrst
PCG.$\overline{\text{MTA}}$.ACDS1
$\overline{\text{IFCIN}}$.SPE.ACDS1
SPIS
SPMS
IFCIN +
swrst
SPD + $\overline{\text{CIDS}}$ + $\overline{\text{CADS}}$
feoi.AXSS
ENIS
ENRS
SDYS.$\overline{\text{SPMS}}$
swrst +
nbaf.AXSS
wdot
wdot
feoi.AXSS
ENAS
ERAS
SDYS.$\overline{\text{SPMS}}$

TAF = $\overline{\text{dat}}$.$\overline{\text{sic}}$.$\overline{\text{IFCIN}}$.(MTA.$\overline{\text{aptmk}}$.ACDS1 + TPAS.aptmk.dacr.cs.AXSS + ton.cs. AXSS)

LAF: See Figure 7.

Figure 8. WD9914 TALKER STATE DIAGRAM

Table 10. TALKER AND LISTENER MNEMONICS

MESSAGES			STATES		
swrst	=	software reset	LIDS	=	listener idle state
dal	=	disable listener	LADS	=	listener addressed state
dat	=	disable talker	LACS	=	listener active state
sic	=	send interface clear	LPIS	=	listener primary idle state
lon	=	listen only	LPAS	=	listener primary addressed state
ton	=	talk only	TIDS	=	talker idle state
c/s	=	clear/set bit of the auxiliary command register	TADS	=	talker addressed state
dacr	=	release 'DAC' holdoff	TACS	=	talker active state
aptmk	=	address pass through interrupt mask	SPAS	=	serial poll active state
nbaf	=	new byte available false	SPIS	=	serial poll idle state
feoi	=	force 'EOI'	SPMS	=	serial poll mode state
wdot	=	write to the Data Out register	TPIS	=	talker primary idle state
ATN	=	attention	TPAS	=	talker primary addressed state
IFCIN	=	internal interface clear message (a debounced signal, suppressed by 'sic')	ENIS	=	end idle state
EOI	=	end or identify	ENRS	=	end ready state
PCG	=	primary command group	ERAS	=	end ready and active state
MLA	=	my listen address	ENAS	=	end active state
MTA	=	my talk address	SDYS	=	source delay state (source handshake)
OTA	=	other talk address	CIDS	=	controller idle state (controller function)
SPE	=	serial poll enable	CADS	=	controller addressed state (controller function)
SPD	=	serial poll disable	ACDS1	=	accept data state 1 (acceptor handshake)
UNL	=	unlisten	AXSS	=	auxiliary command strobe state (auxiliary command register)
PCG	=	primary command group			

Table 11. TALKER FUNCTION MESSAGE OUTPUTS

STATE	QUALIFIER	REMOTE MESSAGES SENT		OTHER ACTIONS DIO(8-1)
		RQS	EOI	
TIDS		(F)	(F)	(NUL)
TADS		(F)	(F)	(NUL)
TACS	ENIS.ENRS	(F)	F	DATA OUT REG
TACS	ENAS.ERAS	(F)	T	DATA OUT REG
SPAS	NPRS.SRQS	F	F	SERIAL POLL REG
SPAS	APRS1.APRS2	T	F	SERIAL POLL REG

SERVICE REQUEST FUNCTION

Figure 9 shows the state diagram for the WD9914 service request function. The device has two means of implementing the request service (rsv) local message of IEEE-488: the first, "rsv1," is bit 7 of the Serial Poll register; the second is the auxiliary command "rsv2." These are simply ORed together to provide an input to the service request function; and, in any particular application, only one would normally be used, the other being left in its hardware reset state.

The affirmative poll response state (APRS) of IEEE-488 is split into two states on the WD9914 for the following reason: Consider the case where a device has requested service, has been serial polled, and then wishes to request service again. The host MPU must clear the "rsv" message and then set it true again. Now suppose this temporary false condition happens within one occurrence of SPAS. If the service request function has been implemented exactly as per IEEE-488, it will not be recognized, and SRQ will not be asserted a second time. Therefore, "rsv" may only be cleared when the device is known not to be in SPAS, which can only happen if it is cleared as a consequence of some pre-arranged action of the controller. This action would normally be a part of the service routine executed by the controller as a response to the request for service. For example, if service was requested by an instrument which had some data to send for processing or to a printing device, then "rsv" could be cleared when it is addressed to talk and send its data over the bus.

For many applications, the fact that the device has been serial polled after requesting service is considered sufficient response from the controller. The "rsv" local message, therefore, simply becomes a request for the controller to read its serial poll status byte. It is then desirable to be able to clear and reassert "rsv" at any time after the serial poll status byte has been polled and the SPAS interrupt set. The WD9914 is able to record a false transition of "rsv1" or "rsv2" by moving from APRS1 to APRS2 even if the device is in SPAS. This makes the above approach to serial polling possible.

To further support this approach, the "rsv2" auxiliary command is automatically cleared when the serial poll status byte is polled, ensuring that "rsv2" is cleared before a second serial poll can occur. If this were not the case, then the same status byte might be polled twice by the controller with the RQS bit true, which may indicate that two reasons for requiring service have arisen.

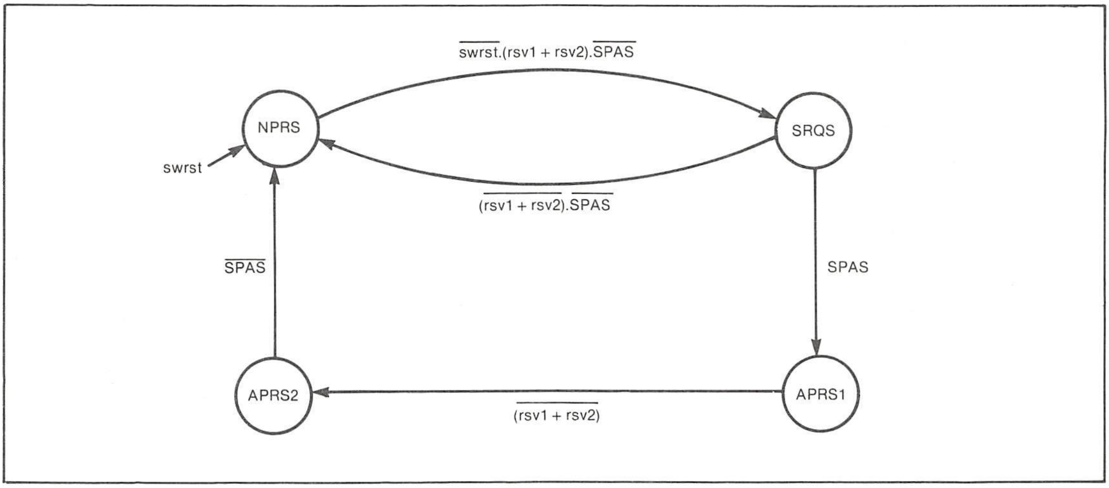

$\overline{swrst}.(rsv1 + rsv2).\overline{SPAS}$
NPRS
SRQS
swrst
$\overline{(rsv1 + rsv2)}.\overline{SPAS}$
$\overline{SPAS}$
SPAS
APRS2
APRS1
$\overline{(rsv1 + rsv2)}$

Figure 9. SERVICE REQUEST STATE DIAGRAM

The WD9914 will only send one serial poll status byte during each active period of SPAS. However, it will send this status byte as many times as the controller is prepared to accept it. Therefore, the controller should only read the status byte once per serial poll; otherwise, each time a status byte is sent with the RQS message true, the SPAS interrupt will be generated and "rsv2" will be cleared.

Table 12. SERVICE REQUEST MNEMONICS

MESSAGES			STATES		
swrst	=	software reset	NPRS	=	negative poll response state
srv1	=	request service 1 (bit 7 of serial poll register)	SRQS	=	service request state
rsv2	=	request service 2 (auxiliary command register)	APRS1	=	affirmative poll state 1
			APRS2	=	affirmative poll state 2
			SPAS	=	serial poll active state (talker function)

Table 13. SERVICE REQUEST MESSAGE OUTPUTS

STATE	REMOTE MESSAGES SENT SRQ	OTHER ACTIONS
NPRS	(F)	
SRQS	T	
APRS1	(F)	— "rsv2" cleared if in SPAS and STRS — SPAS interrupt set if in SPAS when STRS is exited
APRS2	(F)	— same as APRS1

REMOTE/LOCAL FUNCTION

The WD9914 remote local state diagram is shown in Figure 10. If differs little from that of IEEE-488.

The complete listener function (LAF) is used to effect the transition from LOCS to REMS or from LWLS to RWLS. This means that if the APT interrupt is masked, the device will enter one of the remote states in response to its listen address, but if secondary addressing is enabled, then this will not happen until 'dacr' is written with c/s true in response to a valid secondary address. In addition, the transition to one of the remote states will occur if 'lon' is used to address the device to listen.

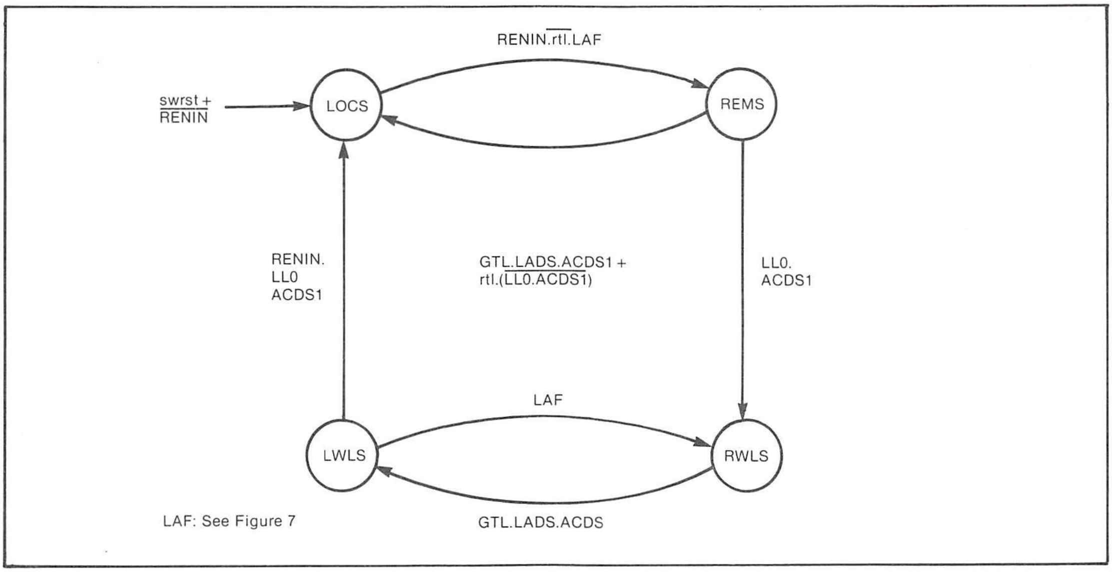


Figure 10. WD9914 REMOTE/LOCAL STATE DIAGRAM

Table 14. REMOTE/LOCAL MNEMONICS

MESSAGES			STATES		
swrst	=	software reset	LOCS	=	local state
rtl	=	return to local	REMS	=	remote state
RENIN	=	internal remote enable message (debounced)			
GTL	=	go to local	RWLS	=	remote with lockout state
LLO	=	local lockout	LWLS	=	local with lockout state
			LADS	=	listener addressed state (listener function)
			ACDS1	=	accept data state 1 (acceptor handshake)

PARALLEL POLL FUNCTION

The parallel poll function of the WD9914 only nominally supports logically-configured parallel poll. With a suitable software package, remotely-configured parallel poll may also be easily implemented. The state diagram is shown in Figure 11.

When the EOI and ATN lines become true simultaneously (the Identify message), the contents of the Parallel Poll register are output to DIO(8-1). If parallel poll is to be used in a particular bus environment, then the Pull-Up Enable (PE) input of the SN75160 must be held low so that the DIO(8-1) are driven by open collector buffers. Parallel Poll, occurring when the Parallel Poll register is in the hardware reset condition of all zeros, will result in none or DIO(8-1) being pulled low. This corresponds to the parallel poll idle state (PPIS). If it is desired to participate in a parallel poll, then the bit corresponding to the desired parallel poll response is set true. This implements the parallel poll standby state (PPSS), and, when the Identify message becomes true, the appropriate line of DIO(8-1) is pulled low. This is equivalent to the parallel poll active state (PPAS). Only one bit of the Parallel Poll register should be set true at once.

Remotely Configured Parallel Poll

The parallel poll configure command (PPC) is treated by the WD9914 as an unrecognized addressed command. It is passed through when the WD9914 is in LADS. If an instrument is to be remotely configured for parallel poll, then the pass through next secondary (pts) auxiliary command should be written before releasing the DAC holdoff. This will cause the next command received to also set a UNC interrupt if it is a secondary command. The secondary command will be either the parallel poll enable command (PPE) or the parallel poll disable command (PPD) and should be read from the Command Pass Through register and identified. If it is the PPE command, then the attendant bits (S, P1, P2, P3) should be extracted and stored by the host MPU. The S bit should then be matched against the individual status of the instrument (represented by 'ist'); and if they are the same, the bit corresponding to the parallel poll response, specified by P1, P2, P3, should be set true in the Parallel Poll register. If this is not the case, then the Parallel Poll register should be cleared if it is not already clear. After this, each time the individual status of the device changes, the 'ist' should again be matched against the S bit and the Parallel Poll register updated accordingly until PPD or PPU is received.

If a PPD command is passed through after the "pts" feature has been written, the Parallel Poll register should be cleared before the DAC holdoff is released. The PPC command that precedes PPD is an address command; it is a means of eliminating individual members of a parallel poll. The parallel unconfigure command is treated by the WD9914 as an unrecognized universal command. When it is passed through, the host MPU should clear its Parallel Poll register before releasing the DAC holdoff. This command will clear all members of a parallel poll.

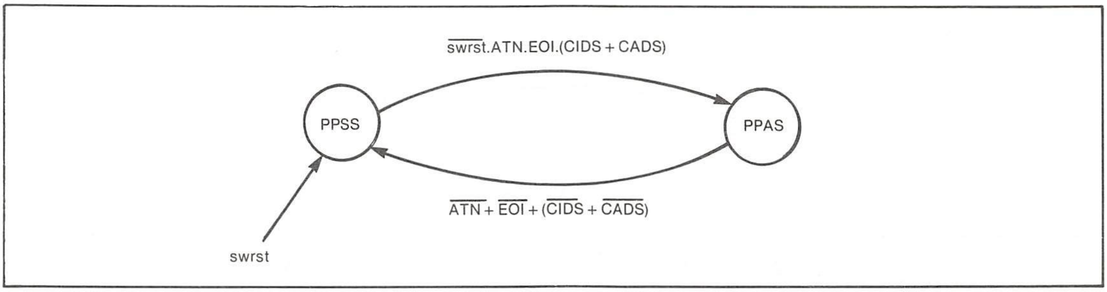

$\overline{\text{swrst}}$.ATN.EOI.(CIDS + CADS)
PPSS
PPAS
$\overline{\text{ATN}}$ + $\overline{\text{EOI}}$ + ($\overline{\text{CIDS}}$ + $\overline{\text{CADS}}$)
swrst

Figure 11. WD9914 PARALLEL POLL STATE DIAGRAM

Table 15. PARALLEL POLL MNEMONICS

MESSAGES			STATES		
swrst	=	software reset	PPSS	=	parallel poll standby state
ATN	=	attention	PPAS	=	parallel poll active state
EOI	=	end or identify	CIDS	=	controller idle state (controller function)
			CADS	=	controller addressed state (controller function)

Table 16. PARALLEL POLL MESSAGE OUTPUTS

	REMOTE MESSAGES SENT	
STATE	DIO(8-1)	OTHER ACTIONS
PPSS	(NUL)	
PPSS	PARALLEL POLL REG*	

*If there is a true bit in the Parallel Poll register, it must be sent active; any false bit must be sent passive.

CONTROLLER FUNCTION

The controller function of the WD9914 is greatly simplified compared with that of IEEE-488. It relies heavily on software support but, with suitable software, it enables all subsets of the controller function to be implemented. With this approach, the controller logic is reduced to a small proportion of the chip area, which means that the device may be economically used in situations where a talker/listener only is required.

Figure 12 shows the controller function state diagram. With suitable software, it will perform the full controller function, as described in the IEEE-488A 1980 supplement to the IEEE-488 1978. It therefore includes the additional state CSHS, which allows time for DAV to be recognized false by all devices on the bus before ATN is asserted. The "tcs" local message is implemented by an immediate execute auxiliary command. The state CWAS therefore is added to record the occurrence of this command until the acceptor handshake enters ANRS and the device can enter CSHS. The "tca" auxiliary command also causes entry into CSHS although IEEE-488A 1980 allows it to move directly from CSBS to CSWS. This is done for convenience of implementation and results in the "tca" auxiliary command taking an extra 1.6 microseconds to assert ATN.

The delay between CSWS and CAWS is slightly less than specified in IEEE-488A 1980 but the total time taken in moving from CSWS to CACS is still greater than the specified minimum.

The Controller Parallel Poll State (CPPS) is not included on the WD9914. To conduct a parallel poll, a WD9914 based controller must set the "rpp" clear/set auxiliary command true when it is in CACS, moving it to CPWS which sends EOI true. The host MPU must then wait two microseconds before reading back the parallel poll responses via the Command Pass Through register. The "rpp" auxiliary command can then be cleared, EOI will go false, and the parallel poll is complete. The host MPU will receive BO interrupt as soon as the WD9914 reenters CACS and the source handshake becomes active.

Controller Self Addressing

The acceptor handshake does not operate when the controller is active. This means commands being sent are not monitored, and special precautions are required as a consequence of this when addressing devices and when passing control.

When the controller is active, it uses "ton" or "lon" to address and unaddress itself. IEEE-488 provides for the controller to locally address itself to listen, but there is no corresponding local message for the talker. The WD9914 should always accompany a "ton" auxiliary command with cs true with its own talk address or an UNT command sent over the bus. Similiarly, if the WD9914 sends the talk address of another device over the bus, it should ensure that it is in TIDS by writing the "ton" auxiliary command false.

Passing Control

As Figure 12 shows, the controller transfer state (CTRS) of IEEE-488 is not present, and all transitions associated with the TCT command have been removed. Instead, two immediate execute auxiliary commands are included. Request control (rqc) will cause a transition from CIDS to CADS, and the release control command (rlc) will return the function to CIDS. The TCT command is treated similarly to an unrecognized addressed command, but will cause a UNC interrupt if the device is in TADS.

Figure 13 is a representation of the sequence of events involved in passing control from one WD9914 based device to another. The device passing control must initially ensure that it is not in TADS; then it should send out the talk address of the device to receive control. The receiving device will enter TADS, and after any DAC holdoff has been released, the host MPU of the device passing control will set a BO interrupt indicating that it may then send the TCT command. The TCT command will cause a UNC interrupt to the host MPU of the receiving device, and also a DAC holdoff will occur. The host MPU of the receiving device must examine its Command Pass Through register, and upon identifying TCT, should write the auxiliary command "rqc" to put its WD9914 into CADS. The receiving device may then release DAC with a "dacr" auxiliary command causing another BO interrupt at the device passing control. This indicates that the "ric" auxiliary command may then be used by the host MPU of the device passing control to return its WD9914 to CIDS and allowing ATN to go false. The receiving device then enters CACS, asserts ATN, and its host MPU gets a BO interurpt as the source handshake becomes active. The passing of control is complete.

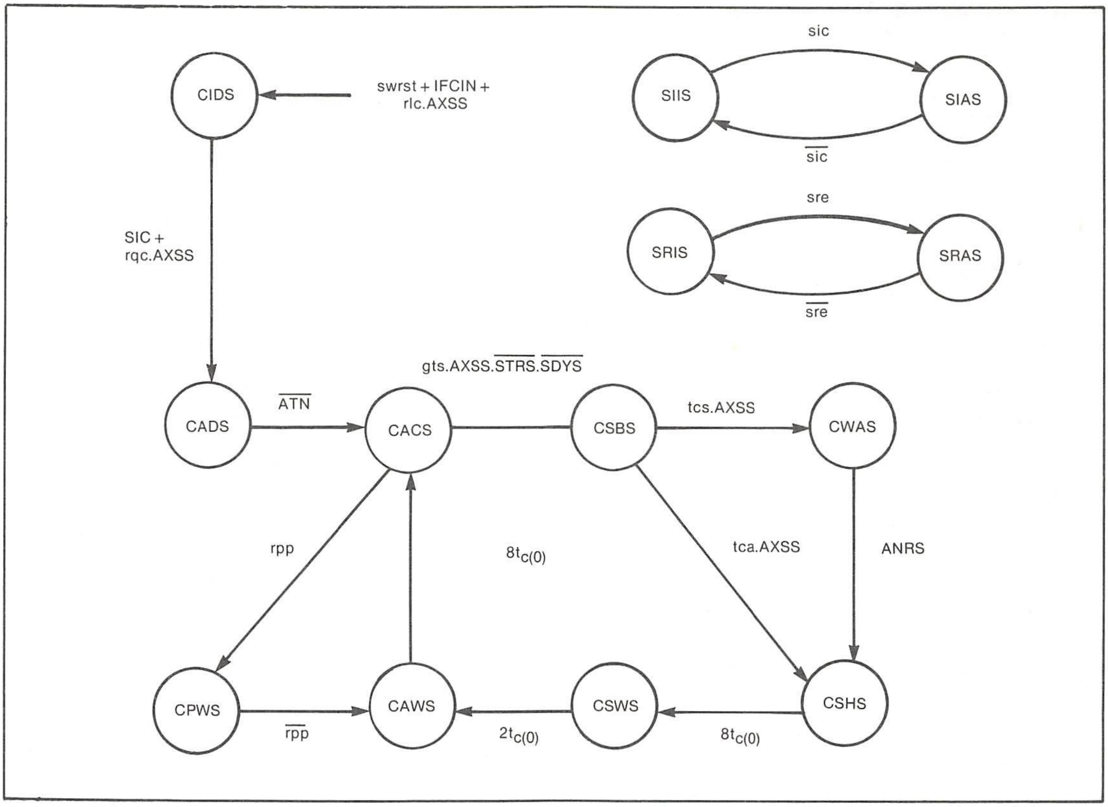


Figure 12. WD9914 CONTROLLER STATE DIAGRAMS

Table 17. CONTROLLER FUNCTION MNEMONICS

MESSAGES			STATES		
swrst	=	software reset	CIDS	=	controller idle state
sic	=	send interface clear	CADS	=	controller addressed state
sre	=	send remote enable	CACS	=	controller active state
rqc	=	request control	CSBS	=	controller standby state
rlc	=	release control	CSHS	=	controller standby hold state
gts	=	go to standby	CSWS	=	controller synchronous wait state
tcs	=	take control synchronously	CAWS	=	controller active wait state
tca	=	take control asynchronously	CPWS	=	controller parallel poll wait state
rpp	=	request parallel poll	ANRS	=	acceptor not ready state (acceptor handshake)
IFCIN	=	internal interface clear message (a debounced signal which is suppressed if "sic" is true)	SDYS	=	source delay state (source handshake)
ATN	=	attention	STRS	=	source transfer state (source handshake)
$t_{c(0)}$	=	clock cycle time	AXSS	=	auxiliary command strobe state (auxiliary command register)
			LWAS	=	controller wait for ANRS state

Table 18. CONTROLLER FUNCTION MESSAGE OUTPUTS

	REMOTE MESSAGE SENT			
STATE	ATN	EOI	D10(8-1)	OTHER ACTIONS
CIDS	(F)	(F)	(NUL)	
CADS	(F)	(F)	(NUL)	
CACS	T	F	DATA OUT REG	Data Out reg. may contain any of the commands in Table 19
CSBS	F	(F)	(NUL)	
CWAS	F	(F)	(NUL)	
CSHS	F	(F)	(NUL)	
CSWS	T	F	(NUL)	
CAWS	T	F	(NUL)	
CPWS	T	T	(NUL)	DIO(8-1) may be read via the Command Pass Through register

	REMOTE MESSAGES SENT	
STATE	IFC	OTHER ACTIONS
SIIS*	(F)	Internal interface. Clear message IFCIN is held false
SIIS	F	
SIAS	T	

	REMOTE MESSAGES SENT	
STATE	REN	OTHER ACTIONS
SRIS*	(F)	
SRIS	F	
SRAS	T	

*Buffers not configured for a system controller; otherwise, buffers are configured for system controller.

The REN and IFC outputs of the WD9914 are controlled by the auxiliary commands "sre" and "sic." These should never be used by the host MPU of a device unless it is the system controller. As may be seen from Figure 14, the REN and IFC outputs of the WD9914 are open drains with internal pull-ups. This means that the outputs are capable of driving the inputs of the buffers if the device is a system controller. If not, the buffers will drive into the REN and IFC pins and override the pull-ups. Hence, no direction control is required.

The false transition of REN and the true transition of IFC are both debounced to prevent noise on these lines from causing permanent state changes on the WD9914. In addition, the internal interface clear signal (IFCIN) is held false if the WD9914 is sending IFC. Figure 12 shows the reason for this. If the device is not a system controller, then the occurence of IFC will return the controller function to CIDS. If, however, the device is a system controller when it asserts IFC and is in CIDS, the "sic" auxiliary command will cause it to enter CADS. As IFCIN is suppressed, it will not be forced back into CIDS, and there will be no conflict.

System Controller

The WD9914 has no on-chip means of determining whether or not it is the system controller. Instead, this is determined by the software and by the configuration of the buffers to the IEEE-488 bus.

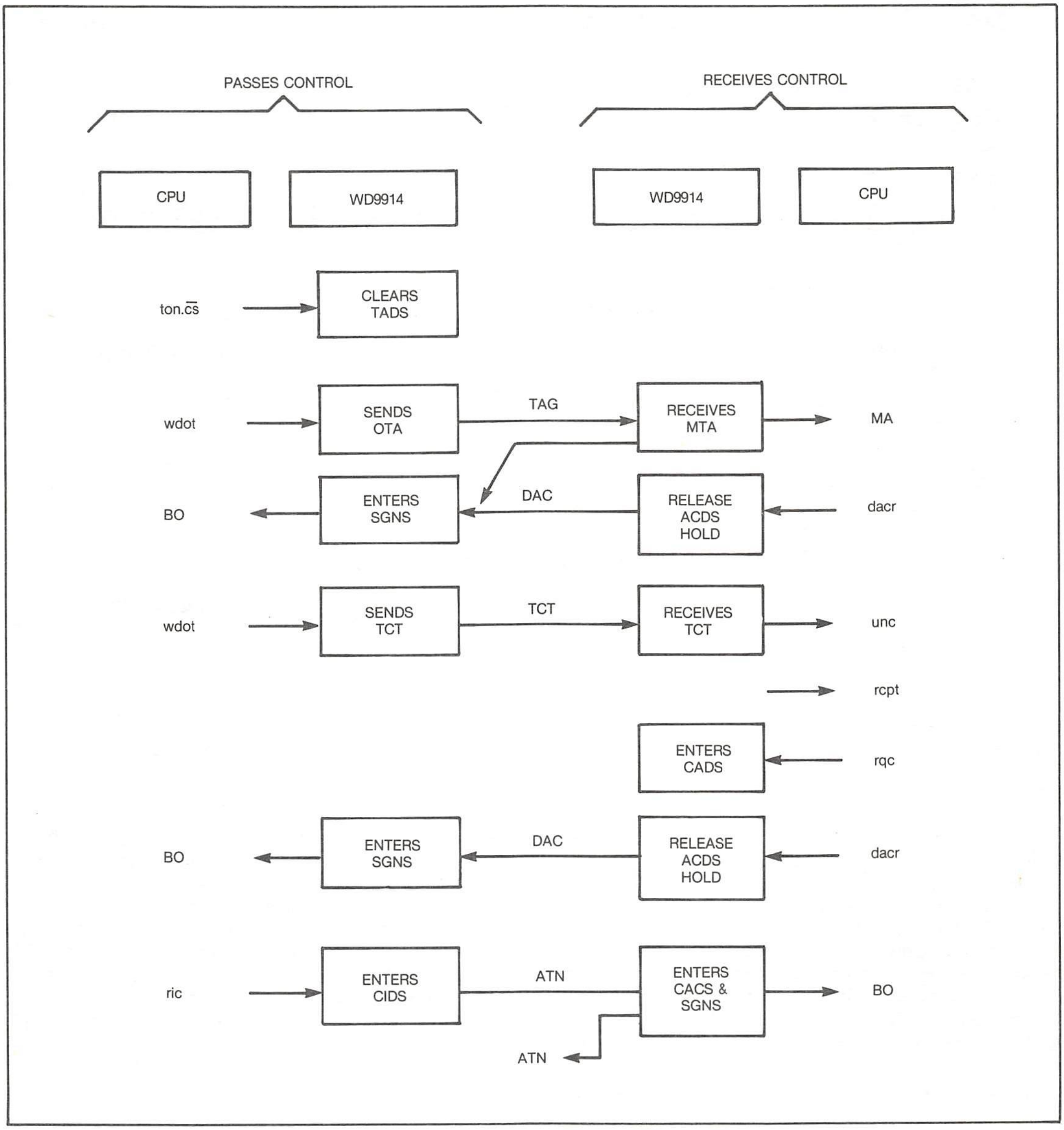
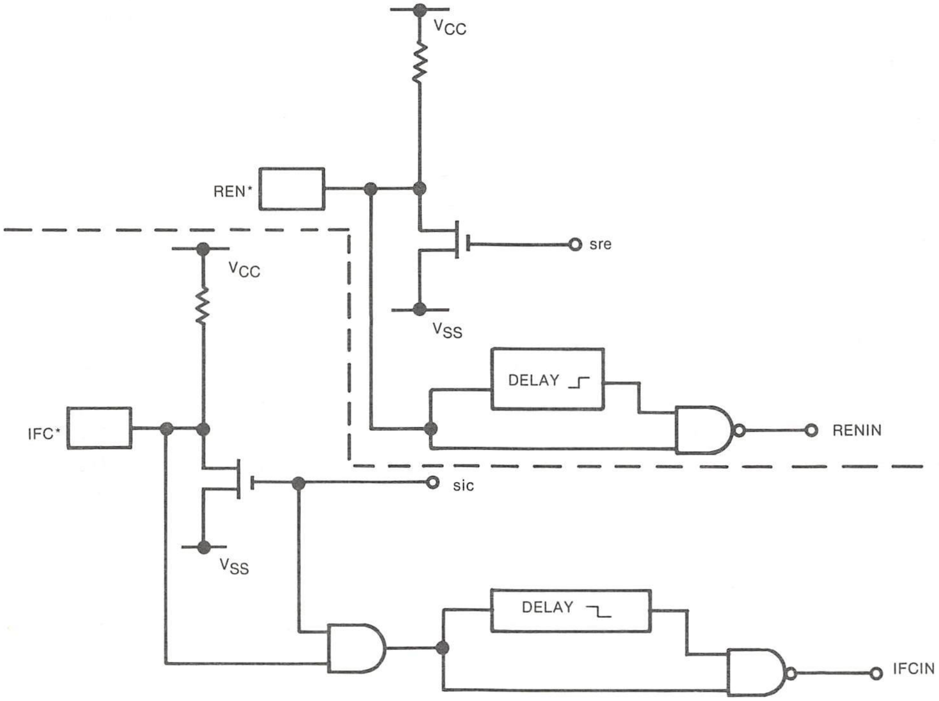


Figure 13. PASSING CONTROL BETWEEN WD9914



*The REN and IFC signals are at the pins of the WD9914 and are therefore negative logic signals. The remaining signals are conventional positive logic signals.

Figure 14. IFC AND REN PINS

Table 19. MULTILINE INTERFACE MESSAGES

COMMAND	SYMBOL	DIO 8 - 1	CLASS	INTERRUPT (1, 2)	DAC (3) HOLDOFF	NOTE
Addressed Command Group	ACG	000XXXX	AC	—	—	
Device Clear	DCL	X0010100	UC	DCAS	Yes	
Group Execute Trigger	GET	X0001000	AC	GET	Yes	
Go To Local	GTL	X0000001	AC	RLC	No	14
Listen Address Group	LAG	X01XXXXX	AD	—	—	
Local Lockout	LLO	X0010001	UC	None	No	
My Listen Address	MLA	X01AAAAA	AD	MA,MAC,RLC	MA Only	4,14
My Talk Address	MTA	X10AAAAA	AD	MA,MAC	MA Only	4
My Secondary Address	MSA	X11SSSSS	SE	APT	Yes	5,6
Other Secondary Address	OSA	SCG.MSA-	SE	APT	Yes	6,7
Other Talk Address	OTA	TAG.MTA-	AD	MAC	No	
Primary Command Group	PCG	ACG + UCG + LAG + TAG	—	—		
Parallel Poll Configure	PPC	X0000101	AC	UNC	Yes	8
Parallel Poll Enable	PPE	X110SPPP	SE	UNC	Yes	9,10
Parallel Poll Disable	PPD	X111DDDD	SE	UNC	Yes	9,11
Parallel Poll Unconfigure	PPU	X0010101	UC	UNC	Yes	12
Secondary Command Group	SCG	X11XXXXX	SE	—	—	
Selected Device Clear	SDC	X0000100	AC	DCAS	Yes	
Serial Poll Disable	SPD	X0011001	UC	None	No	
Serial Poll Enable	SPE	X0011000	UC	None	No	
Take Control	TCT	X0001001	AC	UNC	Yes	13
Talk Address Group	TAG	X10XXXXX	AD	—	—	
Unlisten	UNL	X0111111	AD	MAC	No	
Untalk	UNT	X1011111	AD	—	—	
Universal Command Group	UCG	X001XXXX	UC	None	No	

Classes: UC — universal command
AC — addressed command
AD — address
SE — secondary command

Symbols: 0 — logical zero (high level on GPIB)
1 — logical one (low level on GPIB)
x — don't care (received message)

NOTES:

1. Interrupts listed are as a direct consequence of the command received. They are set during ACDS1 and will cause the INT pin to be pulled low if unmasked.
2. The address commands will only cause their corresponding interrupt if the device is in LADS, with the exception of TCT.

corresponding interrupt is unmasked.

4. AAAAA represents the primary address of a device.
5. SSSSS represents the secondary address of a device.
6. Secondary addresses are handled via address pass through (APT interrupt). The host MPU should respond by writing the "dacr" auxiliary command with cs false.
7. If OSA is passed through via the APT interrupt,

the host MPU should respond by writing the "dacr" auxiliary command with cs false.

8. PPC is not recognized by the WD9914 and is therefore treated as an unrecognized addressed command.
9. PPE and PPD are secondary commands. These may be passed through to the host MPU using the "pts" auxiliary command. When the PPC command is received, the "pts" auxiliary command should be written. PPE or PPD will then cause an APT interrupt.
10. SPPP specifies the sense bit, and the desired parallel poll response is a remotely configured parallel poll.
11. DDDD specifies don't care bits which must be sent as zeros, but need not be decoded by the host MP of the receiving devices.
12. PPU is not recognized by the WD9914 and will cause a UNC interrupt.
13. TCT is not recognized directly by the WD9914. It will cause a UNC interrupt when the device is in TADS.
14. RLC is set if MLA or GTL causes an appropriate transition in the Remote/Local function.

TYPICAL SEQUENCES OF EVENTS FOR THE CONTROLLER

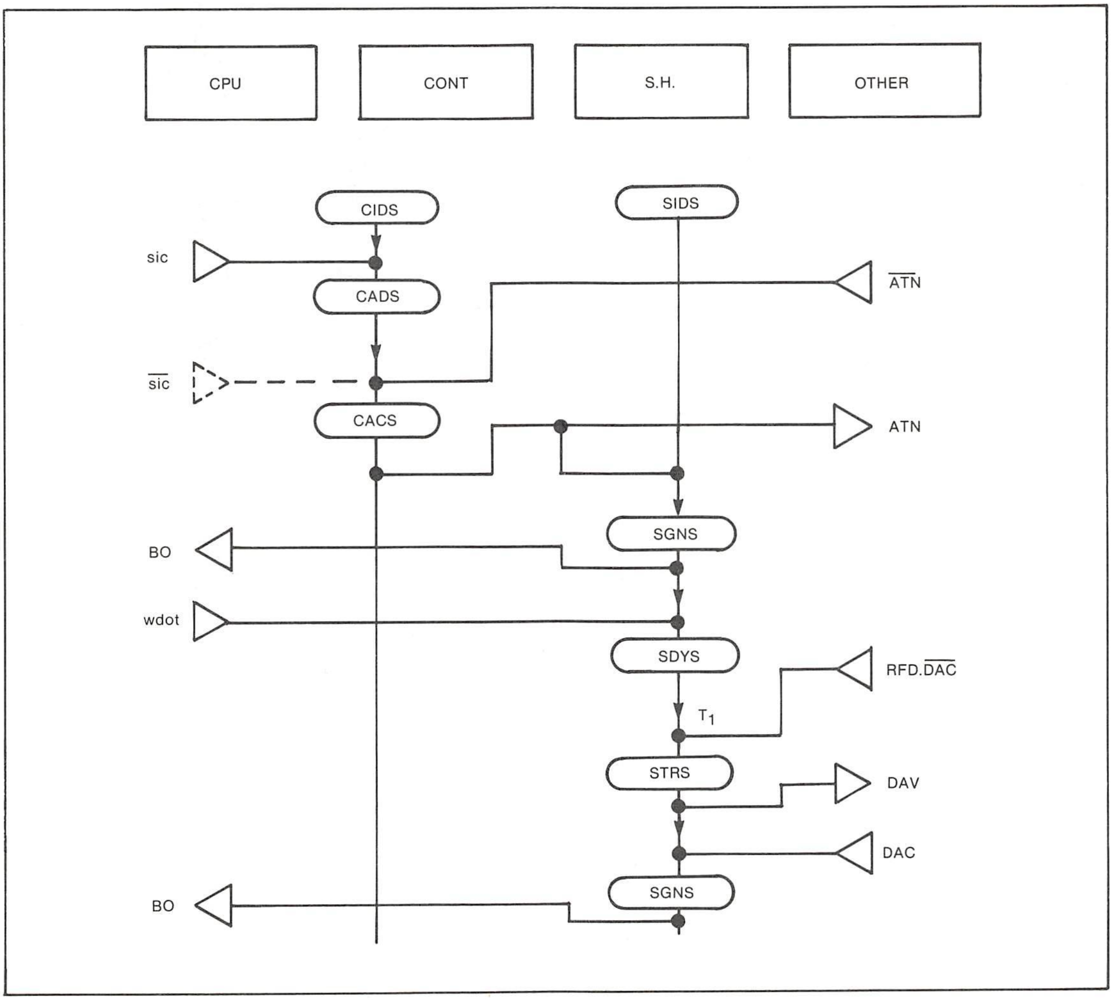
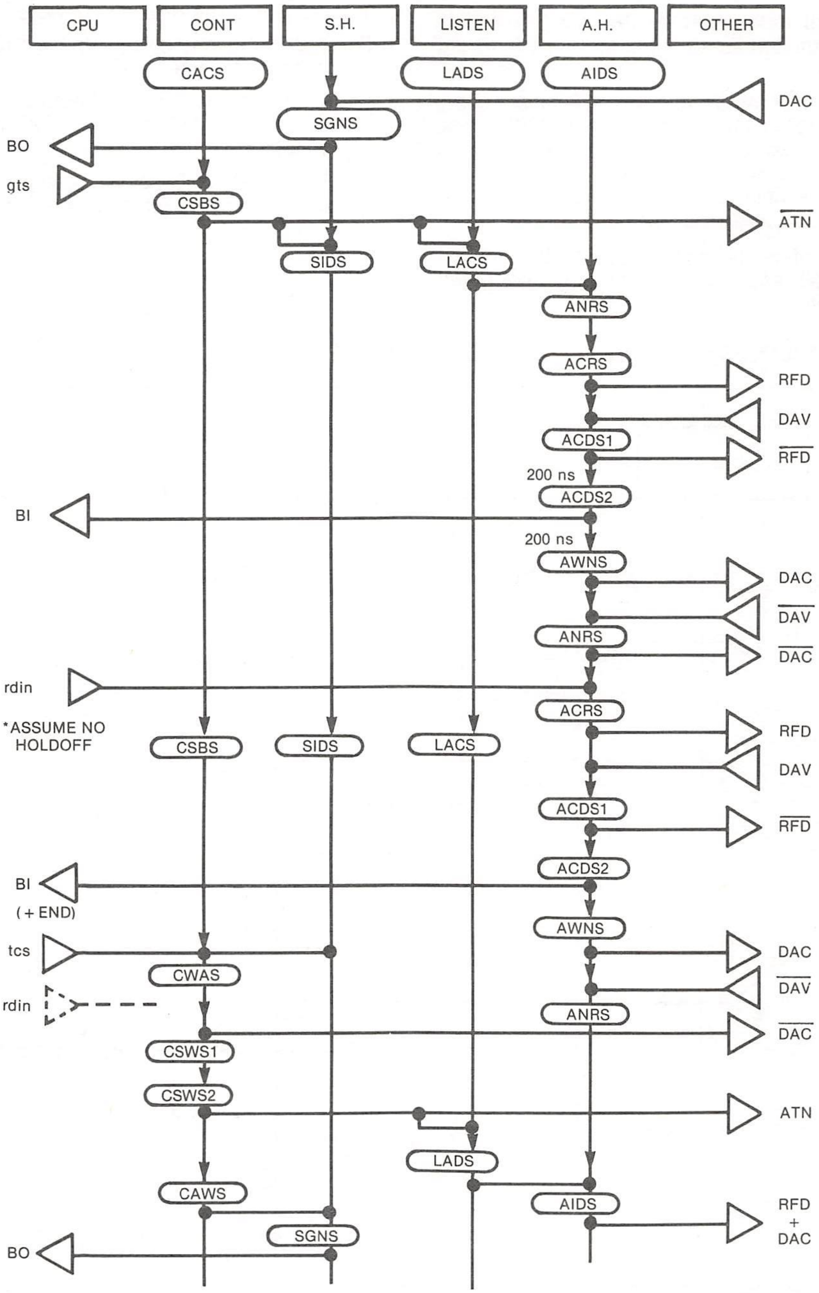
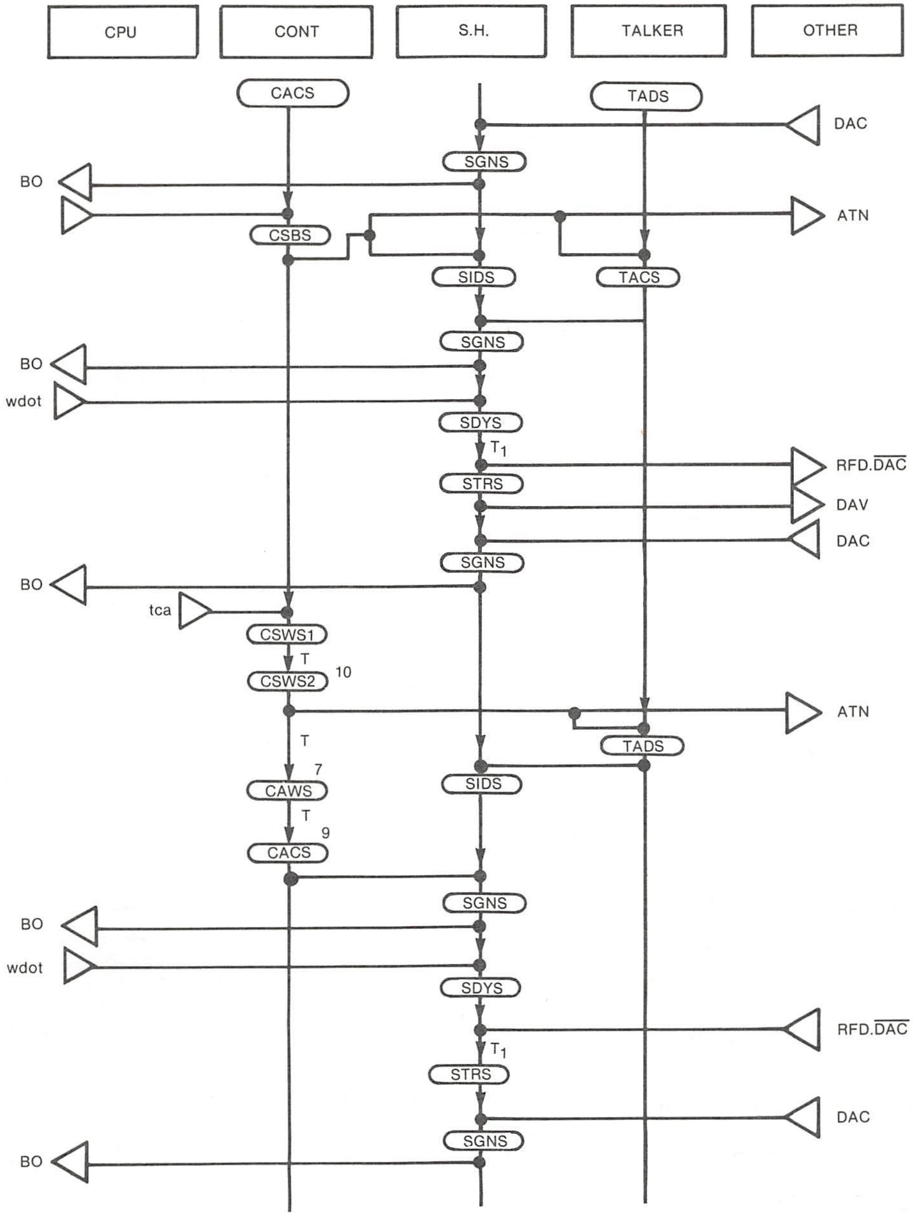


Figure 15. CONTROLLER TAKING CONTROL



*CWAS INHIBITS ANRS → ACRS SO RDIN CAN OCCUR BEFORE ATN IS SET.

Figure 16. CONTROLLER AS A LISTENER (GOING TO STANDBY)



*MOMENTARY TRANSITION FOLLOWING BO INTERRUPT IS GUARANTEED ON THE WD9914.

Figure 17. CONTROLLER AS A TALKER (GOING TO A STANDBY)

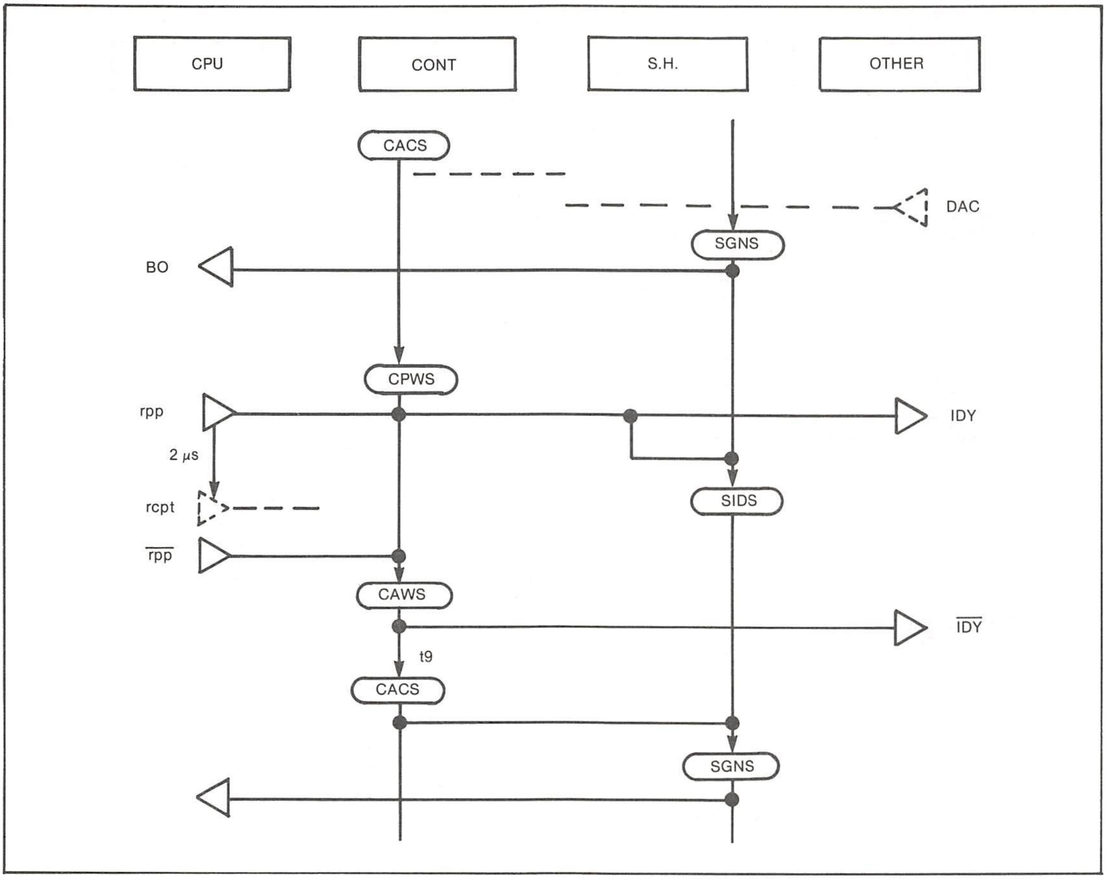


Figure 18. CONTROLLER PARALLEL POLLING

WD9914 ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE-AIR TEMPERATURE RANGE (Unless otherwise noted)*

Supply Voltage, V_{CC} (see Note 1) .... –0.3V to 20V
All Input and Output Voltages .... –0.3V to 20V
Continuous Power Dissipation .... 0.8 W
Operating Free-Air Temperature Range ... 0°C (32°F) to 70°C (158°F)
Storage Temperature Range...... –55°C (–67°F) to 150°C (302°F)

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at those or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1:
Under absolute maximum ratings, voltage values are with respect to V_{SS}.

RECOMMENDED OPERATING CONDITIONS

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.75	5	5.25	V
Supply voltage, V_{SS}		0		V
High-level input voltage, V_{IH}	2		$V_{CC}+1$	V
Low-level input voltage, V_{IL}	V_{SS}-0.3		0.8	V
Operating free-air temperature, T_A	0		158 70	°F °C

ELECTRICAL CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage Except REN,IFC,INT	$I_{OH} = -400\,\mu A$	2.4		V_{CC}	V
	$\overline{REN}$,$\overline{IFC}$ only	$I_{OH} = -100\,\mu A$	2.2		V_{CC}	V
V_{OL}	Low-level output voltage	$I_{OL} = 2\,mA$	V_{SS}		0.4	V
I_I	Input current (any input)	$V_I = 2\,V$ to V_{CC}			± 10	μA
I_{CC}	V_{CC} supply current				150	mA
C_i	Input capacitance (any input)	f = 1 MHz, unmeasured pins at 0 V			15	pF

† All typical values are at $T_A = 25°C$ (77°F) and nominal voltage.

TIMING CHARACTERISTICS AND REQUIREMENTS

Timing characteristics and requirements are given in the following and relevant timing diagrams are shown in Figure 10 through Figure 27.

Clock and Host Interface Timing Requirements Over Full Range of Operating Conditions

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
$t_{c(0)}$	Clock cycle time	200		2000	nsec
$t_{w(0H)}$	Clock high pulse width	100		1955	nsec
$t_{w(0L)}$	Clock low pulse width	45			nsec
$t_{su(AD)}$	Address setup time	0			nsec
$t_{su(DBIN)}$	DBIN setup time	0			nsec
$t_{su(CE)}$	$\overline{CE}$ setup time	150			nsec
$t_{su(WE)}$	$\overline{WE}$ setup time	0			nsec
$t_{w(WE)}$	$\overline{WE}$ low pulse width	80			nsec
$t_{su(DA)}$	Data setup time	60			nsec
$t_{h(DA)}$	Data hold time	0			nsec
$t_{h(AD)}$	Address hold time	0			nsec
$t_{h(DBIN)}$	DBIN hold time	0			nsec
$t_{h(CE)}$	$\overline{CE}$ hold time	80			nsec
$t_{su(GR)}$	$\overline{ACCGR}$ setup time	100			nsec
$t_{h(GR)}$	$\overline{ACCGR}$ hold time	80			nsec

Host Interface Timing Characteristics Over Full Range of Operating Conditions

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
$t_{a(CE)}$	Access time from $\overline{CE}$			150	nsec
$t_{a(DBIN)}$	Access time from DBIN			150	nsec
$t_{su(AD)}$	Address setup time to $\overline{CE}$	0			nsec
$t_{z(DBIN)}$	Hi-Z time from DBIN		50	100	nsec
$t_{z(CE)}$	Hi-Z time from $\overline{CE}$		50	100	nsec
$t_{a(GR)}$	Access time from $\overline{ACCGR}$			150	nsec
$t_{z(GR)}$	Hi-Z time from $\overline{ACCGR}$		50	100	nsec
$t_{d(GR/RQ)}$	Delay of $\overline{ACCRQ}$ high from $\overline{ACCGR}$			100	nsec

Source Handshake Timing Characteristics Over Full Range of Operating Conditions (see Note 1)

SYMBOL	PARAMETER	TEST CONDITIONS	MIN	MAX	UNIT
t_{d1}	Delay of DAV true from end of write operation to data out register	Normal T_1 (see Note 2)	12(0)↑	12(0)↑ + 310	nsec
		Short T_1 (see Note 2)	8(0)↑	8(0)↑ + 310	nsec
		Very short T_1 (see Note 2)	4(0)↑	4(0)↑ + 310	nsec
t_{d2}	Delay of valid GPIB data lines from end of write cycle			140	nsec
t_{d3}	Delay of BO interrupt from DAC true	BO interrupt unmasked		300	nsec
t_{d4}	Delay of $\overline{\text{ACCRQ}}$ DAC true			300	nsec
t_{d5}	Delay of DAV false from DAC true			160	nsec

NOTES:

1. The timing of the source handshake is the same whether ATN is true or false; i.e., whether the device is in TACS, CACS, or SPAS.
2. A very short bus settling time (T_1) occurs on the second and subsequent data byte sent when ATN is false if the "vstd1" feature is set. A slightly longer bus settling time takes place if "std1" is set unless there is a very short bus settling time. In all other instances, a normal bus settling time occurs.

Acceptor Handshake Timing Characteristics Over Full Range of Operating Conditions

SYMBOL	PARAMETER	TEST CONDITIONS	MIN	MAX	UNIT
t_{d6}	Delay of BI interrupt from DAV true	BI interrupt unmarked ATN = false device is in LACS	2(0)↑	2(0)↑ + 415	nsec
t_{d7}	Delay of $\overline{\text{ACCRQ}}$ from DAV true	ATN = false device is in LACS	2(0)↑	2(0)↑ + 290	nsec
t_{d8}	Delay of DNAC false from DAV true	ATN = false device is in LACS	3(0)↑	3(0)↑ + 445	nsec
t_{d9}	Delay of NRFD false from end of read operation of Data In register	ATN = false device is in LACS		220	nsec
t_{d10}	Delay of interface message interrupt from DAV true	ATN = true device not in CACS all interface message interrupts (except UNO)	2(0)↑	2(0)↑ + 415	nsec
		UNO interrupt only	5(0)↑	5(0)↑ + 415	nsec
t_{d11}	Delay of NDAC false from DAV true	ATN = true device not in CACS no DAC holdoff	7(0)↑	7(0)↑ + 415	nsec
t_{d12}	Delay of NDAC false from end of write operation			230	nsec
t_{d13}	Delay of NRFD false from DAV false	ATN = true device not in CACS		180	nsec

ATN, EOI, and IFC Timing Characteristics Over Full Range of Operating Conditions

SYMBOL	PARAMETER	TEST CONDITIONS	MIN	MAX	UNIT
t_{d14}	Delay of NDAC true from ATN true	Device is not in CACS		195	nsec
t_{d15}	Delay of TE high from EOI true	Device is not in CACS		125	nsec
t_{d16}	Delay of valid data from EOI true	Device is not in CACS		140	nsec
t_{d17}	Delay of TE low from EOI false	Device is not in CACS		125	nsec
t_{d18}	Delay of NRFD true from ATN false	Device is in LADS/LACS		140	nsec
t_{d19}	Response time to IFC		$16t_{c(0)}$	$30t_{c(0)}$	nsec

Controller Timing Characteristics Over Full Range of Operating Conditions

SYMBOL	PARAMETER	TEST CONDITIONS	MIN	MAX	UNIT
t_{d20}	Delay of ATN true from end of t_{ca} aux command		$8t_{c(0)}$	10(0)↑ + 220	nsec
t_{d21}	Delay of BO interrupt from end of t_{ca} aux command		$18t_{c(0)}$	22(0)↑ + 415	nsec
t_{d22}	Delay of ATN true from end of t_{cs} aux command	BO unmasked device is in ANRS	$8t_{c(0)}$	10(0)↑ + 220	nsec
t_{d23}	Delay of BO interrupt from end of t_{cs} aux command	BO unmasked device is in ANRS	$18t_{c(0)}$	22(0)↑ + 415	nsec
t_{d24}	Delay of EOI true from r_{pp} aux command set			230	nsec
t_{d25}	Delay of EOI false from r_{pp} aux command cleared			230	nsec
t_{d26}	Delay of EOI from r_{pp} aux command cleared	BO unmasked	$8t_{c(0)}$	10(0)↑ + 415	nsec
t_{d27}	Delay of ATN false from sts aux command	Device is not in SDYS or STRS		210	nsec

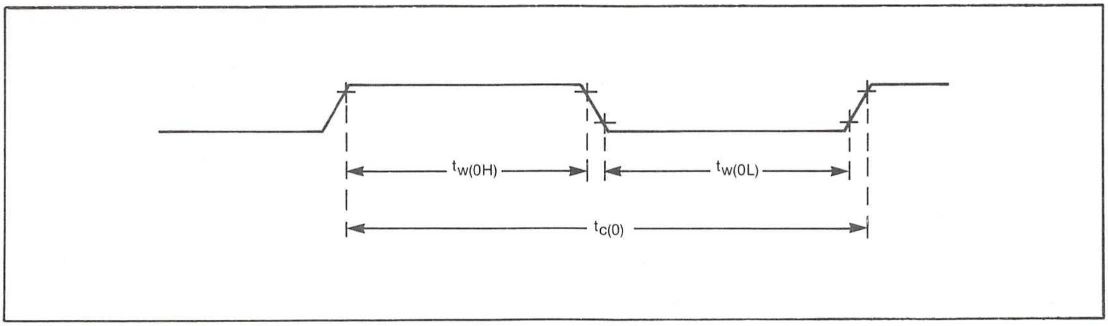


Figure 19. WD9914 CLOCK CYCLE TIMING

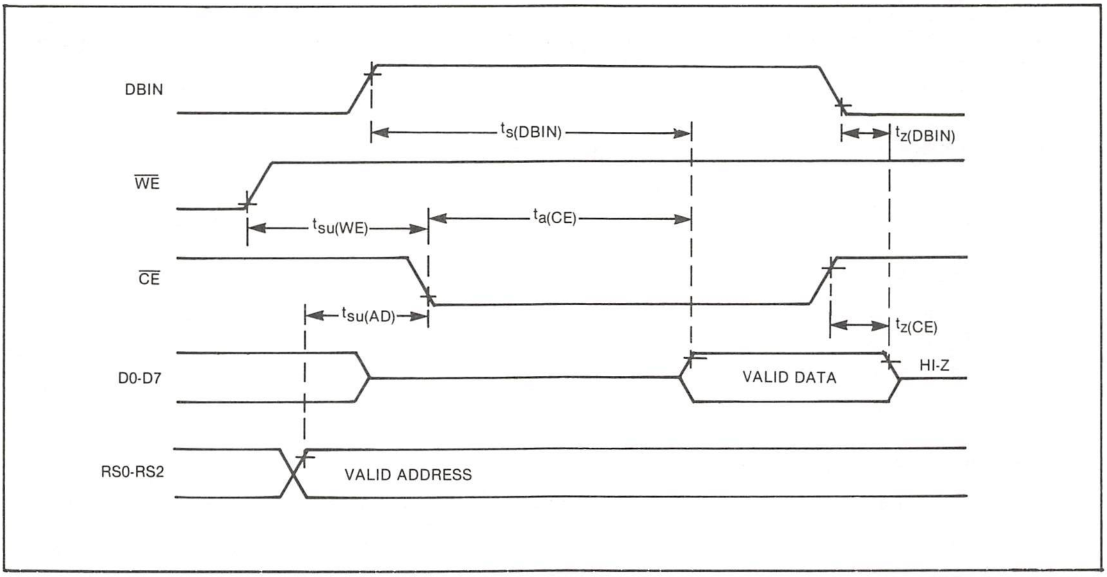
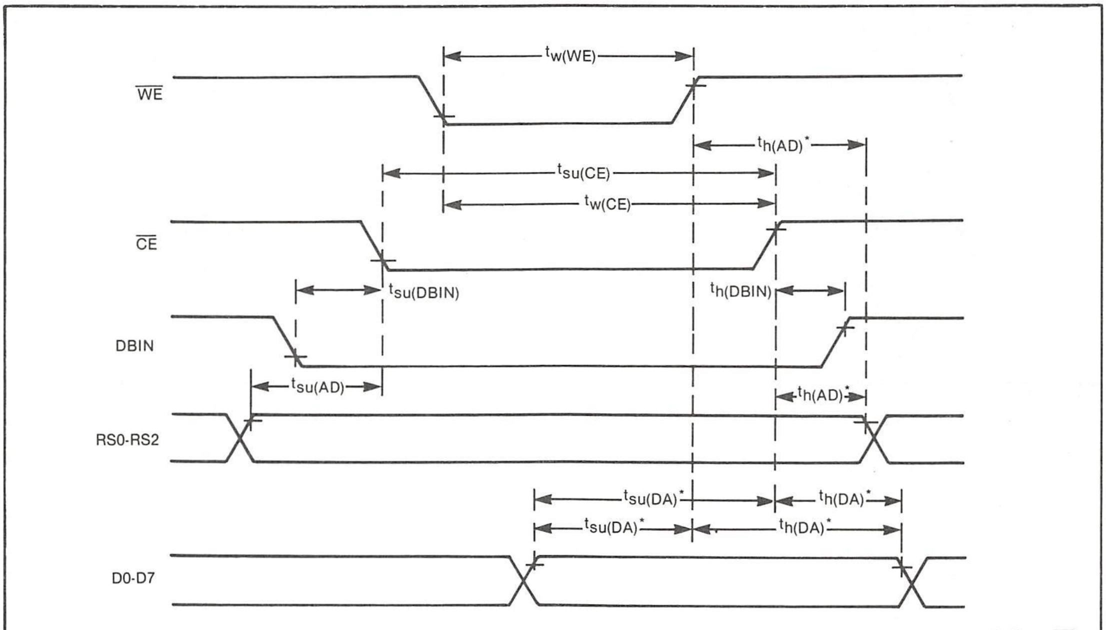
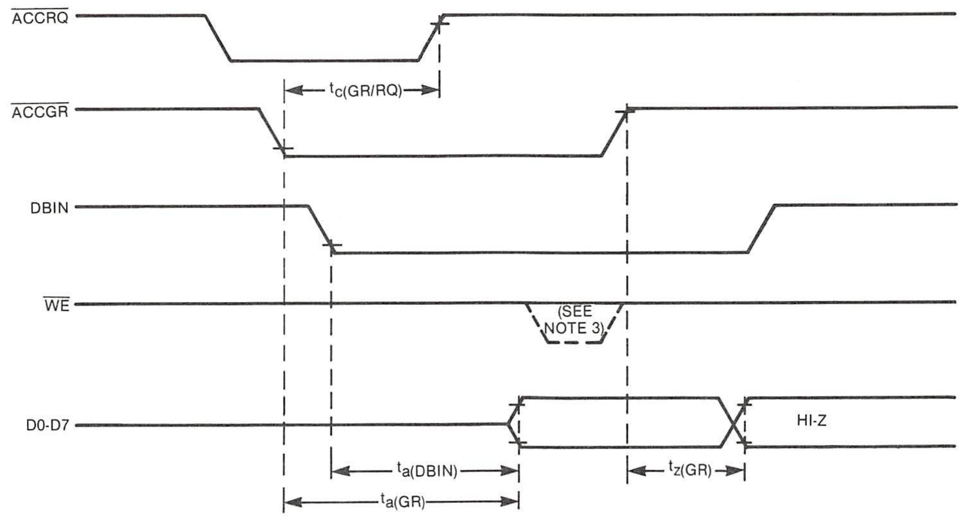


Figure 20. WD9914 READ CYCLE TIMING



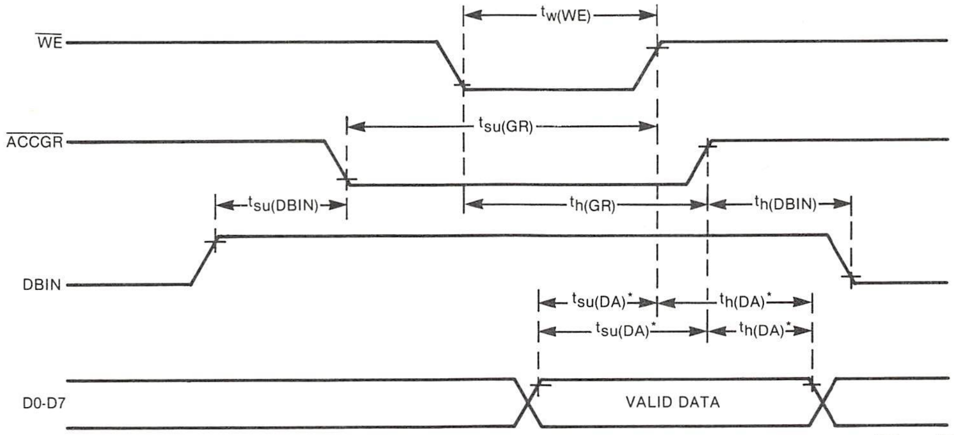
$t_{su(DA)}$, $t_{h(DA)}$, AND $t_{h(AD)}$ ARE ONLY APPLICABLE TO THE FIRST SIGNAL TO BECOME INACTIVE, WHETHER IT IS $\overline{WE}$ OR $\overline{CE}$.

Figure 21. WD9914 WRITE CYCLE TIMING



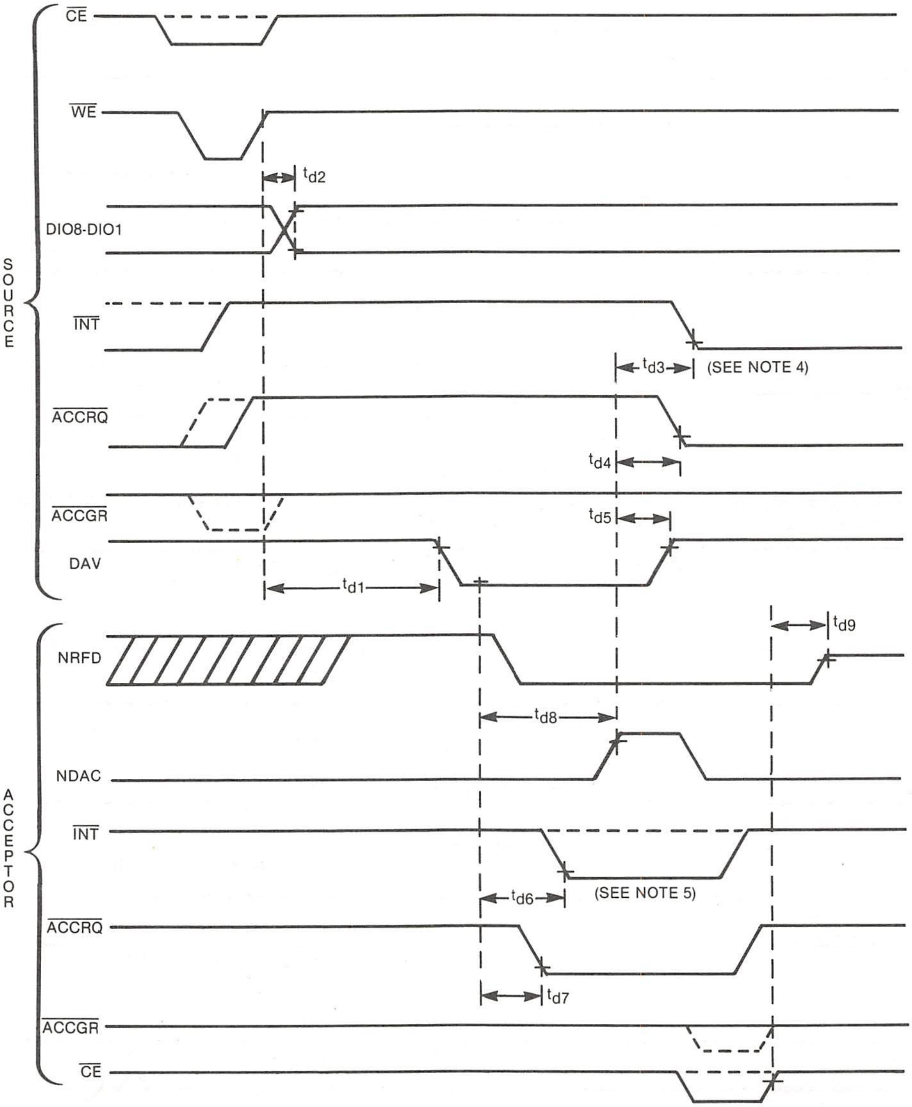
NOTE 3: A WRITE ENABLE PULSE MAY OCCUR IN A DMA READ OPERATION. A WRITE ENABLE PULSE, THEREFORE, MAY BE PROVIDED FOR SYSTEM MEMORY AND NEED NOT BE SUPPRESSED AT THE WD9914.

Figure 22. WD9914 DMA READ OPERATION



*$t_{su(DA)}$ AND $t_{h(DA)}$ ARE ONLY APPLICABLE TO THE FIRST SIGNAL TO BECOME INACTIVE, WHETHER IT IS $\overline{WE}$ OR $\overline{ACCGR}$.

Figure 23. WD9914 DMA WRITE OPERATION

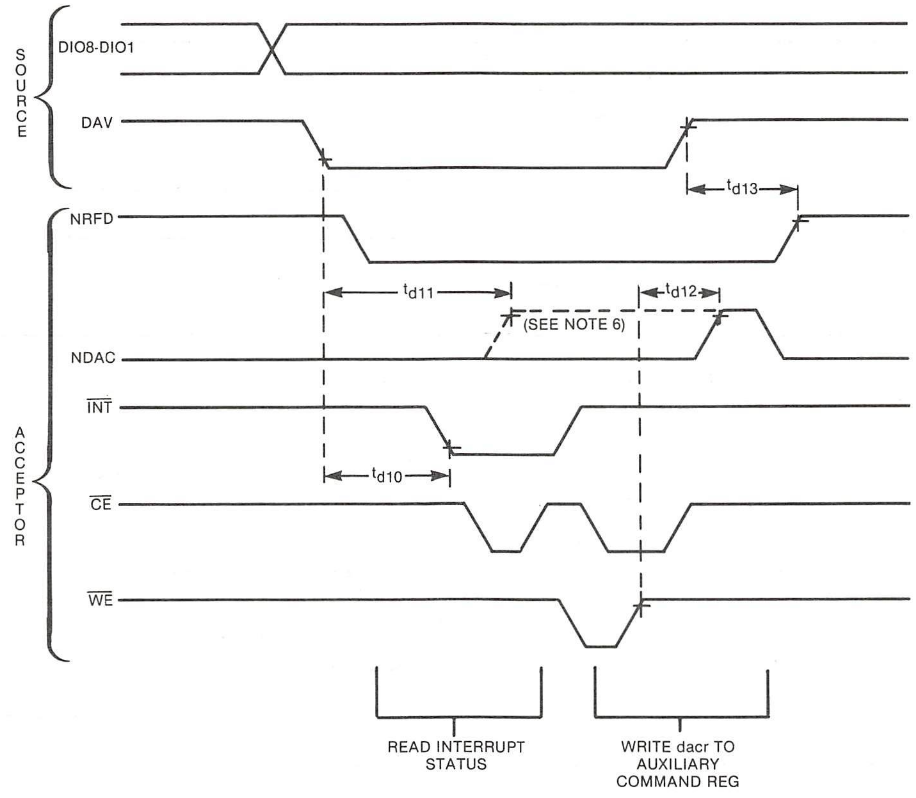


NOTES:

4: THE INTERRUPT LINE IS TAKEN LOW BY A BO INTERRUPT.

5: THE INTERRUPT LINE IS TAKEN LOW BY A BI INTERRUPT.

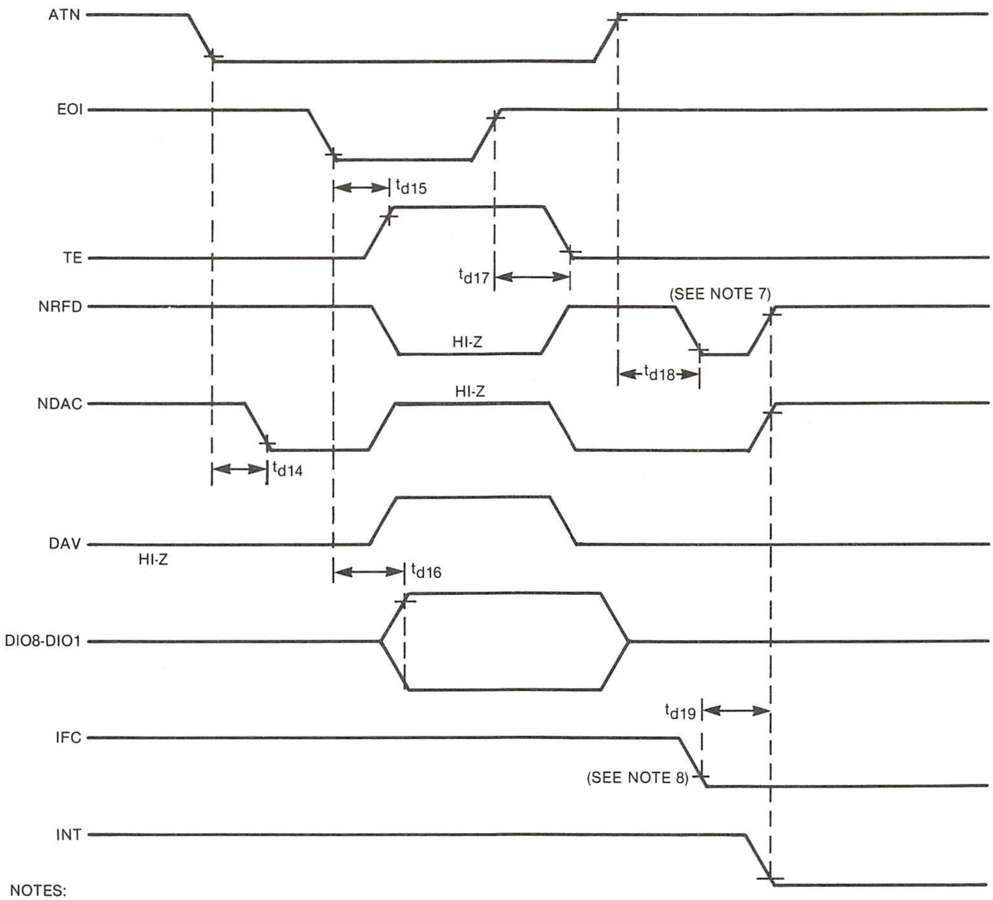
Figure 24. WD9914 SOURCE AND ACCEPTOR HANDSHAKE TIMING(S)



NOTES:

6: THE BROKEN LINE SHOWS THE WAVEFORM IF THERE IS NO DAC HOLDOFF. THE SOLID LINES ASSUME THERE IS A DAC HOLDOFF.

Figure 25. WD9914 ACCEPTOR HANDSHAKE TIMING "ATN" TRUE

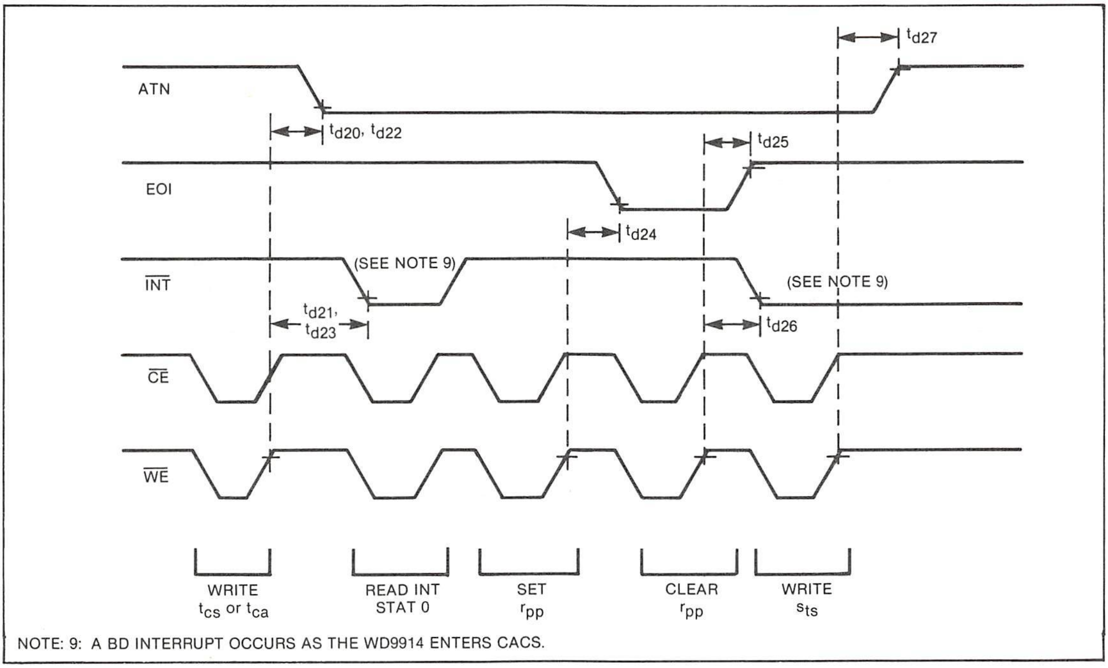


NOTES:

7: THIS ASSUMES THAT AN RFD HOLDOFF OCCURS.

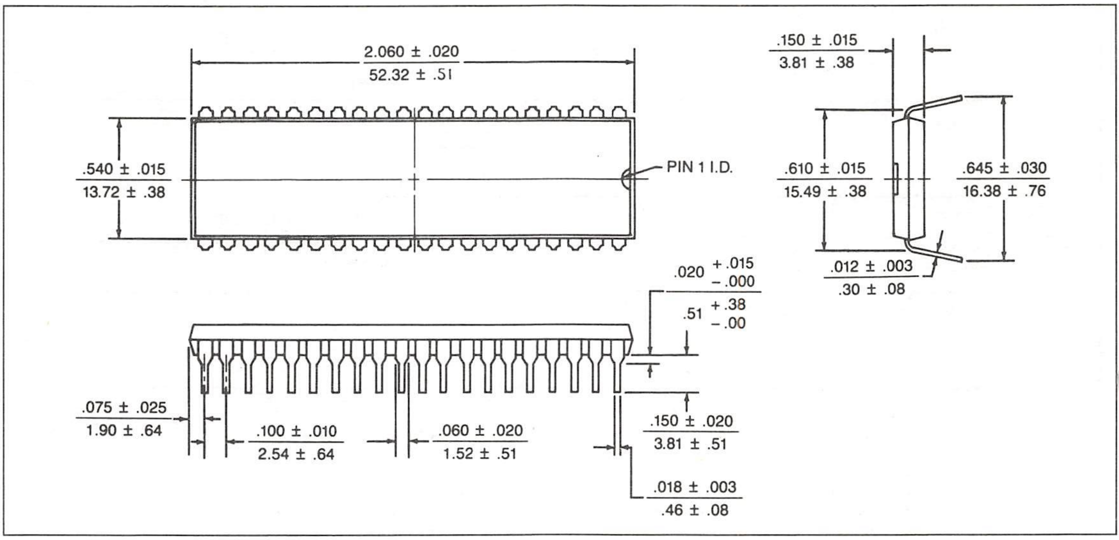
8: IFC CAUSES THE WD9914 TO BE UNADDRESSED AND AN IFC INTERRUPT OCCURS.

Figure 26. WD9914 RESPONSE TO 'ATN' AND 'EOI'

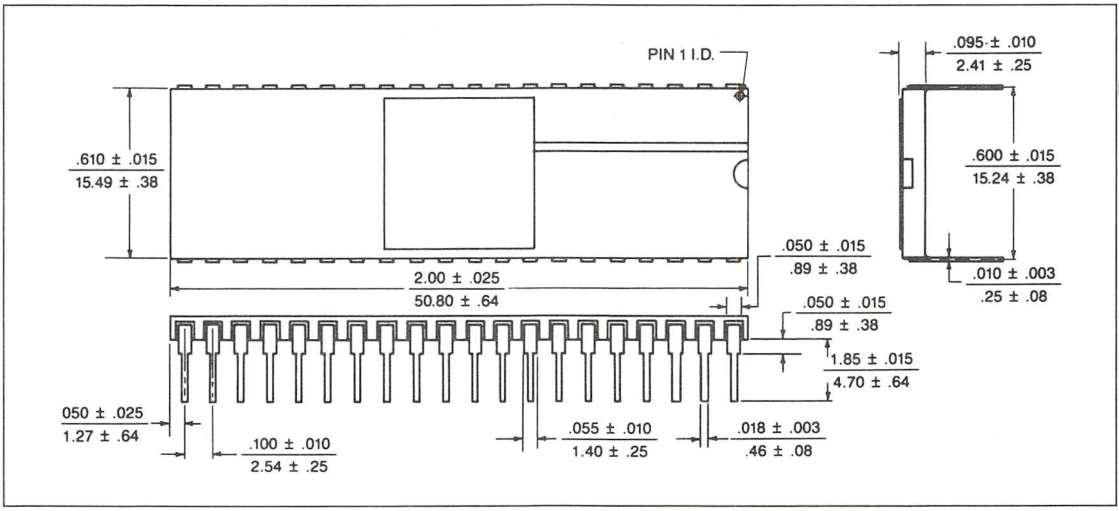


NOTE: 9: A BD INTERRUPT OCCURS AS THE WD9914 ENTERS CACS.

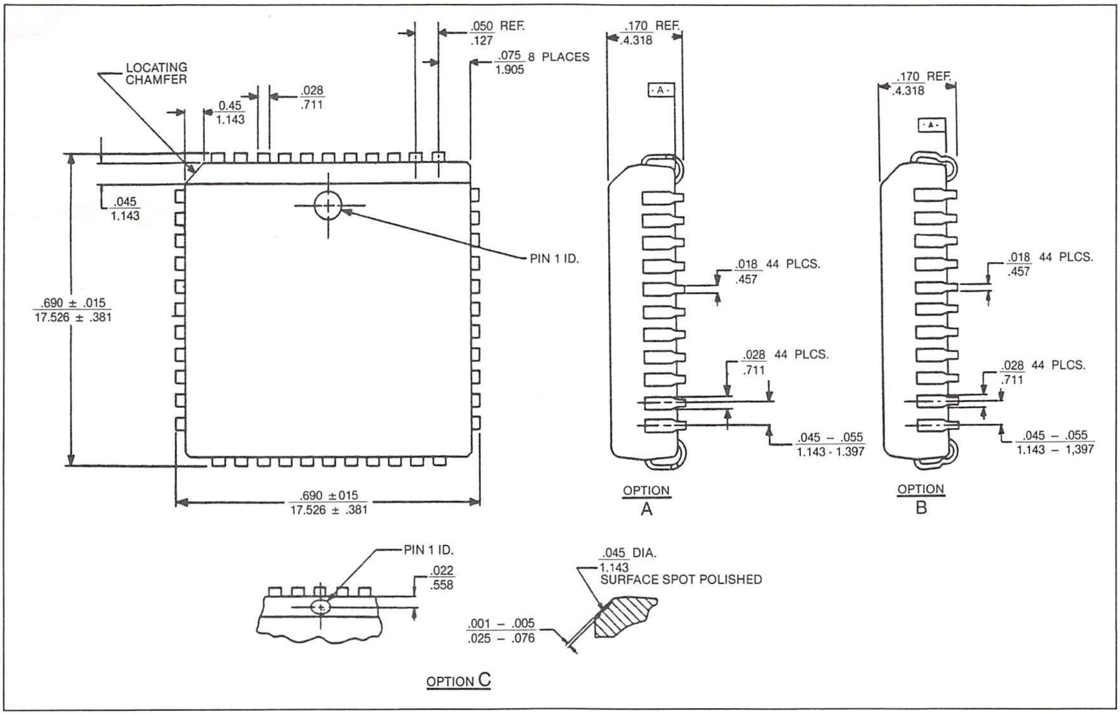
Figure 27. WD9914 CONTROLLER TIMING



40 LEAD PLASTIC "PL"



40 LEAD CERAMIC "AL"



44 LEAD PLASTIC "JM"

ORDERING INFORMATION:

WD9914 AL/PL/JM

AL	=	Ceramic Side Braze, 40-Lead
PL	=	Plastic (Encap), 40-Lead
JM	=	Plastic Chip Carrier-Leaded, 44-Lead

Please contact your local Western Digital Sales Representative or Call Toll Free (800) 847-6181 for package availability and price information.

Information furnished by Western Digital Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Western Digital Corporation for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Western Digital Corporation. Western Digital Corporation reserves the right to change specifications at anytime without notice.

Western Digital
2445 McCabe Way
Irvine, California 92714
(714) 474-2033 (714) 863-0102
FAX 714-660-4909 TLX 910-595-1139
For Information on WD Communications Products
Call: 1-800-NET LEADER (1-800-638-5323) or
(714) 474-2033, Ext. 4629 for General Information, or
Ext. 4900 for Technical Support.

WESTERN DIGITAL

Western Digital

2445 McCabe Way

Irvine, California 92714

(714) 474-2033   (714) 863-0102

FAX 714-660-4909   TLX 910-595-1139

For Information on WD Communications Products Call: 1-800-NET LEADER (1-800-638-5323)

For General Information Call (714) 474-2033, Ext. 4629, or For Technical Support Call (714) 474-2033, Ext. 4900

WD1240C 1/88 1M

79-000155-000