10. Debugger Interface
TABLE OF CONTENTS

10. Debugger Interface

The debugger interface (also called Song Board) provides a parallel communication link between the Dove workstation and a debugger machine (Dandelion). This board replaces the previous debugger board.

The main component of the debugger interface is an Intel 8255A-5, programmable peripheral interface. This component handles both handshakings and interrupts and can be directly accessed by the IOP.

Figure 10.1 illustrates a typical debugging system. A debugger interface is attatched to the debugger machine and the debuggee machine. Transmission cables transmit data between the two boards.

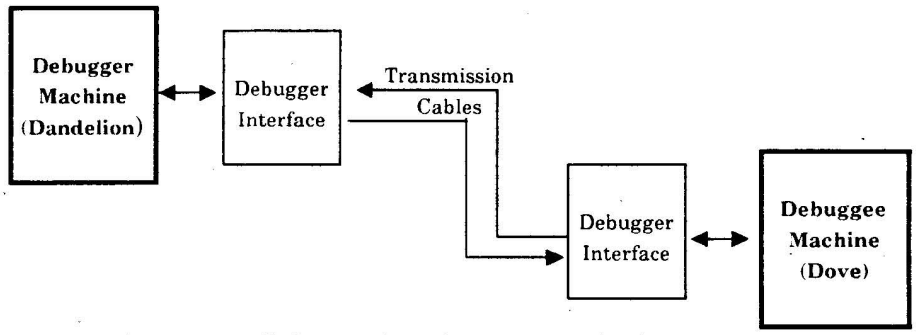


Figure 10.1. Debugger boards in a typical debugging system

10.1 Hardware

The debugger interface PWBA is housed in a $4\frac{3}{4}$-inch x $5\frac{1}{2}$-inch x 1-inch metal case and contains the programmable peripheral interface, line drivers, and line receivers. The total power consumption is about 250 mA. Figure 10.2 illustrates the debugger interface PWB assembly.

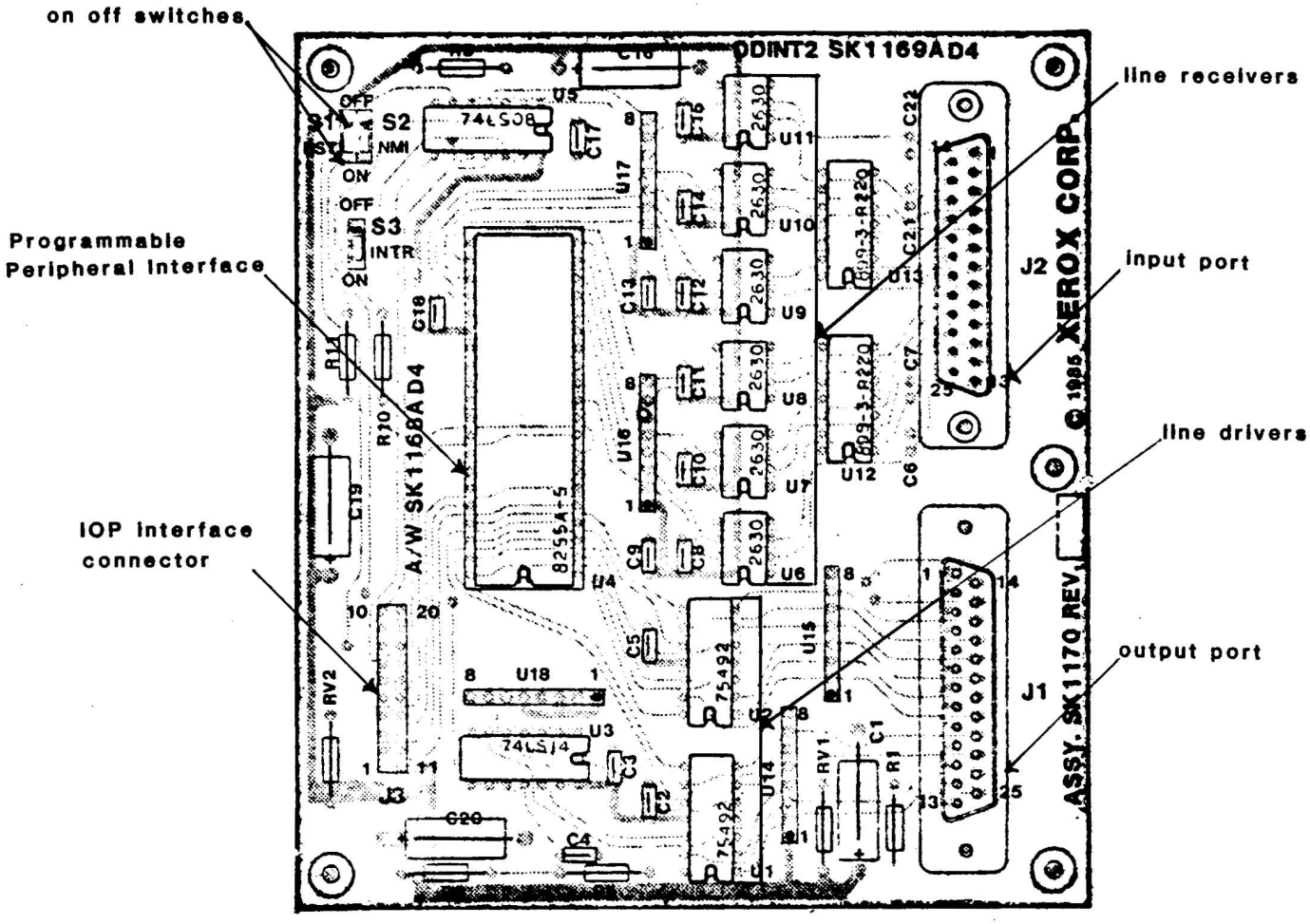


Figure 10.2 Debugger interface PWB assembly

All components on the debugger interface are off-the-shelf components and are socketed for easy servicing. Figure 10.3 illustrates the hardware, broken down into three functional groups: programmable peripheral interface, line drivers, and line receivers. For full schematics, refer to Appendix D.

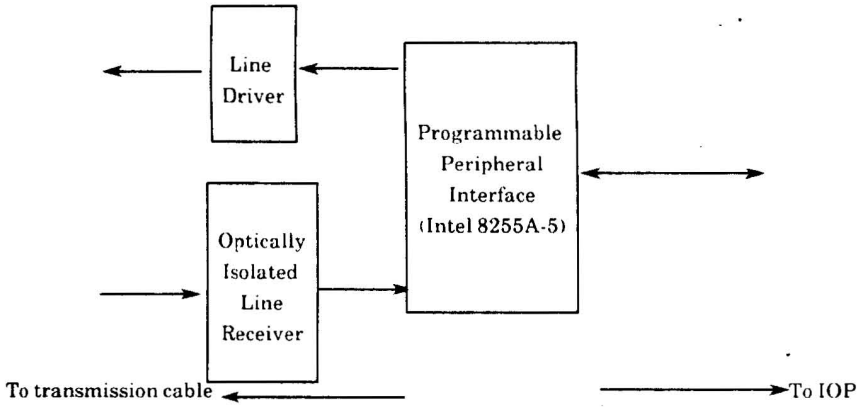


Figure 10.3. Debugger interface block diagram

10.1.1 Interface Connector

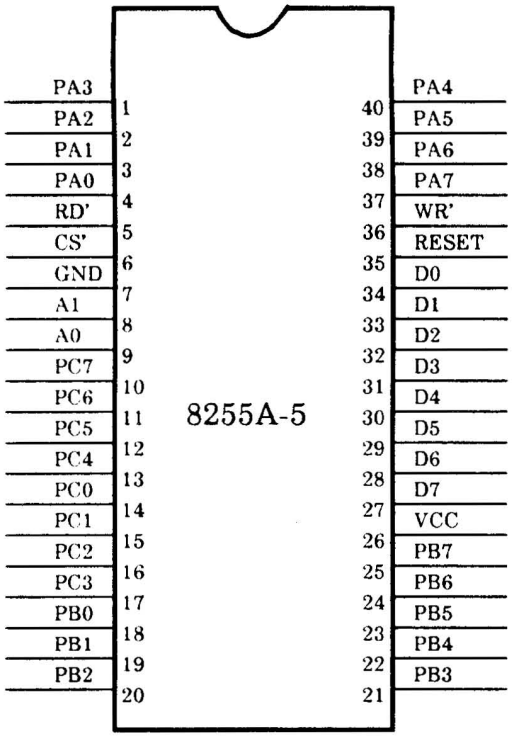
Table 10.1 lists and describes the signals of the of 20-pin connector chip that connects the debugger interface and the IOP.

Table 10.1. 20-Pin Connector Description

Signal	Type	Function
IPRD'	Input	A low on this input pin enables the debugger interface to send the data or status information to the IOP on the data bus.
IPWR'	Input	A low on this input pin enables the IOP to write data or control words to the debugger interface.
IPSEL'	Input	A low on this input pin enables the communication between the debugger interface and the IOP.
IPA1 and IPA0	Input	These input signals address the ports (A, B, and C) or the control register.
RESET	Input	A high on this input pin causes the debugger interface to be reset. During reset, the control word is cleared and all of the ports are set to input mode.
InINTR	Output	An active high signal indicates a byte was received.
IPReset'	Output	An active low signal resets (boot) IOP. This signal should be disabled in debugger side using the switch located on the debugger box.
OutINTR	Output	An active-high signal indicates that the output port is empty and ready to accept another byte from the IOP.
NMI'	Output	An active low signal sends an NMI to the IOP.
IPD7-0	I/O	Lines 7-0 are data buses on the debugger interface. These lines carry data between the debugger interface and the IOP.

10.1.2 Programmable Peripheral Interface

The programmable peripheral interface (PPI), is an Intel 8255A-5. Figure 10.4 illustrates the 40-pin interface chip and also lists the pins and signals.



Pin Name	Pin Signal
D7-D0	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
CS'	CHIP SELECT
RD'	READ INPUT
WR'	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
VCC	+5 VOLTS
GND	0 VOLTS

*See Intel data book for a detailed description of this component

Figure 10.4. PPI pins and signals

10.1.3 Line drivers

The cable between two debugger interfaces is recommended but not required to be at most 10 feet. In order to drive this cable, open-collector Hex LED drivers, DS75492, are used as transmission line drivers. The output of the drivers is individually pulled up with 560 ohm resistors. Figure 10.5 illustrates the line driver circuits.

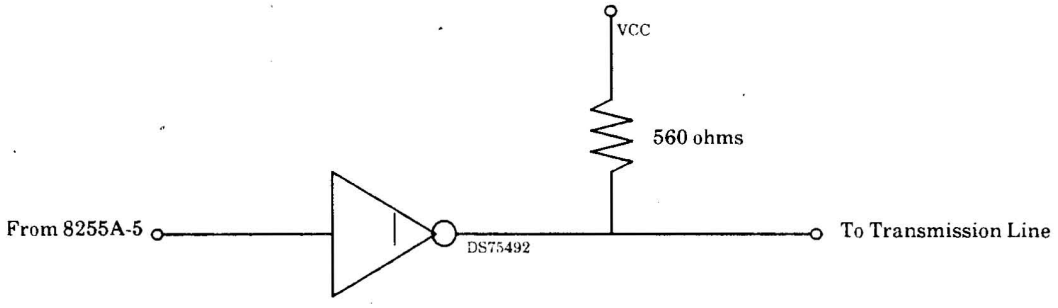


Figure 10.5. Line driver circuit

10.1.4 Line Receivers

To eliminate ground loops between the debugger machine and the debuggee machine, the opto-isolator, HPCL2630, is used as the line receiver. The characteristics of this device depend on the input voltage, input current, and pull-up resistor.

The values of the resistors used with line drivers and receivers are carefully balanced for proper and reliable operations. Figure 10.6 illustrates the line receiver circuits.

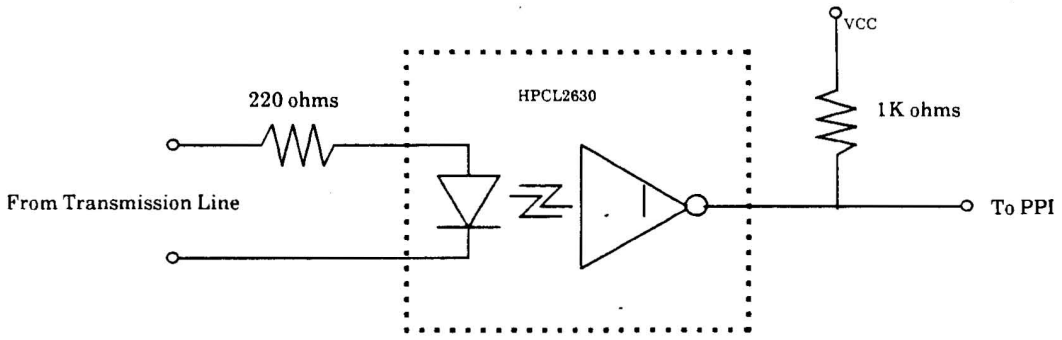


Figure 10.6. Line receiver circuits

10.2 Theory of Operations

The debugger interface performs two major tasks: sending a byte and receiving a byte. This section describes both of these tasks and the steps that must take place before either action can occur. Refer to Intel's *1984 Microprcessor Handbook Volume II* for further information on modes.

10.2.1 Sending and Receiving Data

Figure 10.7 illustrates how data is sent from the debuggee to the debugger.

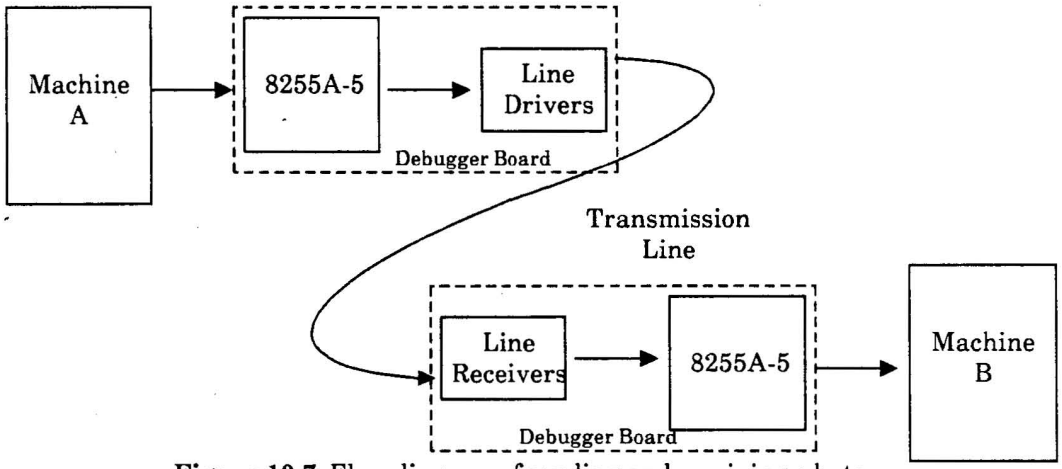


Figure 10.7. Flow diagram of sending and receiving a byte

1. Machine A writes a byte of data to the output port of the PPI attached to machine A.
2. This byte is transmitted through the line drivers and received by the line receivers.
3. The data is obtained by the PPI attached to machine B by handshaking.
4. Machine B retrieves the byte by reading from the PPI input port.

10.2.2 Handshaking

Port A of the PPI is programmed as a mode 1 output port, and port B is programmed as a mode 1 input port. Both ports use handshaking for data transmissions. In mode 1, handshaking is done by PPI port C, internally. Figure 10.8 illustrates the connection between the input port of one debugger interface and the output port of the other debugger interface. The line drivers and receivers are omitted. Table 10.2 lists the steps involved in the handshaking procedure.

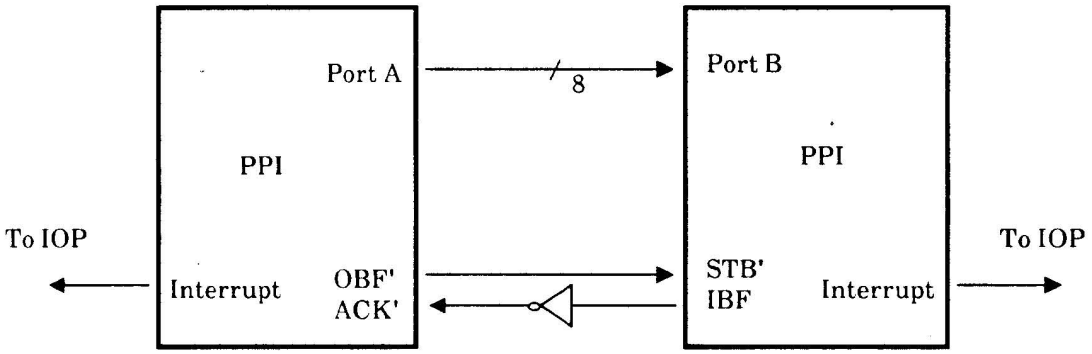


Figure 10.8. Transmission cable flow

Table 10.2. Handshaking Procedure

Output Port	Input Port
- IOP writes a byte to Port A -	
1. Lower OBF'	
PPI turns off the Interrupt (LOW)	
	2. Detect LOW in STB'
	3. Port B latches the byte into the buffer
	4. Raise IBF
	Notify IOP by an interrupt (HIGH).
5. Detect Low in ACK'	
6. Raise OBF'	
Notify IOP by an interrupt (HIGH)	
	- IOP reads the byte from Port B -
	PPI turns off the Interrupt (LOW)

10.3 Programmer Interface

Improper programming of the debugger interface may result in damage to the circuits on the board. The following sections describe information necessary to program the debugger interface.

10.3.1 Addressing the Debugger Interface

The debugger interface uses four address spaces. Table 10.3 lists the I/O address spaces in the IOP.

Table 10.3. Basic Operations of the Debugger Interface (PPI)

Address	RD'	WR'	CS'	**Input Operation (Read)**
70H	0	1	0	Port A X Data Bus
72H	0	1	0	Port B X Data Bus
74H	0	1	0	Port C X Data Bus
				Output Operation (Write)
70H	1	0	0	Data Bus X Port A
72H	1	0	0	Data Bus X Port B
74H	1	0	0	Data Bus X Port C
76H	1	0	0	Data Bus X Control
				Disable Function
	X	X	1	Data Bus X 3-State
76H	0	1	0	Illegal Condition
	1	1	0	Data Bus X 3-State

10.3.2 Programming the PPI

The Programmable Peripheral Interface must be programmed in the following way for correct operations to occur.

10.3.2.1. Operational Descriptions and Configurations

Table 10.4 lists the debugger interface configurations.

10.3.2.2. Initialization

While the PPI is being reset, the control register is cleared and all ports are set to input mode. To configure the PPI as the debugger interface, the following steps are necessary.

1. Configure the ports to the following modes:

   Port A to mode 1 output port
   Port B to mode 1 input port
   Port C, bits 4 and 5, to output port
   Data, A6H, should be written into the control register, 76H.

2. Set the interrupts.

   Set the input interrupt: Data, 05H, should be written into the control register, 76H.

Set the output interrupt: Data, 0DH, should be written into the control register, 74H.

Table 10.4. Configuration of the Intel 8255A-5 in the debugger interface

Port	Pin #	I/O	Use
A	PA7	OUT	Output Port
	PA6	OUT	
	PA5	OUT	
	PA4	OUT	
	PA3	OUT	
	PA2	OUT	
	PA1	OUT	
	PA0	OUT	
B	PB7	IN	Input Port
	PB6	IN	
	PB5	IN	
	PB4	IN	
	PB3	IN	
	PB2	IN	
	PB1	IN	
	PB0	IN	
C	PC7	OUT	OBF_A'
	PC6	IN	ACK_A'
	PC5	OUT	I/O
	PC4	OUT	I/O
	PC3	OUT	$INTR_A$
	PC2	IN	STB_B'
	PC1	OUT	IBF_B
	PC0	OUT	$INTR_B$

Signal Key

OBF_A'	=	Output buffer full'
ACK_A'	=	Acknowledge input
I/O	=	Input/Output
$INTR_A$	=	Interrupt acknowledge. This signal is reset by the falling edge of Write'.
STB_B'	=	Strobe input
IBF_B'	=	Input buffer full
$INTR_B$	=	Interrupt Acknowledge. This signal is reset by the falling edge of Read'.

10.3.2.3. Sending a Byte

First, check the status of the output port, Port A, by reading Port C. A high in bit 3 of Port C indicates that the output port is empty and ready to send the next byte. Write the byte to Port A.

10.3.2.4. Receiving a Byte

Check the status of the input port, Port B, by reading Port C. A high in bit 0 of Port C indicates that the input port is full and ready to be read.

10.3.2.5. Sending Boot (Reset) Signal to Debuggee Machine

In order to boot the debuggee machine, bit 4 of port C in the debugger side must be toggled.

1. Set bit 4 of port C: Write data, 09H, to control register C, 76H.
2. Wait for about 50 ms.
3. Reset bit 4 of port C: Write data, 08H, to control register C, 76H.

10.3.2.6. Sending an NMI Signal

In order to send NMI, bit 5 of port C must be toggled.

1. Set bit 5 of port C: Write data, 0BH, to control register, 76H.
2. Reset bit 5 of port C: Write data, 0AH, to control register, 76H.

10.3.3 Timing

Table 10.5 lists the timing characteristics for a read and a write. Figure 10.9 illustrates the timing for a read, write, and a read-write.

Table 10.5. Timing Characteristics

	Symbol	Parameter	PPI		Unit
			Min	Max	
Read	tAR	Address is stable before a Read	0		ns
	tRA	Address is stable after a Read occurs	0		ns
	tRR	Read pulse width	300		ns
	tRD	Data is valid from Read		200	ns
	tDF	Data floats after Read	10	100	ns
	tAW	Address is stable before a Write	0		
	tWA	Address is stable after a Write	20		
Write	tWW	Write pulse width	300		
	tDW	Data valid to Write	100		
	tWD	Data valid after Write	30		ns
Read/Write	tRV	Time between Reads and/or Writes	850		ns

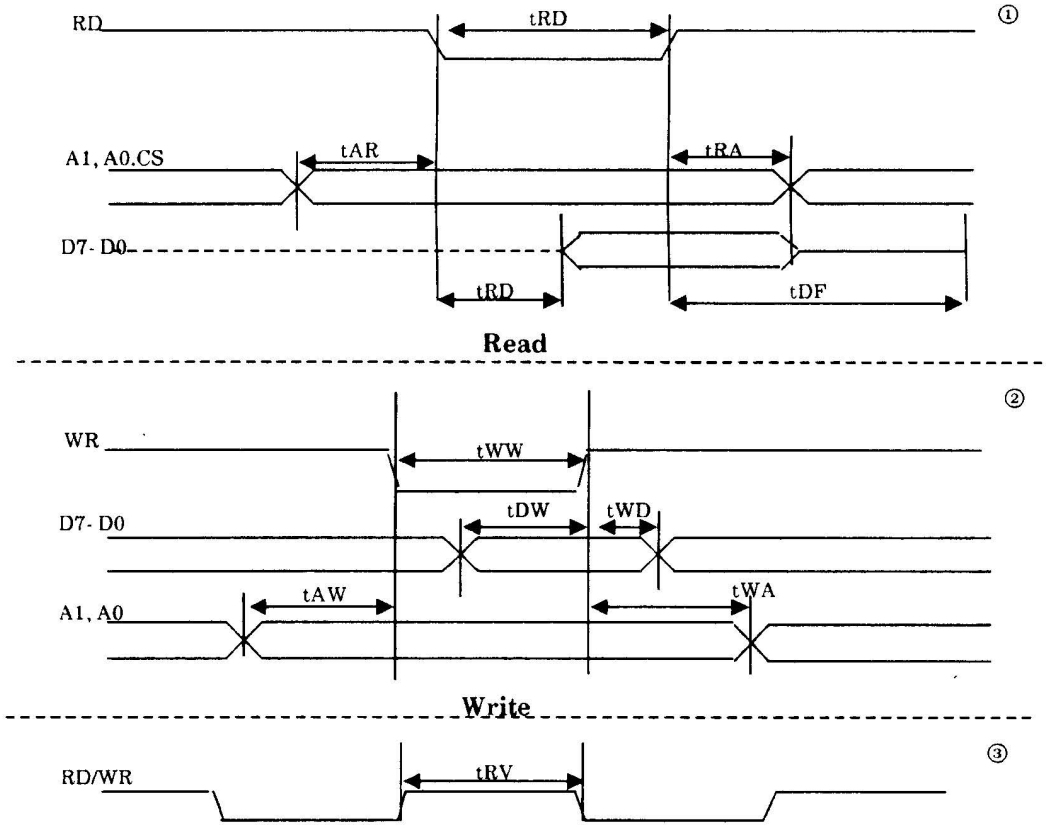


Figure 10.9. Read and write timing