W I C A T

System 150

Hardware Reference Manual

187-055-201B

27 September 1983

WICAT SYSTEMS INCORPORATED Orem, Utah

Copyright Statement

Receipt of this manual must not be construed as any kind of commitment, on the part of WICAT Systems Incorporated, regarding delivery or ownership of items manufactured by WICAT.

This manual is subject to change without notice.

WARNING: The equipment described in this manual generates, uses, and can radiate radio frequency energy, and if not installed in accordance with instructions provided in the hardware documentation for the equipment, may interfere with radio communications. Furthermore, the equipment has been tested and found to comply with the limits for a Class A computing device pursuant to subpart J, Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of the equipment (described in this manual) in a residential area is likely to cause interference. Where the equipment will be used in a residential area, it is the user's responsibility to ensure that any interference is corrected.

Revision History

First Printing July 1982
Second Printing September 1983

The Purpose of This Manual

This document provides the technical information users will need should they want to modify or program the System 150 PC boards to interface with peripheral devices not supplied by WICAT Systems, Inc.

Intended Audience

Field service technicians, OEM representatives, and WICAT maintenance personnel. Readers must be knowledgable in electronics and familiar with the hardware terminology of computer science.

Hardware Reference Manual

Table Of Contents

CHAPTER 1

SYSTEM OVERVIEW

1.1 INTRODUCTION

This manual is organized into chapters or self-contained modules, one for each printed circuit board that can possibly constitute a System 150. Each module or chapter covers the operation of a specific PC board. You need read only those modules that pertain to the boards found with your particular system configuration. If, for example, your system has no cartridge tape subsystems, you need not read chapter six.

The system overview (this chapter) is common to all System 150 configurations.

1.2 FEATURES OF THE SYSTEM 150

The WICAT System 150 is a 68000-based microcomputer system with mainframe capabilities.

- o Central Processing Unit (CPU) (1.4.1)
- o Memory (1.4.2)
- o Storage and storage backup (1.4.3)
- o Several standard and optional peripherals (1.4.4)

1.3 SYSTEM CONFIGURATION

The System 150 is self-contained in a video terminal and has a detachable keyboard as shown in figure 1.1. The left front of the unit contains a 5-1/4 inch Winchester disk drive as the primary boot and storage device, and either a 5-1/4 inch floppy

disk drive as a secondary boot and memory device. The cabinet contains a card cage with six slots for printed circuit boards. Those boards are the subject of this manual. Additional Winchester disk drives and/or a DEI cartridge tape drive may be housed in an accompanying Disk Expansion Unit (DEU).

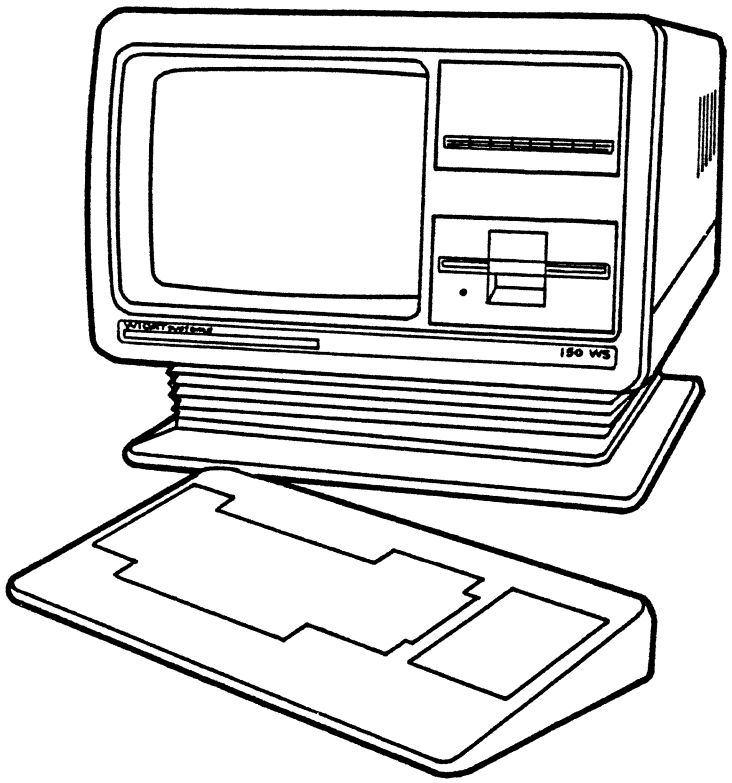


Figure 1.1 System 150 (common configuration)

1.4 SYSTEM COMPONENTS

1.4.1 Central Processing Unit (CPU)

The processor for the System 150 is the Motorola 68000 microprocessor, which runs at 8 MHz and executes up to one million instructions per second. Thirty-two bit internal registers support 32-bit data operations.

See Chapter 2 for more information on the CPU.

1.4.2 Memory

The System 150 is equipped with a single memory board containing 256 Kbytes of Dynamic Random Access Memory (DRAM) that is expandable to 512 Kbytes by fully populating the board. Additional memory boards may be added to increase the available memory space to 1.5 Mbytes dynamic ECC RAM.

See Chapter 4 for more information.

1.4.3 Storage

Mass storage, with the optional DEU, can include up to three 5-1/4 inch Winchester disk drives, a DEI cartridge tape drive, and/or one or more 5-1/4 inch floppy disk drive(s) for backing up and porting files.

See Chapters 5 and 6 for more information.

1.4.4 Other Elements

Other components of the System 150 are:

- 5 RS-232 C serial interfaces (1.8.2)
- 16-bit parallel interface (1.8.2)
- Battery-backed calendar clock (1.4.41)
- Intelligent disk controller (1.4.4.2)

- o IEEE 488 general purpose interface bus (GPIB) (optional) (1.8.3 and Chapter 7)

1.4.4.1 Battery-backed Calendar Clock -

This clock, located on the I/O board, is a real time calendar clock for the system. An on-board battery powers this clock in the event of a system failure or unexpected loss of power.

1.4.4.2 Intelligent Disk Controller -

The Winchester controller board controls the Winchester disk and performs local interpreting functions such as editing, checking input validity and decoding complex commands.

1.5 PLACEMENT OF BOARDS

The motherboard in the standard CPU chassis contains the PC boards shown on table 1-A. (See the appropriate module for a detailed explanation.)

- o CPU board
- o I/O board
- o Memory board
- o Controller boards (only WFC is standard)
  1. Winchester and Floppy Disk (WFC)
  2. DEI cartridge magnetic tape drive

Placement is significant for designating or assigning bus priority, as noted in 1.6.2.

Table 1-A
S150 Placement of Boards

Slot No.	PC Board Identity
6	Primary disk (SMD or WFC)
5	WFC or I/O (1st)
4	2nd I/O or memory (optional)
3	memory (optional)*
2	memory
1	CPU

*Memory options: 256 or 512 Kbytes

If you must replace a board, observe the stuffing order noted in table 1-A.

1.6 PC BOARD INTERACTION

The **bus structure** is a common group of circuit paths over which input and output signals are routed. This structure enables communication between the CPU board and the other PC boards.

1.6.1 Explanation Of Bus Structure

The PC boards and the CPU board are interconnected via the motherboard, located in the base of the chassis. The CPU board controls the bus.

There are three kinds of buses:

1. The **address bus** carries the signals needed to define any of the possible memory or I/O locations in the system.
2. The **data bus** carries all instructions and data.

3. The control bus carries signals generated by the CPU used to direct the action of the other elements in the system.

1.6.2 Bus Control

The MC68000 microprocessor allocates CPU time for requesting processes and devices. The basic theory of allocation is as follows: First, the CPU receives a request signal for bus use. Second, based on internally defined priorities, the request is either granted immediately or delayed until previous or higher priority requests are processed. Finally, when the request is granted, the requesting device acknowledges to the CPU that it is using the bus, and the cycle continues.

The device or process using the bus is called the bus master. (WICAT computers allow only one bus master at a time.) A device called the slave then receives data from or transmits data to the bus master.

1.6.3 Serial Priority Bus Control

Priority functions allow bus masters to break deadlocks that occur when more than one master concurrently requests the bus. The System 150 supports bus mastery for the CPU board, the I/O board, and the various controller boards. The CPU board (slot 1) is the default master and is on the lowest priority end of the motherboard. The SMD board (slot 12) is the highest priority.

BPRN is the signal for bus priority in; BPRO is the signal for bus priority out. BPRN indicates to a particular master that no higher priority master is requesting the use of the bus.

The motherboard contains slots for twelve PC boards (table 1-A). Pin 15 on each slot is BPRN; pin 16 is BPRO. In front of each slot on the motherboard is an extractable multibus jumper pin. If left in place, the jumper pin completes the circuit for the daisy-chain relay of the bus priority signal. A PC board with the jumper pin left in place cannot drive BPRO high and therefore cannot be bus master.

For example, to qualify the primary disk PC board as a

possible bus master, extract JP12. Do the same thing with JP11, JP10, and JP1 to qualify the other bus masters. The relay of the bus priority signal for these boards thereafter takes place on the circuitry of the respective board.

1.7 SYSTEM DATA FLOW

Primary data communication to the System 150 is via an external terminal connected to port zero of the connector port panel (CPP) (see figure 1.2).

FLOW

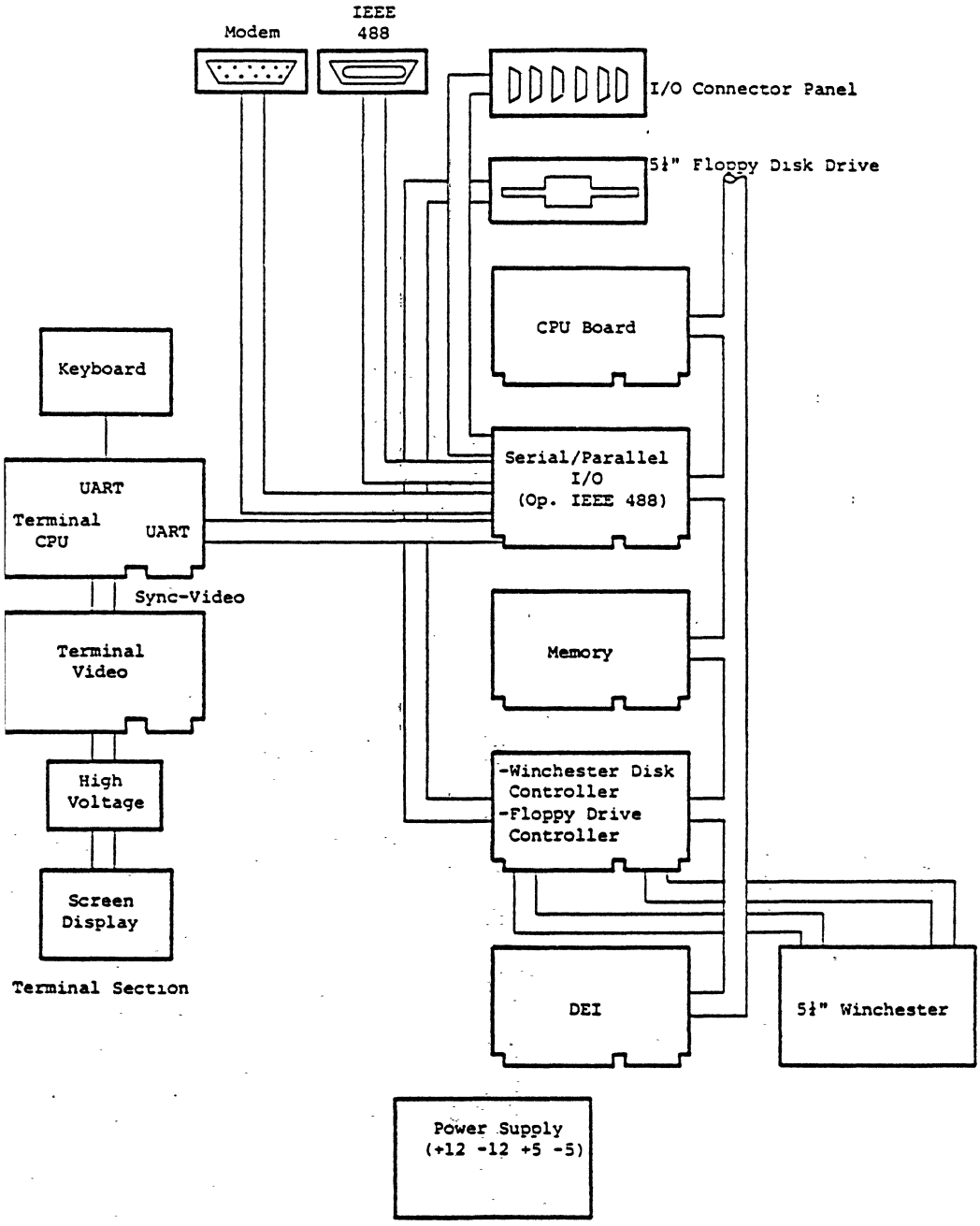


Figure 1.2 System 150 Data Flow Chart

1.8 CONTROLLER INTERFACE

1.8.1 Serial Interfaces

Serial interfaces are used as input/output ports for various peripheral devices such as printers, terminals, etc. The interfaces are located on the I/O board (see module 3) and conform to the standard RS-232 C to ensure asynchronous data transfers. Figure 1.3 is a diagram of the RS-232 C port.

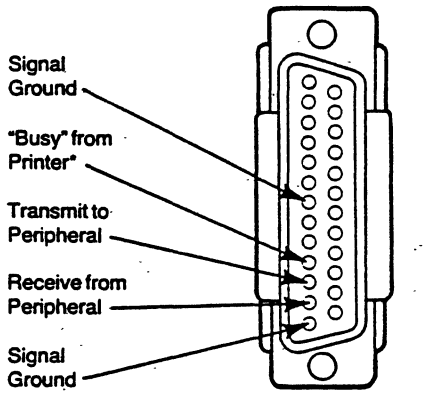


Figure 1.3 The RS-232 C Port

NOTE

Pins 1 and 7 are signal grounds. Do not connect them as a chassis ground for peripherals and terminals.

The System 150 receives on pin 2 and transmits on pin 3.

The TTY device driver automatically handles XONXOFF data handshaking, i.e., control of data flow. Use pins 2, 3, and 7 in this configuration. Pin 4 is used for "busy" handshaking. It may be necessary to remove a jumper on the I/O board to use hardware handshaking on each serial port. Use pins 3, 4, and 7 in this configuration.

1.8.2 Parallel Interface

The parallel interface is a 16-bit parallel port organized as two 8-bit bidirectional ports and set up to act as a standard Centronics interface. The port is used as shown in figure 1.4.

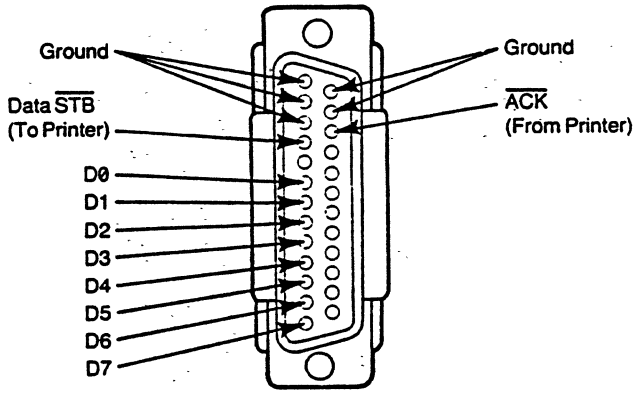


Figure 1.4 The Parallel Port

NOTE

WICAT does not use a busy line in the parallel interface. Each time ACK is received, the next character in the buffer is strobed on the data bits.

1.8.3 IEEE 488 Interface (Optional)

Additional ICs to make up this option can be inserted on the I/O board (see chapter 3). The IEEE 488 interface is a standard General Purpose Interface Bus (GPIB), allowing the use of other external peripherals. The IEEE 488 interface connector and the WICAT memory map using this option are covered in chapter 7.

CHAPTER 2

CPU BOARD

2.1 INTRODUCTION

This module explains the functions of the CPU board (part number 810-077-001).

2.2 CPU BOARD CONFIGURATION

The CPU board comprises four main areas of circuitry.

- o CPU chip and supporting circuits
- o Memory mapping registers
- o On board ROM
- o Multibus interface

2.3 MICROPROCESSOR CIRCUITRY

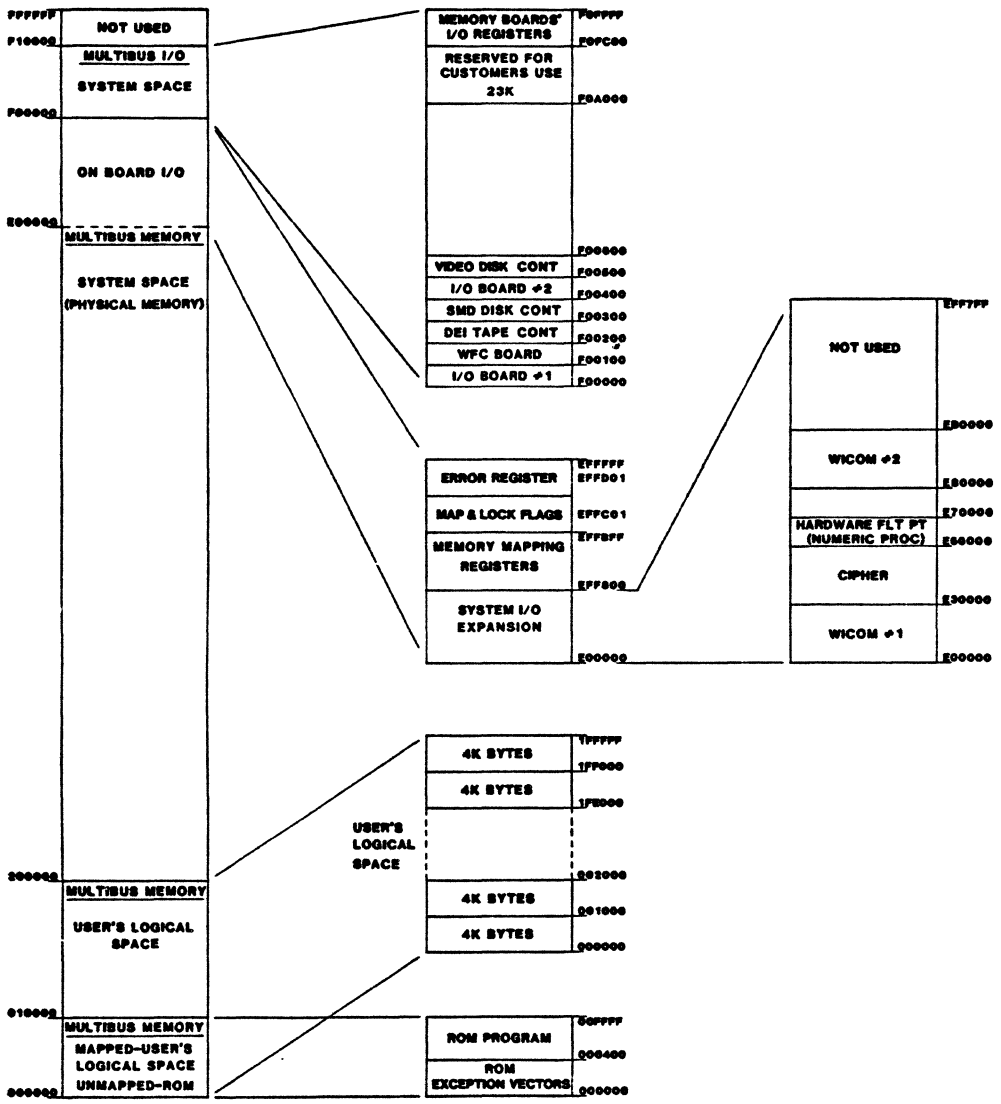
The circuitry of the CPU board is divided according to two principal functions: the board fetches and sends data (bus-related functions), and the board decodes signals and executes commands (on-board functions).

The CPU board contains the Motorola MC68000 Microprocessor, which directs control, logic, and arithmetic operations. The processor circuitry comprises the microprocessor and its associated buffers; system clocks; and bus error, interrupt, and arbitration logic.

For additional information, refer to Motorola's publication entitled: MC68000 16-bit Microprocessor User's Manual.

2.3.1 System Memory Map

Table 2-A
Memory Address Map



2.3.2 MC68000 Peripheral Control

The I/O board contains the only 6800-family device used on the System 150. The MC68000 provides three lines for interfacing 6800-family synchronous bus peripheral devices:

1. **Enable** (E) is a clock signal that synchronizes transactions between the processor and MC6800 peripherals. The clock period is ten times the system clock (10 x 125nS = 1.25uS) and has a 60/40 duty cycle (6 clocks low, 4 clocks high).

2. **Valid Peripheral Address** (VPA) is asserted by the peripheral device when it recognizes its address on the address bus. VPA L, used to distinguish between auto and nonautovectored interrupts, is unavailable on the Multibus.

3. **Valid Memory Address** (VMA) is asserted by the processor in response to the assertion of VPA during an MC6800 peripheral data transfer.

E, VMA, and VPA are normally on an undefined pin of the Multibus (P2 connector).

2.4 THE MULTIBUS INTERFACE

2.4.1 Bus Arbitration Circuitry

The CPU bus arbitration scheme conforms to the Multibus specification for serial priority. Serial priority involves the CPU as the default bus master. The I/O board and the various controller boards can gain control of the bus according to bus priority signals governed by the software. Priority functions allow bus masters to break deadlocks that occur when more than one master concurrently requests the bus. The CPU board (slot 1) is on the lowest priority end of the motherboard. The SMD board (slot 12) is the highest priority.

BPRN is the signal for bus priority in; **BPRO** is the signal for bus priority out. The BPRO of each master is daisy chained to the BPRN signal of the next lower priority master. The BPRN signal tells a particular master that no higher priority master is requesting the use of the bus.

Below is the scheme used in the System 150 for resolving

serial priority.

SERIAL PRIORITY BUS CONTROL

1. A master requests the bus by driving BPRO high
2. This request disables the BPRN of all lower priority masters.
3. The CPU reads this BPRO high signal and drives its BPRN high
4. The CPU releases the busy line by driving the busy line high.
5. The requesting master sees the CPU busy line go high and asserts busy by qualifying BPRN.
6. When finished with the bus, the master drives BPRO low and gives up the bus.

The motherboard contains slots for six PC boards (see table 1-A). Pin 15 on each slot is BPRN; pin 16 is BPRO. In front of each slot on the motherboard is an extractable multibus jumper pin. If left in place, the jumper pin completes the circuit for the daisy-chain relay of the bus priority signals. A PC board with the jumper pin left in place cannot assert BPRN as high and therefore cannot be bus master.

With the appropriate jumper pin extracted, the software needed to designate bus master priority can be effective in designating priorities for bus masters. For example, to enable the primary disk PC board as a master, extract JP6. Do the same thing with JP5, JP4, and JP1. The relay of the bus priority signal for bus master PC boards thereafter takes place on the circuitry of the pertinent board.

2.4.2 Interrupt Circuitry

The CPU board supports seven levels of interrupts, INT0/ through INT6/. INT0/ has the highest priority and is the only nonmaskable interrupt. Multibus INT0/ corresponds to 68000 interrupt level

7. Multibus INT6/ corresponds to 68000 level 1. Multibus INT7/ has no correspondence to a 68000 interrupt. All interrupts are autovectored to addresses designated by the MC68000. (For additional information see section 2.4.5 and the appropriate section of Motorola's MC68000 16-Bit Microprocessor User's Manual.)

2.4.3 Interrupt Control

Interrupts are a subset of a more general class of operations called exceptions. Interrupts are always autovectored, meaning the processor generates the interrupt vector number internally, as a function of the interrupt level.

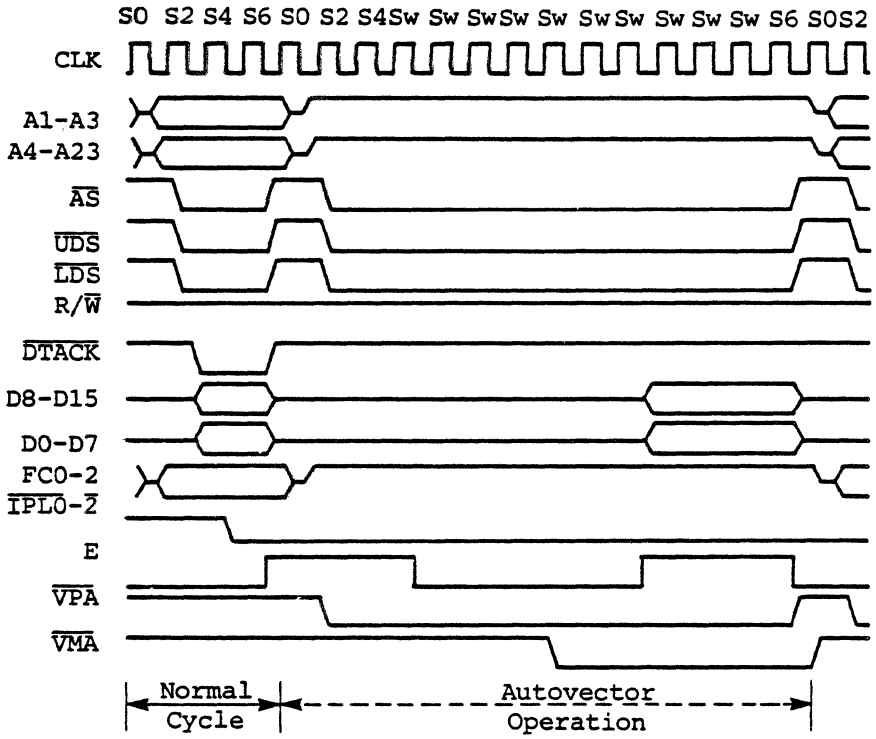


Figure 2.1 Signal Activity During an Autovectored Interrupt Sequence

After the interrupt vector number is obtained, the processor saves the status and return address on the system stack and then uses the vector number to access the interrupt vector (interrupt handler routine address) from memory. The processor then continues execution of the interrupt handler routine.

VPA will always be asserted automatically during an interrupt acknowledge bus cycle.

2.4.4 The Multibus Interface

The CPU conforms to the IEEE 796 proposed Multibus standard, with some minor differences. Data bit 0 of the Multibus is data bit 0 of the 68000, but the 68000 defines the lower byte of the data bus as odd and the upper byte of the data bus as even. The Multibus reverses this organization. A byte swap buffer is implemented between the microprocessor and the Multibus. The byte swap buffer interface ensures that odd and even single-byte transfers take place on the low order data lines. Word transfers use the full set of sixteen data lines, under the control of the BHEN signal, i.e., A0.

NOTE

When writing a byte to a peripheral device, the least significant address bit must be complemented or inverted by the software because of the MC68000 to Multibus incompatibility.

2.4.5 Address Bus

The microprocessor uses a 24-bit address bus to provide 16 megabytes of addressing. Address bit zero (0) is required on the Multibus and is generated by the CPU board as a function of the upper and lower data strobes of the Motorola 68000 (see 68000 manual for additional information). The lower 11 address lines are directly buffered onto the bus connector. Normally, the upper 12 lines are routed directly from the 68000 to the bus connector. The upper 12 lines may also be translated by memory mapping registers into a physical location when the 68000 is in user state or when the the processor is in the supervisory state with the memory mapping flag set and the address less than $200000 (HEX address).

The address bus is asserted LOW at the bus connector and is the logical inversion of the address bus at the microprocessor. The address lines from the CPU board are put into a high impedance state when another master controls the

bus.

2.4.6 Data Bus

The MC68000 uses a 16-bit data bus to transfer processor instructions and data. The data bus is buffered at the bus connector. Both byte and word transfer operations are supported.

The data lines are active low on the Multibus. When another master has control of the bus, the CPU board bus buffers are placed in a high impedance state.

2.4.7 Asynchronous Bus Control

The control signals comprise four command lines and one response line.

1. Memory Read Control (MRDC/)
2. Memory Write Control (MWTC/)
3. I/O Read Control (IORC/)
4. I/O Write Control (IOWC/)
5. Transfer Acknowledge (XACK/)

Together, these lines coordinate data transfer on an asynchronous bus.

2.4.7.1 Command Lines (MRDC/, MWTC/, IOWC/, And IORC/) -

The four command lines are communication links between the bus masters and bus slaves. An active command line tells the slave that the address lines are carrying a valid address, and that the slave should perform the specified operation.

The CPU board defines a portion of the 68000's address space as I/O space. When accesses are made in the address range

$F00000 to $F0FFFF, the CPU board generates an I/O read or write command on the bus (depending on the state of the R/W line from the 68000) to initiate a data transfer. During an I/O transfer, only the lower 16 address bits are defined for a total of 64 K bytes of I/O space.

2.4.7.2 Transfer Acknowledge Line (XACK/) -

This line is the slave's acknowledgement of the master's command. XACK/ tells the master that the slave has placed data on or accepted data from the data bus.

2.4.8 System Control

A timeout of the bus error clock tells the processor that a specified time has elapsed without a slave process responding to the assertion of a data transfer command. The clock provides a nonadjustable bus error timeout value of 20.0 uS.

Figure 2.2 is a record of a bus error sequence.

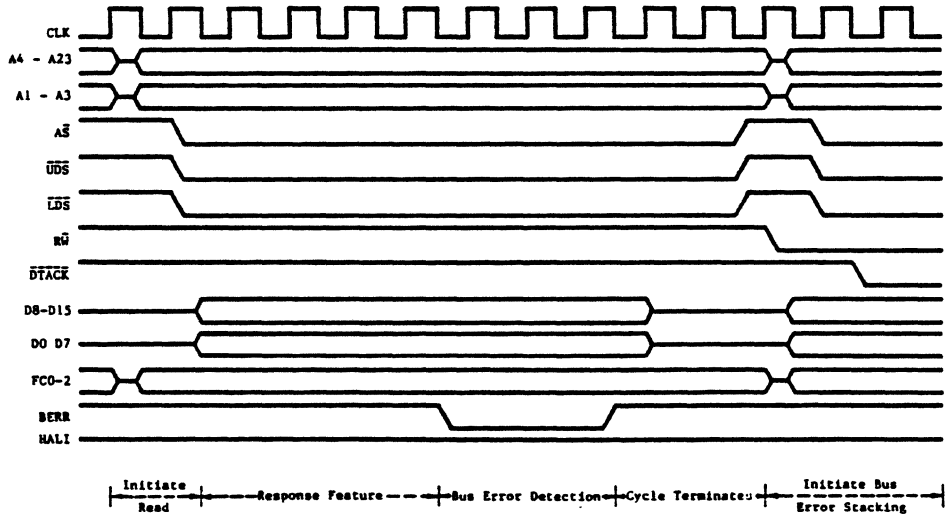


Figure 2.2 Bus Error Sequence

2.5 ON-BOARD FUNCTIONS

2.5.1 Memory Mapping Registers

The memory mapping registers use 1,024 bytes of high speed static RAM. When mapping is enabled, address lines A12-A20 are the register select inputs to the mapping register RAM, and the data lines out of the RAM become the new address lines A12-A23. Mapping is automatically enabled when the processor is in the user mode and the processor address is below $200000, or in supervisor mode and the map flag register is set (shown in figure 2.3).

2.5.2 Memory Mapping

If mapping is enabled, the original upper three address bits from the processor (A21-23) must be zeroes. (Users are confined to the lowest two megabytes of address space.) The next nine address bits (A12-20) are used to access one of 512 locations of the memory mapping registers. These registers are 16 bits wide, containing 12 bits of new address and three bits of access control information. One bit is not used. Each register location within the memory map represents a four-Kbyte segment of logical space, for a total of two megabytes of user logical memory space.

When a memory mapping register is accessed, the 12 bits of new address information replace the original upper 12 address bits from the processor. This scheme allows the system to map any of the user's 512 four-Kbyte segments into any of the system's physical address spaces, including I/O space.

The three bits of access control information interact with the function codes (representing the state of the processor) and the memory mapping flag to check for access errors described in 2.5.4. Figure 2.3 shows the memory mapping register format.

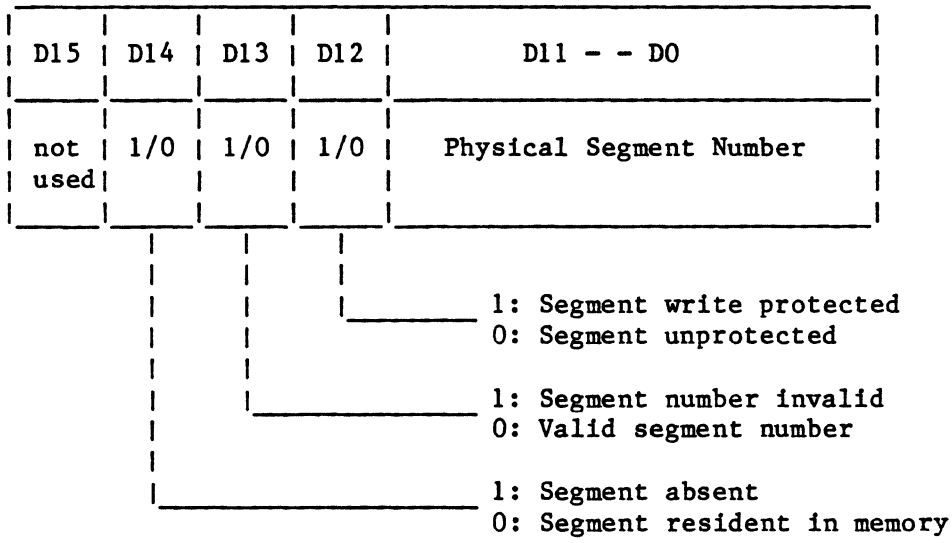


Figure 2.3 Memory Mapping Register Format

The memory mapping registers are accessible as read/write memory, beginning at location $EFF800.

2.5.3 Error Control

The CPU board error control circuitry monitors the operation of the board. When an error occurs, the error type is latched, and a level seven interrupt is generated that allows a routine to handle the error.

2.5.4 Address Errors

An illegal condition on the address bus causes an address error. All addresses are inherently legal to the 68000 (except for a word access on a byte boundary). Therefore, an access error can occur only when mapping is active. Four address errors are associated with use of the memory mapping:

1. **access violation** occurs when accessing outside of user logical space, defined as 000000 - 1FFFFF (2 megabytes).

2. **write violation** occurs when writing to a logical segment that is write protected.

3. **invalid segment** occurs when accessing a nonallocated logical segment.

4. **nonresident segment** occurs when accessing a nonresident segment.

Conditions 2, 3, and 4 result directly from the access control information stored in the upper bits in each location of the memory mapping register.

2.5.5 Memory Mapping Flag And Bus Arbitration Lock Flag

If the processor is in supervisor state and the processor address bus is carrying an address below 2 megabytes, the value of the memory y mapping flag determines whether the address is mapped. (The address is mapped if the value of the flag is a one.) To set the memory mapping flag write 80 to location $EFFC01, which is an even byte address. Reset the flag by writing 00 to the same location. The value of the flag may not be read and is automatically reset to zero when the system is reset.

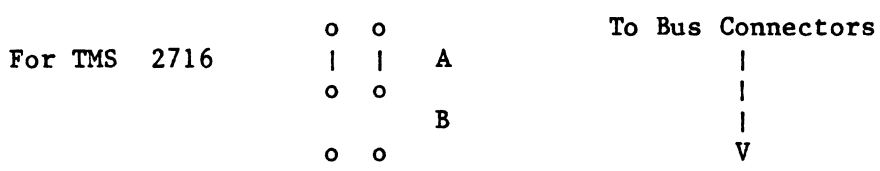
Also at address EFFC01 is the flag to lock bus arbitration. By writing a $81 to EFFC01, all external bus requests are masked. The flag is cleared by writing a $01 to $EFFC01.

2.5.6 On Board Memory

Sockets for 8 UV-EPROMs (2K x 8) allow for 16 Kbytes of ROM. The CPU board supports the 2716. The select jumpers are located between C7 and C8 on the CPU board. The jumper area consists of six pads arranged in three rows of two columns. See appendix A for the various jumper configurations.

2.6 APPENDIX A

JUMPER CONFIGURATION



2.7 APPENDIX B

PAL EQUATIONS

Uncommitted logic space in the chips is configured as per the following equations:

PAL10L8 PAL DESIGN SPECIFICATION
P/N 318-021-001
CONTROL SIGNAL GENERATOR - POSTITON G2, S150 MMU CPU BOARD

```
SYSIO PA8 PA9 PA10 /PA0 /MWTC /MRDC /BHEN /CBUSY GND
SEL UHALF ERRSEL FLGSEL IOXACK LBYTE SWBYTE HBYTE LHALF VCC


LHALF   = /PA0
          +BHEN

UHALF   = PA0
          +BHEN

IOXACK  = SYSIO*MRDC*CBUSY
          +SYSIO*MWTC*CBUSY

ERRSEL  = SYSIO*MRDC*PA10*/PA9*PA8*/PA0

FLGSEL  = SYSIO*MWTC*CBUSY*PA10*/PA9*/PA8*/PA0
```

DESCRIPTION:

This chip generates the byte control lines for the on-board I/O, which includes the ROM, the flag register, the error register, and the map registers.

PAL12H6 PAL DESIGN SPECIFICATION
P/N 318-022-001
ERROR AND VAS GENERATOR - POSITION B4, S150 MMU CPU BOARD

```
NRS NVS SWP /AS RW FC2 /MAPPED MAPFLG PIN9 GND
PIN11 PIN12 PIN13 AV WV SNV SNR VADD USP VCC


VADD = MAPPED*/NRS*/NVS*RW*AS
       +MAPPED*/NRS*/NVS*/SWP*AS
       +FC2*/MAPFLG*AS
       +FC2*AS*/USP

AV   = /FC2*/USP

SNV  = NVS*MAPPED

WV   = SWP*/RW*MAPPED

SNR  = NRS*MAPPED
```

DESCRIPTION:

This PAL generates the valid address strobe and the error flags.

2.8 APPENDIX C

EPROM CONFIGURATION FIRMWARE

A 32 x 8 PROM on the CPU board is used for ROM address decoding. Its contents depend on the size of ROM for which the CPU has been configured. The following tables show the contents of the PROM for three typical EPROM configurations.

Table 2-B
2K X 8 EPROMS (standard)

Address	Data	Address	Data
0	0E	10	0F
1	0D	11	0F
2	0B	12	0F
3	07	13	0F
4	0F	14	0F
5	0F	15	0F
6	0F	16	0F
7	0F	17	0F
8	0F	18	0F
9	0F	19	0F
A	0F	1A	0F
B	0F	1B	0F
C	0F	1C	0F
D	0F	1D	0F
E	0F	1E	0F
F	0F	1F	0F

CHAPTER 3

INPUT/OUTPUT (I/O) BOARD

3.1 INTRODUCTION

The System 150 I/O board, part number 810-104-001, handles all system input and output, and fully supports the IEEE-488 data bus option.

This module deals with the physical and logical aspects of the I/O board and discusses in particular the I/O board configuration, the serial interface, the interval timer, and the parallel port.

3.2 INSTALLATION AND OPERATION

The I/O board malfunctions if used with the incorrect connector panel module. The correct correspondence is shown here by part numbers:

I/O Board	Connector Panel Module
810-104-00X	810-086-001

If you ever need to install the 810-104-001 board, remove the bus arbitration jumper that corresponds to the I/O board slot on the motherboard. Otherwise, the system will fail.

Configure and correctly wire your system to printers, terminals, and modems according to the I/O board in use. The 810-104 board requires the pin-out of the RS-232 port to be configured as DTE.

3.3 I/O BOARD CONFIGURATION

The I/O board comprises eight main areas of circuitry:

- o the board address select jumpers (3.3.1)
- o the serial interface (3.3.2)
- o the real-time calendar clock (3.3.3)
- o the interval timers (3.3.4)
- o the general purpose parallel port (3.3.5)
- o the parallel port direction and LED register (3.3.6)
- o the select/configuration switches (3.3.7)

3.3.1 The Board Address Select Jumpers

The I/O board registers are located in Multibus I/O space. The board compares address bits 8/ through F/ on the bus with the board select jumpers. Changing the select jumpers allows multiple boards to be placed in the same system. The addresses described in this manual are shown in the following format:

$XX YY

where:

XX = base board address in the range $00 YY to $FF YY,

and

YY = specific device register on the board.

3.3.2 The Serial Interface - Signetics 2661

The Signetics 2661 EPCI (Enhanced Programmable Communications Interface) is used to allow seven RS-232 C serial interfaces with full handshaking to be implemented on the I/O board. The PCIs generate the baud rate, which is software selectable from 110 baud to 19.2K baud.

All seven PCIs (0-6) are selected at the even address locations (see figure 3.1, I/O Memory Map) between XX00 and XX36, with the first PCIs using the first four locations, the second using the next four, and so on.

All PCIs use INT5/ on the Multibus, which is interrupt level 2 on the 68000 chip. The PCIs are tied to the lower byte of the data bus. Select internal registers on address lines BADR1-BADR2. PCIs 1-5 communicate externally through the I/O connector panel. PCI 0 connects to the internal terminal.

PCI six is configured to be used with a modem and for this reason is brought off the board on its own connector, which is labelled P4 on the circuit board. All handshaking lines, including data carrier detect (DCD) input signals, are available.

A register at I/O location $(XX)D4 controls the port 6 PCI transmitter interrupt to be enabled or disabled. It is enabled by writing a $_ to I/O location $XXD4 and disabled by writing a $_ to I/O $XXD4.

XX = base board I/O address

XX00-06	Serial Port 0
XX08-0E	Serial Port 1
XX10-16	Serial Port 2
XX18-1E	Serial Port 3
XX20-26	Serial Port 4
XX28-2E	Serial Port 5
XX30-36	Serial Port 6
XX40-5E	Parallel Port & Interval Timers
XX60-7E	Calendar Clock
XXD0	LEDs/Parallel Port Direction
XXD2	Select/Configuration Switches

Figure 3.1 System 150 I/O Memory Map

NOTE

The addresses specified in this document are hexadecimal Multibus I/O addresses.

3.3.3 The Real-Time Calendar Clock--National Semiconductor 58174

The CPU can set and read the real-time clock (RTC), a calendar clock with registers all in binary coded decimal format. The RTC is equipped with an array of registers, so the CPU can increment the RTC by one interval every fixed period. Twelve fixed periods are possible:

1/10 of a second	10 minutes	10 days
seconds	hours	day of week
10 seconds	10 hours	months
minutes	days	10 months

A battery backup circuit provides power to the RTC for one to two years. Select the RTC at even byte addresses from F00060-F0007E, and select internal registers on address lines BADR1 - BADR4 (see figure 3.1, I/O Memory Map). The RTC data bus is four bits wide and is tied to the four least significant bits of the low byte of the system data bus.

3.3.4 The Interval Timers - Synertek 6522A

Two interval timers are included on the I/O board. The MCS uses these timers for precise timing of external events, i.e., for real-time references. Each timer is programmable to operate in several modes and can interrupt the microprocessor when software-specified conditions occur. The timers are contained in the SY6522 timer/PIA IC and are tied to the lower byte of the data bus. Select the internal register on address lines BADR1-BADR4.

The Synertek 6522A IC is selected at even byte I/O addresses XX40-XX5E and uses INT6/ on the Multibus, which is interrupt level 1 on the 68000 processor.

3.3.5 The Parallel Port - Synertek 6522

The I/O board has a 16-bit, general purpose, bidirectional parallel port, with four handshaking lines. The parallel port is made up of two bidirectional, eight-bit ports, port A and port B, supplied by the SY6522 Timer/PIA IC. An input and an output handshaking line are associated with each eight-bit data port. Both eight-bit ports are externally buffered with bidirectional buffers.

On power up both eight-bit ports are configured as inputs. To configure the ports as either inputs or outputs, the correct data must be written to the SY6522 IC and also to a bit-addressable latch that controls the external buffers. The addressable latch is described in detail in the parallel port direction and LED register description in 3.3.6.

3.3.6 The Parallel Port Direction And LED Register -74LS259

The parallel port buffers and the six LEDs are controlled by an addressable latch. Select the latch at location XXD0. The function of each output is described below.

1. BDIR - Direction control for port B. A one at this output causes the buffer on port B to become an output, while a zero causes the buffer to become an input.

2. ADIR - Direction control for port A. A one at this output causes the buffer on port A to become an output, while a zero causes the buffer to become an input.

3. LED1-LED6 - On/off control for the SIX LEDs. A low at one of these outputs causes the corresponding LED to be turned on, while a one causes the corresponding LED to be turned off.

4. All outputs are cleared to zeros on power up and on reset.

*Decoded address from bit address latch

0	Port A	Buffer Input
1	Port B	Buffer Input
2	CR1 ON	
3	CR2 ON	
4	CR3 ON	
5	CR4 ON	
6	CR5 ON	
7	CR6 ON	
8	Port A	Buffer Output
9	Port B	Buffer Output
A	CR1 OFF	
B	CR2 OFF	
C	CR3 OFF	
D	CR4 OFF	
E	CR5 OFF	
F	CR6 OFF	

Figure 3.2 Explanation of Addressable Latch Operation

An addressable latch is a write-only latch on which one bit is written at a time. This is accomplished by using three bits of the input data as an address to select which of the latch outputs is to be written to. Another bit of the input data is used as the data to be written to the addressed output. The input data byte is organized as shown below:

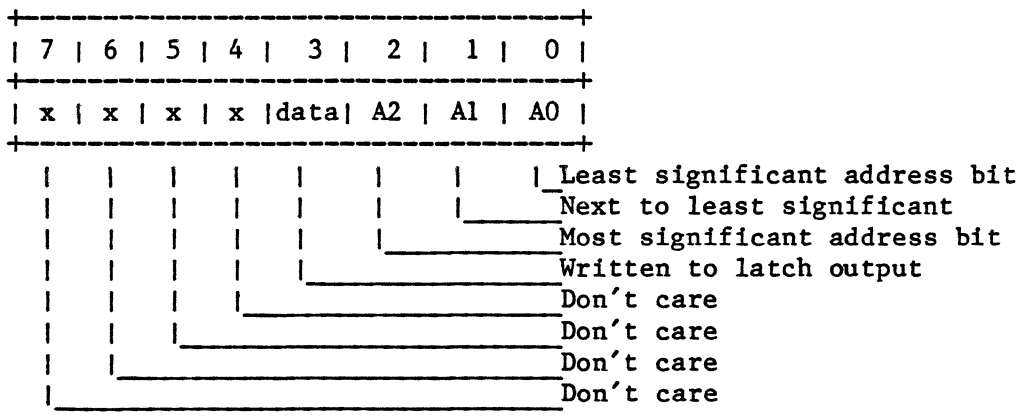


Figure 3.3 Input Data Byte Organization

3.3.7 The Select/Configuration Switches

The eight read-only DIP switches can be read by the system and are selected at address XXD2.

CHAPTER 4

MEMORY BOARD

4.1 INTRODUCTION

This module describes the memory board signals, the electrical characteristics of the board, and the board configuration, including the location and the use of address switches.

4.2 DEFINITION AND FEATURES

The memory board is a dynamic RAM memory module organized as 256K words by 16 bits and uses the 64K DRAM chip. The board's circuitry incorporates the following:

- o **ECC:** Error checking and correction (ECC)
- o **Status registers:** A control status register to select options, and an error status register to make error conditions available to software
- o **Decoding:**
  - 24 lines of address capability
    - Four extended address lines to select any of 16 1-megabyte pages
    - Data access in either word or byte mode

4.3 MEMORY MODULE SIGNALS

Signals common to the Multibus are defined in the Intel Multibus Specification Manual 9800683 or in the system bus standard for the IEEE 796 bus.

4.4 ELECTRICAL CHARACTERISTICS

4.4.1 Error Detection And Correction (EDAC)

EDAC employs an additional six bits of check bit data to detect and correct single bit errors and to detect double bit/gross errors. When enabled, EDAC completes all operations having a single bit READ error by attempting to write corrected data back to memory.

4.4.2 Status Registers (CSR And ESR)

The memory module has a control status register (CSR) and an error status register (ESR).

Address Selection: Both registers are accessed at an I/O address. CSR is selected with ADR0 = 1 (an electrical high); ESR is selected with ADR0 = 0. The I/O port base address is designated with eleven on board switches. Eight switches are compared with ADR8/ and ADR4/, and the three start address switches are compared with ADR1/ through ADR3/. Selecting addresses this way permits the eight more significant bits of the I/O port address to be identical to other memory boards in the system and to mirror the I/O port address selection made based on the position of the memory board within the address space.

See table 4-G for the switch settings.

CSR Operation: The CSR is used on a programmable I/O chip for read and write operations. It controls operation and stores information about errors. To use this I/O chip, the microprocessor must configure the chip by executing two operations for every port:

1. Load the CSR

2. Designate lines as inputs or outputs

CSR Formats: Figure 4.1 shows the format for reading data from the CSR:

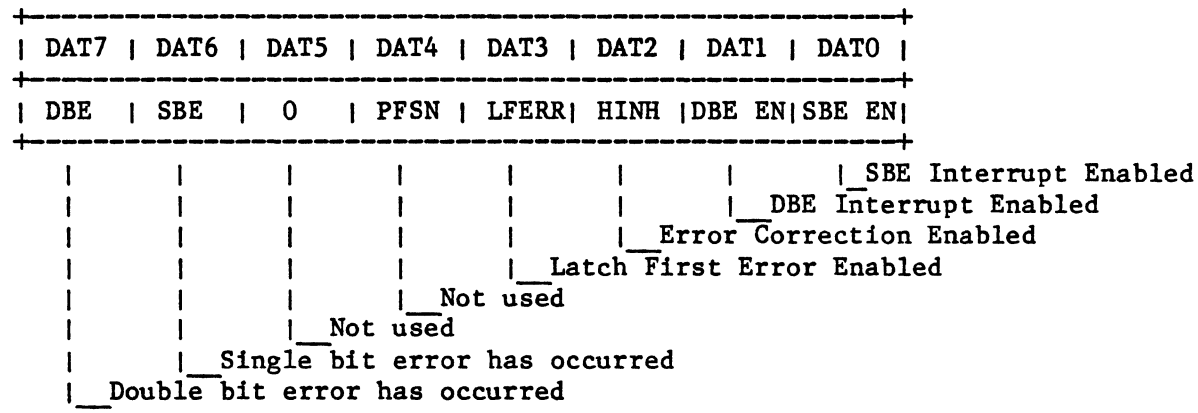


Figure 4.1 CSR Read Format

Figure 4.2 shows the format for writing data into the CSR:

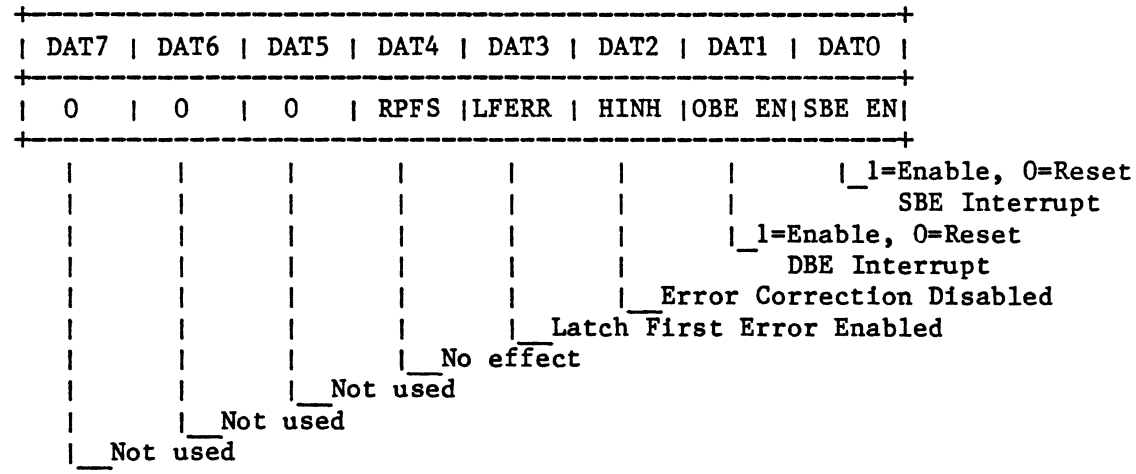


Figure 4.2 CSR Write Format

CSR Flag Control Bits: The system supports six CSR Flag Control Bits, described below.

Double Bit Error Flag (DBE)

DBE indicates that two bits in the same word failed or that a gross error has been detected. DBE is set when a double bit error occurs, then reset when the DBE EN control bit is taken to zero. Setting DBE EN

back to a logic one enables detection of the next double bit error.

Single Bit Error Flag (SBE)

SBE, a read only signal, indicates that a single bit error has been detected. SBE is set when the error is detected, then reset when the SBE EN control bit is taken to logic zero. Setting the SBE EN control bit back to a logic one enables the detection of the next single bit error.

Power Fail Sense F/F Flag (PFSN L)

This function is not used on WICAT equipment.

Latch First Error (LFERR)

LFERR allows you to select whether ERR data are updated each time a SBE/DBE is detected, or only once on the first error detected. Writing a logic one to the LFERR control bit enables LFERR so that the next error can be stored. When you set LFERR to zero, the last error always updates the ESR. LFERR can be read or written.

Error Correction Disabled (HINH)

When equal to logic one, HINH disables error detection, error correction, and the write function to the checkbits. Use HINH only for testing. HINH can be read or written, and the board cannot be initialized when this bit is set.

Enable DBE Interrupt (DBE EN)

When DBE EN equals logic one, interrupts on DBEs are possible. Taking this DBE EN control bit to a zero resets any current DBE flag bit. Leaving DBE EN at a zero disables DBE interrupts. DBE EN can be read or written.

Enable SBE Interrupt (SBE EN)

When SBE EN equals logic one, interrupts on SBEs are

possible. Taking the SBE EN control bit to a zero resets any current SBE. Leaving SBE EN at zero disables SBE interrupts. SBE EN can be read or written.

4.4.2.1 Error Status Register (ESR) -

When a single bit error occurs and the conditions for FE/LE have been satisfied, the ESR stores the error information. The ESR is a read-only register. Clear the ESR either by writing to it or by resetting the system.

Figure 4.3 shows the format for reading data from the ESR.

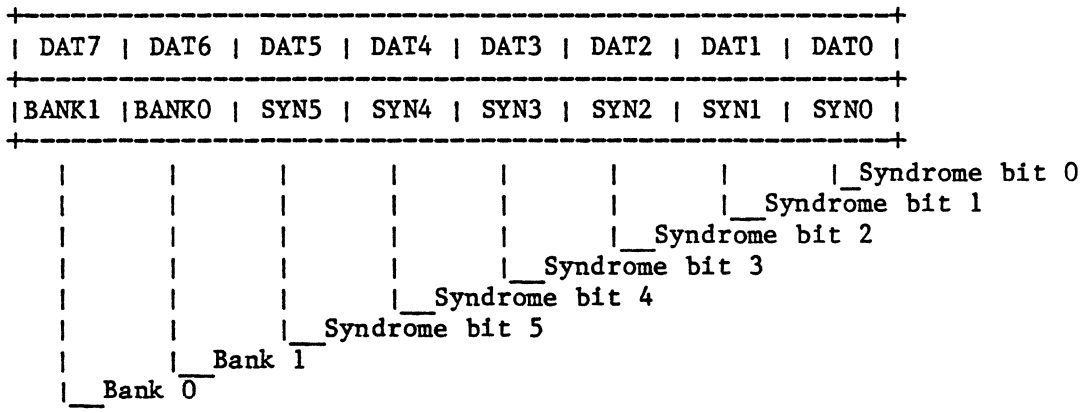


Figure 4.3 ESR Read Format

4.4.2.2 ESR Signal Definitions -

Bank:

Bank 1 and Bank 0 define the physical row of DRAMS (BANK) in which an SBE occurs. Table 4-A shows the relationship between the bank address s and the reference designator for memory devices within that row.

Table 4-A
Bank Signals Reference Table

Bank 1	Bank 0	Memory	Reference	Chip Numbers
0	0	U000	through	u021
0	1	U100	through	u121
1	0	U200	through	u221
1	1	U300	through	u321

Syndrome bit 5 through bit 0 (SYN5 - SYN0)

Bits 5 through 0 define the modified Hamming code generated by the SN74LS630 EDAC device when a SBE occurs. Table 4-B shows the relationship between the syndrome code and bit location within a given bank.

Table 4-B
Syndrome Code

Syndrome Code (Bit 5 4 3 2 1 0)	Data Bit	Check Bit	Bit Location X=0,1,2,3 (Bank)
1 1 0 1 0 0	0		X00
1 1 0 0 1 0	1		X01
1 1 0 0 0 1	2		X02
1 0 1 1 0 0	3		X03
1 0 1 0 1 0	4		X04
1 0 1 0 0 1	5		X05
1 0 0 1 0 1	6		X06
1 0 0 0 1 1	7		X07
0 1 1 1 0 0	8		X08
0 1 1 0 1 0	9		X09
0 1 0 1 1 0	10		X10
0 1 0 1 0 1	11		X11
0 1 0 0 1 1	12		X12
0 0 1 1 1 0	13		X13
0 0 1 1 0 1	14		X14
0 0 1 0 1 1	15		X15
1 1 1 1 1 0		0	X16
1 1 1 1 0 1		1	X17
1 1 1 0 1 1		2	X18
1 1 0 1 1 1		3	X19
1 0 1 1 1 1		4	X20
0 1 1 1 1 1		5	X21
0 0 0 0 1 1	Gross Error Condition		
1 1 1 1 0 0	Gross Error Condition		

Note: 1 = lamp on

The syndrome codes for DBEs are mutually exclusive of any SBE codes. If you clear the ESR by writing to it, then by reading the ESR and comparing it to zero you can check (poll) the board for errors.

4.4.2.3 Error Status LEDs -

The ten light emitting diodes (LEDs) near the address switches display the error status: the left-most light indicates a DBE; the next light indicates a SBE. Interpretation of the next eight lights is the same as the ESR. Writing to

the ESR clears the ten LEDs. An active input on the INIT/line clears the LEDs and the ESR.

4.4.2.4 Error Detection And Correction (EDAC) -

EDAC Enabled

EDAC is enabled by an active INIT/ during power up. When the CSR bit HINH is low, the EDAC device, SN74LS630, is enabled. EDAC generates checkwords, syndrome bits, and error flags (DBE and SBE), and corrects data words. Corrected data from the EDAC are stored in a data latch so that during a read operation the corrected data are available on the bus, can replace error data (write back on error), and are made available for byte write operations.

EDAC Disabled

When the CSR bit HINH is high, the EDAC device is held in an input mode, and the error flags (DBE and SBE) are held reset. Read data are not corrected, and the write operation to the checkbits is inhibited. Thus, write data generate no checkbits, and writing a word to a location with HINH active modifies the 16 data bits but leaves the six check bits unchanged from the last write operation to where HINH was inactive.

4.4.2.5 Interrupt Options -

Hardware Programming

The interrupt request lines generate nonbus vectored interrupts to the bus master (68000 interrupt level 7).

Interrupts are always wired so that a SBE or a DBE causes INT0/ to be asserted.

Software Programming

With SBE EN and DBE EN bits in the CSR, you can

enable or disable SBE interrupt and DBE interrupt respectively. To clear the current interrupt, disable the interrupt, then reenable so that the interrupt will operate on future errors.

4.5 CONFIGURING THE MEMORY BOARD

On powerup, the CSR is initialized so that error detection and correction are enabled, the ESR captures the last error, and DBE and SBE interrupts are disabled. For initial checkout, you need set only the four address switches.

Turn the board so that the switches are facing you. From left to right the switches facing you are the starting address, ending address, and I/O port base address switches. The megabyte page address switch is in the upper right-hand corner.

NOTE

The MSB is on the left of each switch. ON equals logical zero. OFF equals logical one.

4.5.1 Starting Address

The starting address is WX000 hexadecimal. WX represents two hex digits (eight bits) that correspond directly to the eight bits of the starting address switch, S2. Consider the example in table 4-C.

Table 4-C
Starting Address Settings

Desired Starting Address (hex)	Switch S2								
	WX	-1	-2	-3	-4	-5	-6	-7	-8
00000	00	0	0	0	0	0	0	0	0
40000	40	0	1	0	0	0	0	1	0
80000	80	1	0	0	0	0	0	0	0
C0000	C0	1	1	0	0	0	0	0	0

Physical memory begins at location $200000 in a system using the 810-077-001 CPU board (multi-user mapped) and

must be contiguous. The physical address of fully populated (512K) board 0 starts at $200000, of board 1 at $280000, and of board 2 at $300000.

Starting switch settings for a fully populated board are given in table 4-G.

4.5.2 Ending Address

The ending address is YZFFF hexadecimal. YZ represents two hex digits (eight bits) that correspond directly to the eight bits of the ending address switch, S3. Consider the example in table 4-D.

Table 4-D
Ending Address Settings

Desired Ending Address (hex)	Switch S3 : YZ : -1	-2	-3	-4	-5	-6	-7	-8
3FFFF	: 3F : 0	0	1	1	1	1	1	1
7FFFF	: 7F : 0	1	1	1	1	1	1	1
BFFFF	: BF : 1	0	1	1	1	1	1	1
FFFFF	: FF : 1	1	1	1	1	1	1	1

The physical address on fully populated 512K boards ends at 27FFFF on board 0, at 2FFFFF on board 1, and at 37FFFF on board 2.

Ending address switch settings for fully populated boards are given in table 4-G.

4.5.3 Enabling Extended Address Lines

To enable the four extended address lines, locate the 5 position DIP switch, S5, in the upper right-hand corner when the board is component side up and the gold fingers are away from you. The leftmost switch, S5-1, enables or disables the four additional address lines as shown here:

Switch	Position	Function
S5-1	0	enables ADR14/ through ADR17/
S5-1	1	disables ADR14/ through ADR17/

When used in a mapped system (S150, S155, or S160), S5-1 is always set to logical zero.

4.5.4 Setting Address For Extended Lines

The board's location is one of 16 possible one-megabyte pages. From the remaining four positions of the five-position DIP switch S5, select one page as shown in table 4-E.

Table 4-E
Switch Setting for Each Page

One-megabyte page	Switches S5-2	S5-3	5-4	S5-5
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

4.5.5 I/O Port Address

The I/O port base address is OPQR hexadecimal. PQ represents two hex digits (eight bits) in direct correspondence to the eight bits of the I/O port base address switch, S4, and R is a hex digit comprising the three more significant bits of W (from the module starting address) plus the state of ADR0/ from the bus.

Table 4-F shows an example of how to set the I/O port base address.

Table 4-F
Address Settings for I/O Port

PQ Switch S4 I/O Port Setting								R Switch S2 Start Address			Logical ADR0	(PQR) I/O Port Addr	REG
-1	-2	-3	-4	-5	-6	-7	-8	-1	-2	-3			
1	1	1	1	0	0	1	0	0	0	0	L	0F20	CSR
1	1	1	1	0	0	1	0	0	0	0	H	0F21	ESR
1	1	1	1	0	0	1	1	0	0	0	L	0F30	CSR
1	1	1	1	0	0	1	1	0	0	0	H	0F31	ESR
1	1	1	1	1	1	1	1	0	0	0	L	0FF0	CSR
1	1	1	1	1	1	1	1	0	0	0	H	0FF1	ESR

4.5.6 Switch Settings

Memory board DIP switch settings for a fully populated board (512K) are shown in table 4-G.

Table 4-G
Memory Board Switch Settings (512 K boards)

Board	Address	Switch Settings (0=ON, 1=OFF) 1	2	3	4	5	6	7	8
0	Starting Address	0	0	0	0	0	0	0	0
	Ending Address	0	1	1	1	1	1	1	1
	I/O Port base	1	1	1	1	0	0	1	0
	S-5 (Page Address)	0	0	0	1	0			
1	Starting Address	1	0	0	0	0	0	0	0
	Ending Address	1	1	1	1	1	1	1	1
	I/O Port base	1	1	1	1	0	0	1	0
	S-5 (Page Address)	0	0	0	1	0			
2	Starting Address	0	0	0	0	0	0	0	0
	Ending Address	0	1	1	1	1	1	1	1
	I/O Port base	1	1	1	1	0	0	1	1
	S-5 (Page Address)	0	0	0	1	1			

Table 4-H
Switch Settings for 256K boards

Board	Address	Switch Settings (0=ON, 1=OFF) 1	2	3	4	5	6	7	8
0	Starting Address	0	0	0	0	0	0	0	0
	End	0	0	1	1	1	1	1	1
	I/O Port	1	1	1	1	0	0	1	0
	S-5C Page	0	0	0	1	0			
1	Starting	0	1	0	0	0	0	0	0
	Ending	0	1	1	1	1	1	1	1
	I/O Port	1	1	1	1	0	0	1	0
	S-5	0	0	0	1	0			
2	Start	1	0	0	0	0	0	0	0
	End	1	0	1	1	1	1	1	1
	I/O Port	1	1	1	1	0	0	1	0
	S-5	0	0	0	1	0			

CHAPTER 5

THE WINCHESTER FLOPPY CONTROLLER BOARD (WFC)

5.1 INTRODUCTION

This module describes the Winchester Floppy Controller Board (WFC), including specifications, configuration, theory of operation, programming, diagnostics, and adjustments.

5.2 GENERAL DESCRIPTION

The WFC can accommodate up to four 5 1/4-inch Winchester disk drives and four 5 1/4-inch floppy disk drives on the S150 host computer. The WFC board includes all necessary buffers and receivers/drivers for direct connection.

The WFC is compatible with the IEEE 796 Bus. All parameters and commands are given through the registers mapped into the I/O space of the system bus. Data transfers use direct memory access (DMA).

5.3 FEATURES

The major features of the WFC board include:

- Built-in data separator
- Built-in write precompensation logic
- Data rates up to 5 Mbits/sec
- Control for up to 8 R/W heads per drive
- 1024 cylinder addressing range

THE WINCHESTER FLOPPY CONTROLLER BOARD (WFC)
FEATURES

- o 256 sector addressing range
- o CRC generation/verification
- o Automatic formatting
- o 128, 256, 512, or 1024 bytes per sector (ROM selectable)
- o Unlimited sector interleave capability
- o Implied seek on all commands
- o Automatic retries on all errors
- o Automatic restore and seek on seek errors
- o IEEE 796 bus compatible interface

5.4 SPECIFICATIONS

Table 5-A is an item by item display of the System 150 specifications.

Table 5-A
System 150 Specifications

ITEM	SPECIFICATION
Winchester Interface:	ST-506 compatible
Floppy Interface:	Shugart compatible
Encoding Method:	Modified Frequency Modulation (MFM)
Cylinders (max):	
per Winchester:	1024
per Floppy:	256
Sectors per Track (max):	256
Heads per Drive (max):	8
Winchester:	8
Floppy:	2
Drive Selects:	8 (4 Winchester, 4 Floppy)
Step Rate:	
Winchester:	10 uS to 7.5 mS (0.5 mS increments)
Floppy:	6, 12, 20, or 30 mS
Disk Data Rate:	
Winchester:	5 Mbits per second
Floppy:	250 Kbits per second
Write Precompensation Time	
Winchester:	10 nanoseconds
Floppy:	300 nanoseconds
Sectoring:	Soft
Drive Cable Length:	5 ft. (max)
DMA Address:	24 bits
DMA Data:	16 bits
Sector Length:	128, 256, 512, 1024
Voltages	+ 5 volts + 12 volts - 12 volts
Board address	Decodes lower 16 address lines Base address selectable on 32-byte boundary

5.5 THEORY OF OPERATION

5.5.1 Host Interface

All data transfers between the host and the WFC take place over the IEEE 796 bus using DMA transfers.

The WFC produces interrupt requests (INTRQ) to signal the end of all disk functions. The INTRQ originates on the MFM generator (U56) as an auxiliary function of the chip. The WFC sets INTRQ using INTCLK, which is produced by U86. Interrupts are cleared by BOARD SEL and A0, 1 when the host reads the status register, issues a command, or accesses the sector number register. During power on or reset, INTRQ is reset.

5.5.2 Task Files

To perform a disk function the WFC employs a set of registers called the task file. You must load the task file before you issue a command. Load these registers with parameters such as sector number, cylinder number, etc. Individual registers are selected via A0-2. Table 5-B shows the available registers.

Table 5-B
Task File Registers

ADDRESSED DATA BITS						DISK FUNCTIONS	
				IEEE 796	68000		
A4	A3	A2	A1	A0	A0	READ	WRITE
0	0	0	0	1	0	Error Register	Write Precomp
0	0	0	0	0	1	Switches	Switches
0	0	0	1	1	0	Sector Number	Sector Number
0	0	0	1	0	1	Sector Count	Sector Count
0	0	1	0	1	0	Cylinder High	Cylinder High
0	0	1	0	0	1	Cylinder Low	Cylinder Low
0	0	1	1	1	0	Status Register	Command Register
0	0	1	1	0	1	Size/Drive/Head	Size/Drive/Head
0	0	0	0	1	0	Reserved	Reserved
0	0	0	0	0	1	Reserved	Reserved
0	0	0	1	1	0	Reserved	Reserved
0	0	0	1	0	1	Reserved	Reserved
0	0	1	0	1	0	DMA Addr 9-16	DMA Addr 9-16
0	0	1	0	0	1	DMA Addr 1-8	DMA Addr 1-8
0	0	1	1	1	0	Reserved	Reserved
0	0	1	1	0	1	DMA Addr 17-23	DMA Addr 17-23
1	X	X	X	X	X	Reserved	Reserved

5.5.3 Command Register (Write Only)

Load all commands into the command register after you have set the task registers. Writing to this register causes the INTRQ line to be reset.

5.5.4 Size/Drive/Head Register

This register contains the sector size, drive select, and head select bits. The register is organized as shown in table 5-C.

Table 5-C
Size/Drive/Head Register

Bit	7	6 5	4 3	2 1 0
Function	0	sector size	drive select	head select

Size

bit 6	bit 5	sector size
0	0	256 bytes
0	1	512 bytes
1	1	128 bytes
1	0	1024 bytes

Drive

bit 4	bit 3	drive selected
0	0	drive select 0
0	1	drive select 1
1	0	drive select 2
1	1	drive select 3

Head

bit 2	bit 1	bit 0	head selected
0	0	0	head 0
0	0	1	head 1
0	1	0	head 2
0	1	1	head 3
1	0	0	head 4
1	0	1	head 5
1	1	0	head 6
1	1	1	head 7

5.5.5 Cylinder Number

Two registers form the cylinder number where the head is to be positioned on a seek, read, or write command.

Table 5-D
Cylinder Register Number

	Cylinder High	Cylinder Low
Register bits	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Cylinder bits	- - - - - - 9 8	7 6 5 4 3 2 1 0

A separate set of cylinder register values is maintained internally for each drive. The two least significant bits of the cylinder high register form the most significant bits of the cylinder number.

5.5.6 Sector Number

Load this register with the desired sector number before you issue a read or write command. The sector number register may be read or written to by the host.

5.5.7 Sector Count

Load this register with the number of sectors to be formatted during a format command. This register is decremented to zero during the format command and must be reloaded for each format operation.

5.5.8 Write Precompensation Register

The write precompensation register is needed only for the Winchester interface because write precompensation always starts at track 44 for floppy drives. The register holds the cylinder number where the RWC line is asserted and write precompensation logic is enabled on the Winchester interface. This write-only register is loaded with the cylinder number divided by 4 to achieve a range of 1024 cylinders. For example, if write precompensation is desired for cylinder 128 (80 hex) and higher, this register must be loaded with 32 (20 hex). The write

precompensation delay is fixed at 10 nS from nominal.

5.5.9 Switches Register

The switches register is preset to the value of six switches on the WFC board. The host reads a closed switch as a one, an open switch as a zero. The host may write over the value in this register, and the new value remains in the register until it is written over again or until a reset places the value of the switches in the register.

5.5.10 Status And Error Registers

Use the status and error registers (table 5-E) to monitor the execution of commands. Each bit of these registers defines a particular kind of status or error condition:

Table 5-E
Status and Error Registers

bit	Status Register	Error Register
7	Busy	Bad Block Detect
6	Ready	CRC Error - Data Field
5	Write Fault	CRC Error - ID Field
4	Seek Complete	ID Not Found
3	Bus Error	Internal Error
2	Interrupt	Aborted Command
1	Write Protected	TR0000 Error
0	Error	DAM Not Found

After a command is executed, status information pertaining to that command is loaded into the status register. The host must read this read-only register to determine successful execution of the command. If the busy bit is set, all other bits in this register are invalid. Accessing this register resets the INTRQ.

The error register contains specific fault information pertaining to the last command executed. This register is valid only if the error bit in the status register is set. The error register is read-only.

Status Register Bits

Error

When set, bit zero indicates that one or more bits are set in the error register. This bit is an efficient means for the host to check for an error condition. This bit is reset on receipt of a new command.

Seek Complete

Bit four indicates the conditions of the seek complete line on the selected drive.

Write Fault

Bit five indicates the condition of the write fault line on a selected drive for the Winchesters and the write fault status from the floppy controller chip for the floppy drives. The WFC ignores all commands if this bit is set.

Ready

Bit six indicates the condition of the ready line of the selected drive. The WFC ignores all commands unless the ready bit is set.

Busy

Bit seven is set after every command to indicate that the WFC is busy executing a command. All other bits and registers are invalid when this bit is set.

Bus Error

When set, the bus error bit indicates that the memory did not respond within 16 uS after a read or write signal was given during a DMA transfer.

Interrupt Request

When an interrupt is requested, bit two is set. Interrupt request bit is cleared by reading the status register.

Write Protected

The write protected bit (bit one) applies only to the floppy drives and indicates that a write was attempted on a write protected drive.

Error Register Bits

Data Address Mark (DAM) Not Found

Bit zero is set during a read sector command if, after successfully identifying the ID field, the data address mark went undetected for 16 bytes after the ID field. It applies to Winchesters only.

TR000 Error

Bit one is set during a restore command if, after issuing 1023 stepping pulses on the Winchesters and 256 stepping pulses for the floppy drives, TRACK 000 line is not asserted by the drive.

Aborted Command

Indicates that a valid command has been received that cannot be executed based on status information from the drive. For example, if a write sector command has been issued while the write fault line is set, the aborted command bit (bit two) will be set. The host must interrogate the status and/or error registers to determine the cause of failure.

Internal Error

Bit three is set when the controller encounters a controller error which is the result of a malfunction of the controller hardware.

ID Not Found

When set, bit four indicates that an ID field containing a specified cylinder, head, sector number, or sector size was not found.

CRC Error - ID Field

Indicates that a CRC error was encountered in an ID field.

CRC Error - Data Field

Indicates that a CRC error was encountered in the data field of the sector being read.

Bad Block Detect

Indicates that a bad block mark has been detected in the specified ID field. If the command issued was a write sector command, no writing is performed. If generated from a read sector command, the data field is not read. Note that a bad block will go undetected if the flaw is in the ID field.

5.5.11 Macro Commands

The WFC executes five easy to use macro commands. Most macros feature automatic implied seek, so the host system need not tell the WFC where the R/W heads of each drive are or when to move them. The controller automatically performs all needed retries on all errors encountered including data CRC errors. If the R/W head mispositions, the WFC automatically executes restore and seek. If the error is completely unrecoverable, the WFC simulates a normal software completion to simplify the host system's software.

Commands are executed by loading the command byte into the command register while the controller is idle. (The controller is idle after it has completed the previous command.) The task file must be loaded before a command is issued. No command executes if the seek complete or ready lines are false or if the write fault line is true. You can normally issue a command without polling the seek complete line, the ready line, and the write fault line.

The five macro commands fall into three types summarized in table 5-F.

Table 5-F
Macro Command Types

TYPE	COMMAND	BITS							
		7	6	5	4	3	2	1	0
I	Restore	F	0	0	1	r3	r2	r1	r0
I	Seek	F	1	1	1	r3	r2	r1	r0
II	Read Sector	F	0	1	0	0	0	0	0
III	Write Sector	F	0	1	1	0	0	0	0
III	Format Track	F	1	0	1	0	0	0	0

5.5.11.1 Floppy Winchester -

The 5G bit of the macro commands is the floppy/Winchester bit and should be set as follows:

0 = Winchester command
1 = Floppy command

Table 5-G shows the Winchester and floppy stepping rates.

Table 5-G
Winchester & Floppy Stepping Rates

WINCHESTER

r3-r0 - Stepping Rate	
0000 = 10 uS	1000 = 4.0 mS
0001 = 0.5 mS	1001 = 4.5 mS
0010 = 1.0 mS	1010 = 5.0 mS
0011 = 1.5 mS	1011 = 5.5 mS
0100 = 2.0 mS	1100 = 6.0 mS
0101 = 2.5 mS	1101 = 6.5 mS
0110 = 3.0 mS	1110 = 7.0 mS
0111 = 3.5 mS	1111 = 7.5 mS

FLOPPY

r3-r0 - Stepping Rate
XX00 = 6.0 mS
XX01 = 12.0 mS
XX10 = 20.0 mS
XX11 = 30.0 mS

5.5.11.2 Type I Commands -

These commands simply position the R/W heads of the selected drive. Both commands have explicit stepping rate fields. The lower four bits of type I commands form the stepping rate.

Restore

The restore command calibrates the position of the R/W head on each drive by stepping the head outward until the track zero line goes true. The restore sequence is:

1. Upon receipt of the restore command, the busy bit in the status register is set.

2. Cylinder high and cylinder low registers are cleared.

3. The lower four bits of the command byte are stored in the stepping rates register for subsequent implied seeks.

4. The seek complete, ready, and write fault lines are sampled, and if an error condition exists,

   - o The aborted command bit in the error register is set,
   - o The error bit in the status register is set,
   - o An interrupt is generated, and
   - o The busy bit is reset.

5. If no errors are encountered thus far,

   - o The internal head position register for the selected drive is cleared.
   - o The TR000 line is sampled.
   - o If TR000 is true, an interrupt is generated, and the busy bit is reset.
   - o If TR000 is false, stepping pulses at a rate determined by the stepping rate field are issued until the TR000 line is activated.
   - o When TR000 is activated, the busy bit is reset and interrupt is issued.
   - o If the TR000 line is not activated within 1023 stepping pulses for Winchesters or 256 pulses for floppy drives,
     - the TR000 error bit in the error register and the error bit in the status register are set.
     - The busy bit is reset.

- An interrupt is issued.

Seek

The seek command positions the R/W head to a certain cylinder. Execute **seek** primarily to start two or more concurrent seeks on drives that support buffered stepping. (Floppy drives do not support buffered stepping.) Here is the **seek** sequence:

1. The WFC board receives the command, then sets the busy bit in the status register.

2. The lower four bits of the command byte are stored in the stepping rate register for subsequent implied seeks.

3. The seek complete, ready, and write fault lines are sampled, and if an error condition exists,

   - o the aborted command bit in the error register is set,
   - o the error bit in the status register is set,
   - o an interrupt is generated, and
   - o the busy bit is reset.

4. If no errors are encountered thus far,

   - o the internal head position register for the selected drive is updated,
   - o the direction line is set to the proper direction, and
   - o a step pulse is issued for each cylinder to be stepped.

5. When all stepping pulses have been issued, the busy bit is reset, and an interrupt is issued.

NOTE

The seek complete line is not sampled after the seek command, allowing multiple seek operations to be started using drives with buffered seek capability.

5.5.11.3 Type II Command -

Read sector is the command to transfer a block of data from the WFC buffer to the host and has an implicit stepping rate as set by the last restore or seek command.

Read Sector

Execute read sector to read a sector of data from the disk to the host computer. Here is the read sector sequence:

1. The WFC receives the read command

2. The busy bit in the status register is set

3. The WFC samples the seek complete, ready, and write fault lines and if an error condition exists,

   - o the aborted command bit (bit 2) in the error register is set,
   - o the error bit (bit 0) in the status register is set, and
   - o a normal completion is simulated

To understand the type II command thoroughly, read the definitions below of correlative actions

involved in reading a sector.

Implied Seek

If no errors are encountered so far, a seek command is executed. The seek complete line is sampled. If the seek complete line stays false after 128 Winchester index pulses, then the aborted command bit (bit 2) in the error register is set, and a normal completion is simulated.

Retries

Once the head has settled over the desired cylinder, the WFC attempts to read the setor. The WFC executes retries until the data during the read command is recovered. The controller tries to read the desired sector up to 16 times for Winchesters and 10 times for floppy drives. It retries under four conditions:

1. If it does not find an ID
2. If the ID of that sector has a bad CRC
3. If it couldn't find the data address mark (DAM)
4. If the data were actually read from the disk but incurred a data CRC error.

Auto Restore

When the controller encounters an error, it records that error in an internal register. If, after retries, the controller cannot get a match on the ID field, it assumes a mispositioned head and executes an auto restore. During the auto restore, the stepping rate is implied to be equal to the seek complete period. After the auto restore has been successfully completed, the controller executes seek and tries to read the sector once again. An auto restore is performed only once per read or write sector command.

Hard Errors

If the controller encounters a nonrecoverable error, the controller examines its internal error history register. It then sets the bit in the error register of the highest severity error recorded. The data are not transferred to system memory. The error bit in the status register is set and a normal completion is simulated.

5.5.11.4 Type III Commands -

A type III command transfers a block of data from the host to the WFC buffer. These commands have implicit stepping rates as set by the last restore or seek command.

Write Sector

Use write sector to write a sector of data from the host computer to the disk. Having received the write command, the WFC uses DMA to transfer the sector data into the WFC board's sector buffer. After this transfer, the WFC samples the seek complete, ready, and write fault lines.

An error condition generates the same four events as listed under seek and restore (type I commands).

Implied Seek

If no errors are encountered so far, a seek command is executed. The seek complete line is sampled. If the seek complete line remains false after 128 index pulses (five index pulses for the floppy drives), then the WFC sets the aborted command bit (bit 2) in the error register and bit 0 in the status register, generates an interrupt, and resets the busy bit.

Retries

When settled over the desired cylinder, the

head tries to read the sector's ID. The WFC performs all retries necessary to recover the ID during the write command. The controller attempts to read the ID of the desired sector up to 16 times (10 times for the floppy drive). It retries if it does not find an ID or if the ID of that sector has a bad CRC.

Auto Restore

Same as auto restore described under read sector (type II command).

Hard Errors

If the controller encounters a nonrecoverable error, the WFC

- o examines its internal error history register,
- o sets the bit in the error register,
- o sets the error bit in the status register,
- o generates an interrupt, and
- o resets the busy bit.

If the proper sector is located, the WFC writes the sector buffer to the disk, generates an interrupt, and resets the busy bit.

Format Track

Use the format command to initialize the ID and data fields on a particular disk. Upon receipt of the format command, the WFC sets the busy bit in the status register and transfers the interleave table to the buffer. Information on setting up an interleave table can be found in the next section (Programming) of this manual. In all cases, the number of bytes transferred to the buffer must correspond to the current sector size.

After all data have been sent to the buffer, the

controller samples the seek complete, ready, and write fault lines. An error condition causes the same four events described for type I commands.

Implied Seek

If no errors are encountered so far, the controller executes a seek command No verification of track positioning accuracy is performed because no ID fields may be on the track. After the seek operation has been performed, the seek complete line is sampled. If the seek complete line is not set for 128 index pulses (five index pulses for the floppy drives), the WFC sets the aborted command bit in the error register, generates an interrupt, and resets the busy bit.

Once the head has settled over the designated cylinder, the controller starts writing a pattern of 4Es until the index is encountered. Once the index is found, the controller writes a number of ID fields and nulled data fields to the disk. The number of sectors written is equal to the contents of the sector count register. As each sector is written, the sector count register is decremented, and consequently, must be updated before each format operation.

After the last sector is written, the controller back fills the track with 4Es. Formatting terminates when the next index pulse is encountered. After termination, the WFC generates an interrupt and resets the busy bit.

Sector Format

The structure of the sector is shown in table 5-H.

Table 5-H
Sector Format

gap 4 (4E)	gap 3 (4E)	14 bytes (00)	AM (A1)	cyl hi	cyl low	SH	sec	CRC -2-

12 bytes (00)	DAM (A1)	(F8)	data field	CRC -2-	3 bytes (00)

NOTES:

1. When MSB of head byte = 1, bad block is detected.

2. Write Gate turn-on is 3 bytes after the ID field's CRC bytes.

3. Write Gate turn-off is 3 bytes after the data field's CRC bytes.

4. 12 bytes of zeros are rewritten on a data field update.

5. The 2 LSBs of the IDENT byte are used for cylinder high. These values are:

   FE - 0 - 255 cylinders
   FF - 256 to 511 cylinders
   FC - 512 to 767 cylinders
   FD - 768 to 1023 cylinders

6. GAP values are:

Sector Length	Gap 3	Gap 4	Sector Count
128	15	356	54
256	15	352	32
512	30	800	17

5.6 PROGRAMMING THE WFC BOARD

The WFC incorporates a substantial amount of the intelligence normally found in the host computer. Consider these five features:

1. The WFC performs all needed retries, even on data CRC and head positioning errors.

2. Most commands feature automatic implied seek, so seek commands need not be issued to perform basic read/write functions.

3. The WFC keeps track of the position of up to eight read/write assemblies, so the host need not maintain track tables.

4. All transfers to and from the disk are through an on-board full sector buffer and use DMA.

5. In the event of an unrecoverable error, the WFC simulates a normal completion so that special error recovery software is superfluous.

5.6.1 Setting Up Task Files

Before you execute the five macro commands you must establish a set of parameter registers called the task file. A normal read or write sector operation requires that you write four parameters in the order given below:

TASK FILE PARAMETERS

1. The sector number
2. The size/drive/head
3. The cylinder number
4. The command registers

For most commands, writing these parameters tells the WFC the exact disk location where the transfer should take place.

Although these registers are not normally read from, most of them are readable as well as writable. The read feature is provided so that error reporting routines can determine the physical location of an error without recalculating the sector, head, and cylinder parameters.

Controller Recall: The WFC can recall all the task file parameters it receives, thus storing these parameters in the WFC as they are calculated. This means only one copy of the information is maintained and saves the programmer a few instructions and microseconds.

Cylinders and Tracks: Because most hard disk drives contain more than one head per positioner, it is more efficient to step the R/W head assemblies of most disk drives by cylinders, not by tracks. In other words, design the disk driver to read or write all data that is directly accessible by all the heads on a positioner before stepping to a new cylinder.

Table 5-I illustrates a cylinder-by-cylinder sequential file read on a four-head, two-platter disk drive.

Table 5-I
Stepping the Head Assemblies by Cylinders

Physical Cylinder	Logical Head Number	Physical Head Side	Physical Platter
25	3	top	B
26	0	bottom	A
26	1	top	A
26	2	bottom	B
26	3	top	B
27	0	bottom	A

5.6.2 Type I Command Programming (Restore And Seek)

Restore and seek commands position the R/W heads of the selected drive and set the implied stepping rate register. No data are transferred to or from system memory. To execute a restore or a seek, the system software must execute three functions in sequence.

1. Set up task file and issue command (WFC attempts to execute type I command).

2. Wait for the WFC to interrupt or to reset the busy bit in status register.

3. Check error bit in status register for proper completion.

Use of Busy Bit: On smaller, single-user systems one way to tell the CPU that a command has executed is to poll the busy bit (bit 7) of the status register. The WFC sets bit 7 whenever the controller starts a disk operation and resets it whenever the controller is ready to communicate with the host.

Use of Interrupts: Using interrupts is a more efficient way to notify the CPU that the WFC has completed a command. The INTRQ line on the WFC is pulled low whenever the disk controller requires host CPU intervention. This allows the host to run other tasks while the WFC is reading or writing data to the disk.

Use of the Error Bit: Encountering an error has no effect on the operation of the WFC, because the WFC simulates normal completions. The only way to check error status is to check the error bit in the status register. The contents of the error register are invalid unless the error bit is set.

5.6.3 Type II Command Programming (Read Sector)

The read sector command entails the transfer of a block of data from the WFC buffer to the host and features implied seek with an implicit stepping rate. With interrupts, the system software reads a sector by:

- o Setting up the task file and issuing a command.
- o Waiting for an interrupt from WFC.
- o Checking error bit in status register for proper completion.

Block Moves: The WFC performs all transfers between it and the disk drive through an on-board full sector buffer.

Once the disk has been read, the data are transferred to the system memory using DMA.

5.6.4 Type III Command Programming (Write Sector)

The write sector command uses DMA to fill the sector buffer of the WFC with user data. Write sector and format track entail the transfer of a block of data from the host to the WFC buffer. As with read sector, the type III commands feature implied seek with an implicit stepping rate. To write a sector or format a track, the system software:

- o Sets up task file and issues a command.
- o Waits for an interrupt from WFC.
- o Checks the error bit in status register for proper completion.

Formatting: The format command fills the sector buffer with interleave and bad block information. Two bytes are written to the buffer for each sector to be formatted. The first byte is either a 00 or an 80 hex. If the first byte is a 00, the sector is marked as good. If the first byte is an 80, the sector sets the bad block bit in the satus register if there is any attempt to read or write to it. Please see cautions in the section on bad block mapping.

The second byte is the logical sector number of the next sector to be formatted. This number is recorded on the disk.

Format Operation & Sector Count Register: During each format operation on a 32 sector per track disk, 32 pairs (64 bytes) of formatting information must be supplied to the drive. To start the format operation, the buffer must be completely filled even if the sector table is shorter than the buffer. For example, if the sector size of the disk is 256 bytes, then the format operation will DMA in 192 bytes of garbage before starting.

Because the contents of the sector buffer do not imply how many sectors are to be formatted, a dedicated sector count register is provided. This register must be loaded with

the number of sectors to be formatted before each format operation.

Interleaving: Interleaving is a mapping technique in which logically sequential sectors are not written in physical sequence on the disk. This allows for delays in processing WFC interrupts.

Physically sequential disk sectors pass the read/write head so fast that one rotation happens too quickly to allow the head to read or write the trailing sector. The disk has to rotate a complete turn to pick up the next sector. To read all 32 sectors on a particular track would, therefore, take 32 rotations (about one half second per 8K bytes). This operation can be accelerated through interleaving, thus allowing the system to read or write more than one sector per rotation.

INTERLEAVING EXAMPLE

Suppose a system takes less than three sector times (3/32 rotational period) to digest the data that it has read and to set up the next read operation. Placing the second logical sector physically four sectors away from the first one allows the controller to read that next sector without much delay. This four-to-one interleave factor supports reading the entire track in only four rotations. In our particular example, this increases the throughput by a factor of eight.

Experiment to determine the optimum interleave for your particular system. If the system maintains its directories on the disk separate from their data, it might make good sense to arrange two interleaves, one for regular disk operations, the other for directory functions.

AUTOMATIC MAPPING

To simplify driver software, the WFC automatically maps logical to physical sectors to achieve interleave. This logical to physical map is recorded on each track of the disk in the ID fields of the sectors. This map is recorded on the disk during format operations. Table 5-J shows an example of an interleave table for a 32-sector track with 4:1 interleave and no bad blocks.

Table 5-J
Interleave Table, 32 Sectors, 4:1 Interleave

```
00 00 00 08 00 10 00 18 [00 01] 00 09 00 11 00 19
00 02 00 0A 00 12 00 0A  00 03  00 0B 00 13 00 1B
00 04 00 0C 00 14 00 1C  00 05  00 0D 00 15 00 1D
00 06 00 0E 00 16 00 1E  00 07  00 0F 00 17 00 1F
```

The first byte in each byte pair in the preceding example is set to 00. This marks each block as a good block. The second byte of each pair is the logical sector number. The first byte pair above represents the first logical sector of the track. The byte pair in brackets represents the second logical sector.

5.6.5 Mapping Bad Blocks

Storage media (disks and tapes) are manufactured with an unpredictable number of flaws (bad blocks). This industry-wide problem necessitates mapping affected media so that data transfer can accommodate the bad blocks.

Mapping the bad blocks can be done many ways, some of which depend on the operating system. The Multi-user Control System (MCS) automatically records bad blocks and avoids writing to them, although MCS does try to read them. Four suggestions for mapping inherent media imperfections follow in this subsection.

Sector Pre-allocation

If the operating system supports random sector or group allocation, the bad blocks can sometimes be mapped by recording an undeletable file using all the bad sectors on the disk. When the operating system tries to write to the bad block, it sees that the sector or group that contains the error has already been allocated. The operating system then automatically maps over the bad sector.

Sector pre-allocation entails one minor restriction. Never move the file that contains the bad sector to another section of the disk. The bad sector may not be read (for obvious reasons), and reads or writes to the disk that fail to consult the disk allocation map (physical reads/writes) are disallowed.

Alternate Tracks

Whenever a read or write is attempted, the track number (cylinder and head select) is checked against a table maintained by the operating system or driver. If the track number matches the table, the driver recognizes a flaw somewhere on that track and searches for the track that is the alternate for the specific flawed track. The read or write is performed on the alternate track.

The primary disadvantage of mapping bad blocks via the alternate tracks method is that it requires more software overhead. When the system is brought up, the alternate track table has to be read from a flawless area of the disk. After reading the table, the operating system must check the table for every read or write before executing a respective operation.

Alternate Sectors

Mapping bad blocks via alternate sectors is probably the simplest method to implement. During format, the WFC writes the physical sector that contains the flaw with some illegal sector number. The physical sector following the flawed sector contains the real logical sector and its data. Table 5-K shows that a user mapped the fifth physical sector by telling the WFC to write a logical sector number of FF to it.

Table 5-K
Interleave Table (Alternate Sectors Method)

32 sectors and 4:1 interleave with
physical sector five mapped out

00	00	00	08	00	10	00	18	[00	FF]	00	01	00	09	00	11
00	19	00	02	00	0A	00	12	00	1A	00	03	00	0B	00	13
00	1B	00	04	00	0C	00	14	00	1C	00	05	00	0D	00	15
00	1D	00	06	00	0E	00	16	00	1E	00	07	00	0F	00	17

NOTE

When formatting the disk as in table 5K, at least one sector must have an illegal sector number. Also, because we have allocated one sector to bad block mapping, we no longer have a sector 1F.

The primary disadvantage is that at least one sector must be set aside as a spare for each track.

Bad Block Bit

With the WFC you can set a marker for a bad block and have that marker recorded into the ID field. When the WFC attempts to read or write a sector with a bad block marker set, the operation is aborted and the error bit in the status register and the bad block bit in the error register are set. The size, head, cylinder, sector, and ID CRC fields of the selected sector must be correct in order to detect a bad block mark, so it is impossible to use the bad block marker to map out bad areas in an ID field.

In table 5-L the user has marked the fifth sector (logical sector 1) as a bad block.

Table 5-L
Interleave Table (Bad Block Bit Method)

32 sectors and 4:1 interleave with
logical sector one marked as a bad block

00	00	00	08	00	10	00	18	[80	01]	00	09	00	11	00	19
00	02	00	0A	00	12	00	1A	00	03	00	0B	00	13	00	1B
00	04	00	0C	00	14	00	1C	00	05	00	0D	00	15	00	1D
00	06	00	0E	00	16	00	1E	00	07	00	0F	00	17	00	1F

A flaw in the ID field prevents reading the bad block bit. Therefore, avoid using this method as your primary means for mapping bad blocks.

5.7 SOFTWARE INTERFACE DATA

Sixteen registers communicate the command and status information. The registers are offset from a base address that can be located on any 32-byte boundary in the I/O space.

Table 5-M is an I/O map for the WFC's registers. The System 150 base address is hexadecimal F00180.

NOTE

The addresses are 68000 addresses. The least significant address on the 68000 (ADDR0) is inverted to that of the Multibus.

Table 5-M
Winchester Registers

I/O port Address		Input Command I/O Read	Output Command I/O Write
68000 Base	IEEE 796 Plus		
0	1	Error	Write Precompensation
1	0	Switches	Switches
2	3	Sector Number	Sector Number
3	2	Sector Count	Sector Count
4	5	Cylinder High	Cylinder High
5	4	Cylinder Low	Cylinder Low
6	7	Status Register	Command Register
7	6	Size/Drive/Head	Size/Drive/Head
8	9	Reserved	Reserved
9	8	Reserved	Reserved
10	11	Reserved	Reserved
11	10	Reserved	Reserved
12	13	DMA Addr. 9-16	DMA Addr. 9-16
13	12	DMA Addr. 1-8	DMA Addr. 1-8
14	15	Reserved	Reserved
15	14	DMA Addr. 17-23	DMA Addr. 17-23

5.7.1 34-Pin Winchester Drive Control Connector

Drive control connector (J7) is a 34-pin vertical header on tenth-inch centers that mates with Burndy FRS34BS. Use a flat ribbon cable or twisted pair under 10 feet long.

Table 5-N shows the cable pin-outs for this connector.

Table 5-N
34-pin Winchester Control Connector Pin-outs

Signal Ground	Signal Pin	I/O	Signal Name
1	2	O	-RWC
3	4	O	-Head Select 2
5	6	O	-Write Gate
7	8	I	-Seek Complete
9	10	I	-TR000
11	12	I	-Write Fault
13	14	O	-Head Select 0
15	16		NC
17	18	O	-Head Select 1
19	20	I	-Index
21	22	I	-Ready
23	24	O	-Step
25	26	O	-Drive Select 1
27	28	O	-Drive Select 2
29	30	O	-Drive Select 3
31	32	O	-Drive Select 4
33	34	O	-Direction In
35	36	O	-Step

5.7.2 Winchester Drive Data Connector

Four data connectors (J1-J4) are provided for clock signals and data between the WFC and each drive. All lines associated with the data transfer between the drive and the WFC are different and may not be multiplexed. The data connectors are 20-pin vertical headers on tenth-inch centers that mate with Burndy FRS20BS. Use a flat ribbon cable or twisted pair less than 10 feet long.

Table 5-O shows the cable pin-outs for this data connector.

Table 5-O
Data Connector Pin-outs

Signal Ground	Signal Pin	I/O	Signal Name
2	1	I	-Drive Selected
4	3		NC
6	5		NC
8	7		NC
	9	O	+Timing Clock
	10	O	-Timing Clock
11			GND
12			GND
	13	O	+MFM Write Data
	14	O	-MFM Write Data
16			GND
	17	I	+MFM Read Data
	18	I	-MFM Read Data
19			GND
20			GND

5.7.3 Drive Control Timing

Table 5-P shows the WFC drive control timing.

Table 5-P
WFC Drive Control Timing

Symbol	Characteristic	Min .	Max	Units
tWG	Write gate pulse width	1 sector	2 rotation	
tDS	Direction to step delay	250		nS
tSW	Step pulse width	5(typical)		uS
tSP	Programmed step pulse period	0.01	7.5	uS
tSS	Step to seek complete false		9	uS
tSC	Last step to seek complete		128	index times

Notes:

1. tWG depends on the sector size and the rotational rate of the disk.

2. tSP equals seek complete time during auto restore.

5.7.4 Drive Data Timing

Table 5-Q shows the WFC drive data timing.

Table 5-Q
Drive Data Timing

Symbol	Characteristic	Min	Max	Units
tTC	Timing clock period	WCLK/16 (typical)		
tWD	Write data pulse width	60	120	nS
tRD	Read data pulse width	25		nS

5.7.5 Adjustment Procedures

Temperature affects the parameters of the board components. Therefore, procedures for adjusting the floppy and the Winchester drives both require a five-minute warmup after power is applied to the board.

Adjusting the Floppy Data Separator

1. Apply power to the board and wait five minutes.

2. Verify that either U42 pin 11 or U47 pin 8 is always high. If this is not the case, deselect all floppy drives. The drives are deselected after each command and reset.

3. Attach a voltmeter to U41 pin 2 and adjust R45 until the meter reads 1.4 +/- 0.05 volts.

4. After adjusting R45, attach a frequency counter on U41 pin 7 and adjust R46 until the counter reads 2 MHz +/- 20 kHz.

5. While writing continuously to the floppy disk, put an oscilloscope on U42 pin 4 and set it to trigger on a low going edge. Adjust R47 until the low going pulse is 300 nS +/- 10 nS.

Adjusting the Winchester Data Separator

1. Apply power to the board and wait for five minutes.

2. Monitor TP14 with a 10X oscilloscope probe while attempting to read continuously from a Winchester drive. The scope should be set to trigger on a high to low transition. While observing TP14, adjust R48 until the low pulse from the DRUN single-shot is 250 nS +/- 5 nS.

3. Ground the cathode of the tuning diode CR3 to the closest accessible ground using a low inductance shorting cable. This cable should consist of the shortest piece of wire able to make the connection.

4. Ensure a logic 1 exists at U20 pin 4. A logic 0 at this point inhibits the VCO and prohibits adjustments. If a logic 1 is not present, verify that the DRUN circuitry is functioning and adjusted properly.

5. Connect a frequency counter to the buffered VCO output at TP9. Verify the range of adjustment by first verifying that the counter reads 9.0 MHz by adjusting R50. Now verify that the counter reads 11 MHz by adjusting R50.

6. After verifying the range of adjustment, vary R50 until the frequency counter reads 10.00 MHz +/- 10 kHz.

7. Disconnect all test jumpers and test equipment in preparation for step nine.

8. Ground U27 pin 4 and connect a 100-ohm resistor between U19 pin 8 and ground.

9. Connect a voltmeter to TP5 and adjust the BAL pot R49 until the meter reads 0 V +/- 20 mV on TP5.

10. Disconnect all jumpers and test equipment.

CHAPTER 6

DEI CARTRIDGE TAPE CONTROL BOARD

6.1 INTRODUCTION

The controller for the DEI cartridge tape drive uses an 8085 microprocessor, and employs firmware to support a link to the host multibus and to a maximum of four DEI cartridge tape transports. The firmware ensures long-term flexibility to cope with changing transport technologies. Interface to an encoder/decoder allows the controller to be immune to the recording technique.

For detailed information on the DEI cartridge tape drive itself, refer to the Series CMTD-3400S2 6400 BPI High Density Cartridge Magnetic Tape Drive Operation and Maintenance Manual published by Data Electronics Incorporated, 11633 Sorrento Valley Rd., San Diego, CA 92121.

Standard features of the controller include:

1. Interface via a selectable block of four I/O ports. Completion is signaled via a STATUS READ or any of the eight parallel interrupt requests.

2. Power required: five volts at less than two amperes

3. Automatic READ/WRITE retry facility significantly reduces software burden

4. Standard command set includes:

   READ
   WRITE
   WRITE FILE MARK
   FILE AND RECORD SKIP (bi-directional)

5. Off-line tape to tape COPY; archival or distribution copies are thus possible with no host interaction

6. High-speed file and record skip

7. Record search under up to a 512 byte mask

8. Record lengths from 512 to 8208 under host control

6.2 DRIVE INTERFACE REQUIREMENTS

The interface between the drive and the controller is signal ended, TTL.

All input and output lines are defined as true low (0 to .8 VDC), and all drive output lines are driven by an open collector or tri-state driver (25 ma maximum sink current). Input lines into the drive supply various 74XX gates.

6.3 PERFORMANCE SPECIFICATIONS

When the drive is operated within the previously described conditions, the following performance shall be achieved:

6.3.1 Tape Motion - Steady State

Bit Period Definitions

The nominal bit periods for the various drive speeds and the corresponding nominal data transfer rates are shown in table 6-A.

Table 6-A

SPEED RANGE	NOMINAL APPARENT SPEED (IN. PER SEC)	NOMINAL BIT PERIOD (MICROSECONDS)	NOMINAL DATA TRANSFER RATE (BITS PER SECOND)
LOW	30	5.21	192,000
HIGH	90	N.A.	N.A.

Long Term Tolerances

The long term average of the nominal bit period at an apparent 6400 flux reversal per inch (frpi) as shown in table 6-A will be within ± 2% for 30 ips. The long term average of the nominal bit period will be within ± 4% for high speeds.

The long term average bit period measured in a reverse direction and forward direction shall be within +/-3%.

The bit period shall be as seen at the read data signal exiting the drive and use defined as being measured over 150 inches (3.81 m) of tape or more. The total variations are listed in Table 6-B.

Table 6-B

	LONG TERM AVERAGE TOLERANCE			SHORT TERM AVERAGE TOLERANCE		
CONDITION	DRIVE	CARTRIDGE	TOTAL	DRIVE	CARTRIDGE	TOTAL
LOW SPEED	+/-2%	+/-1%	+/-3%	+/-3%	+/-4%	+/-7%
HIGH SPEED	+/-4%	+/-1%	+/-5%	+/-3%	+/-4%	+/-7%

Short Term Tolerances

The short term average bit period at 6400 frpi as shown in Table 6-B will be within ± 3% of the long term average bit period during recording. In a subsequent read mode an

additional ± 7% short term speed can occur. The short term speed observations shall be from 5Hz to 20 KHz.

6.3.2 Reliability And Failures

These failure definitions should be used in conjunction with the Data Electronics, Inc. warranty and recommended preventive maintenance schedule; neither is contained within this document.

Mechanical Failure

A mechanical failure is defined as a persistent failure of the drive mechanism to perform as specified and includes the ability to load, unload and properly position the cartridge for the purposes of recording and reproducing data and to act as basic mechanical structure for other drive components.

Control Failure

A control failure is defined as a persistent failure of the drive to respond to external command or supply a status signal as specified. This assumes that all of the specified requirements have been met during the unit's life.

Data Failure

A data failure is defined as a persistant failure in the drive to write or read data as specified with an error rate in excess of that specified. The error rate assumes a cartridge that has been certified free of recordable areas which fall below 30% of the nominal amplitude at 6400 frpi. Cartridge permanent errors are to be excluded. A temporary (soft) error is defined as any particular error which persists for ten (10) or less successive read and/or write attempts. A permanent (hard) error is one which persists after ten retries in the read or write mode.

Data Reliability

The temporary error rate shall be less than 1 error in 108 bits read wherein recording and reading is done on a drive

as specified.

The drive, in a laboratory environment, shall have a permanent error rate of less than 1 error in 1010 bits read assuming the recording and reading is done on a drive as specified.

6.4 DATA INPUT FORMAT

Pre and Postambles

An non-data bearing pattern of at least 39 'zeros' followed by one (1) 'one' is added to the beginning of each data block. This pattern is called a preamble. The reverse pattern (i.e., one (1) 'one' followed by at least thirty-nine (39) 'zeros' is added to the end of each data block. This pattern is called a postamble. The preamble is stripped from the read data by the drive in the forward mode and the postamble is stripped from the read data in the reverse mode.

Hence, a data block is:

$\geq$ 39 'zeros'; 1 'one', Data, 1 'one', and $\geq$ 39 'zeros'.

Check Characters

Any check characters shall be agreed to by the interchanging parties. ANSI CRCC: X/16 + X/15 + X/2 + 1 is often adopted.

Minimum Data Block Size

A data block contains at least 1024 data bits.

Tape Mark

A tape mark consists of a preamble, 16 data bits and a postamble.

The Interblock Gap

The minimum interblock gap (IBG) is 1.2 inches. Commanding WEN true causes the removal (using the drive AC erase capability) of

any recoverable data or spurious transitions from the interblock gap.

Number of Tracks

There are four recordable tracks.

6.5 THEORY OF OPERATION

6.5.1 DEI Processor

The Intel 8085 microprocessor (U72) is the heart of the DEI portion of the SMD/DEI board. It operates with a 6.144 MHz crystal yielding a state time of 330 nanoseconds.

The bus provides an external reset (BRESET, active low) to the 8085. When this signal is received, interrupts are disabled and program flow begins at address zero. The 8085 RAM is decoded to the eight Kbyte level by a coarse address map decoder (U33). A memory read or write command prequalifies the RAM's output. Two 2114 static RAMs (U57 and U58) provide one Kbyte of RAM.

The A9 input of the two 2114s is connected to A15 of the processor bus. The bus divides this RAM into two 512-byte sections. The lower half is cyclic within the fourth 8K partition. The upper half is cyclic within the entire upper 32K of the memory map. Two sections of a 74S11 (U56) contribute to this cyclic redundancy.

The 8085 has a shared LSB address and data bus. An 8212 (U86) latches the least significant 8 bits of the address bus during the ALE pulse from the 8085.

The program for the 8085 resides in the 2716 EPROM (U73).

6.5.2 DEI Buffer RAM

Four 6116s (U4, U5, U19, and U20) make up the main 8K buffer memory on the controller board. A 74LS138 decoder (U18) selects the individual 2K blocks. This block of memory is mapped at 4000-5FFF hex in the 8085's memory map.

Two 74LS138 decoders (U34 and U21) provide these discrete input and output commands. Since they are only qualified by A7, they are cyclic within the lower 128 I/O codes.

6.5.3 DEI Bus Interface

Jumper blocks JP12-JP16, in conjunction with comparators U107, U119-U121, and U133, determine the address of the board. The board is trapped to respond to bus addresses FDA000 through F0A006 hex (even byte only).

All qualified output commands from the bus have their data latched by an 8212 (U96) during the DBW pulse. Further, in order to determine to which output port the data were sent, the states of busses ADDR1 and ADDR2 are latched by a 74LS74 (U97) during an output to the controller board.

Each time the bus issues an output command, the INT output from the U96 latch is activated simultaneously. This causes an RST 5.5 interrupt to the 8085. If the output is to Port 0 (DATA), an RST 6.5 interrupt is generated. The latter interrupt is of higher priority to the 8085, and is serviced first. These interrupts remain asserted until the latch is read by the 8085. Until the latch is cleared, a BSY (active low) is indicated to the bus each time the status port is polled.

In a similar manner, when the controller board has data to pass to the bus, the data bits are loaded into latch 8212 (U95). This latch is pollable by the bus via the status port INTG. Once the bus determines that the data bits are available, it reads them.

The controller passes the data onto the bus during the qualified BDR. Reading the data clears the INT output of U95, which is connected to the serial input data line (SID) of the 8085. This tells the controller board when it may output new data. Again, as above, the source (DS, IS, or DATA) is signalled by the latching of address bits 4 and 5 by a 74LS368 (U110) during the latch load command. The bus can sense this vector by decoding data bits 6 and 7 during a status read.

The controller board asserts serial output data (SOD) from the 8085 each time it loads the drive status (DS). This eliminates the bus's need to poll to determine when a command has been completed. It also activates an asynchronous interrupt to the bus through jumper block JP19. If you want to use the interrupt, you must place a

jumper to the desired interrupt level in JP19 (INT level 4 for System 200).

On all qualified input or output commands, the transfer acknowledge (BDTACK, active low) is generated by ICs 74LS04 (U94) and 74S240 (U85).

6.5.4 DEI Tape Interface

The two 8212 latches (U1 and U2) perform static control of the drive. U1 controls commands such as motion and function, while U2 selects the desired drive. The status of the selected drive is read into the 8085 via the buffer 74LS240 (U3).

The programmable communications interface 2651 (U31) serializes the data to the drive, appends parity, and converts the serial data from the drive to parallel for the 8085. In this application, U31 is operated in synchronous mode with external send and receive clock supplied by the encode/decode electronics in the drive. It is internally double buffered in both directions to provide sufficient compliance for firmware synchronization. A four bit status buffer 74LS240 (U54) allows the 8085 to input critical status with a single instruction.

6.6 OPERATIONAL FUNDAMENTALS

The IEEE 796 BUS TAPE PROTOCOL involves a three byte sequence (plus write data when appropriate). The input sequence includes the mode argument (MA), the positional argument (PA), and the command argument (CA).

The MA specifies the data type (mask or write data), the data record size, the drive selection, and the track selection. The PA contains a count used by the space commands. The CA specifies the actual command to execute.

The returned sequence includes the drive status (DS) and the interface status (IS). The DS reports the status of the currently selected drive. The IS reports the command status and the currently selected drive and track.

All data/status is transferred via four I/O ports as 8-bit bytes. See the individual word explanations in sections 6.10 and 6.11 and the communications flow diagrams in figures 6.1 and 6.2 for

further information.

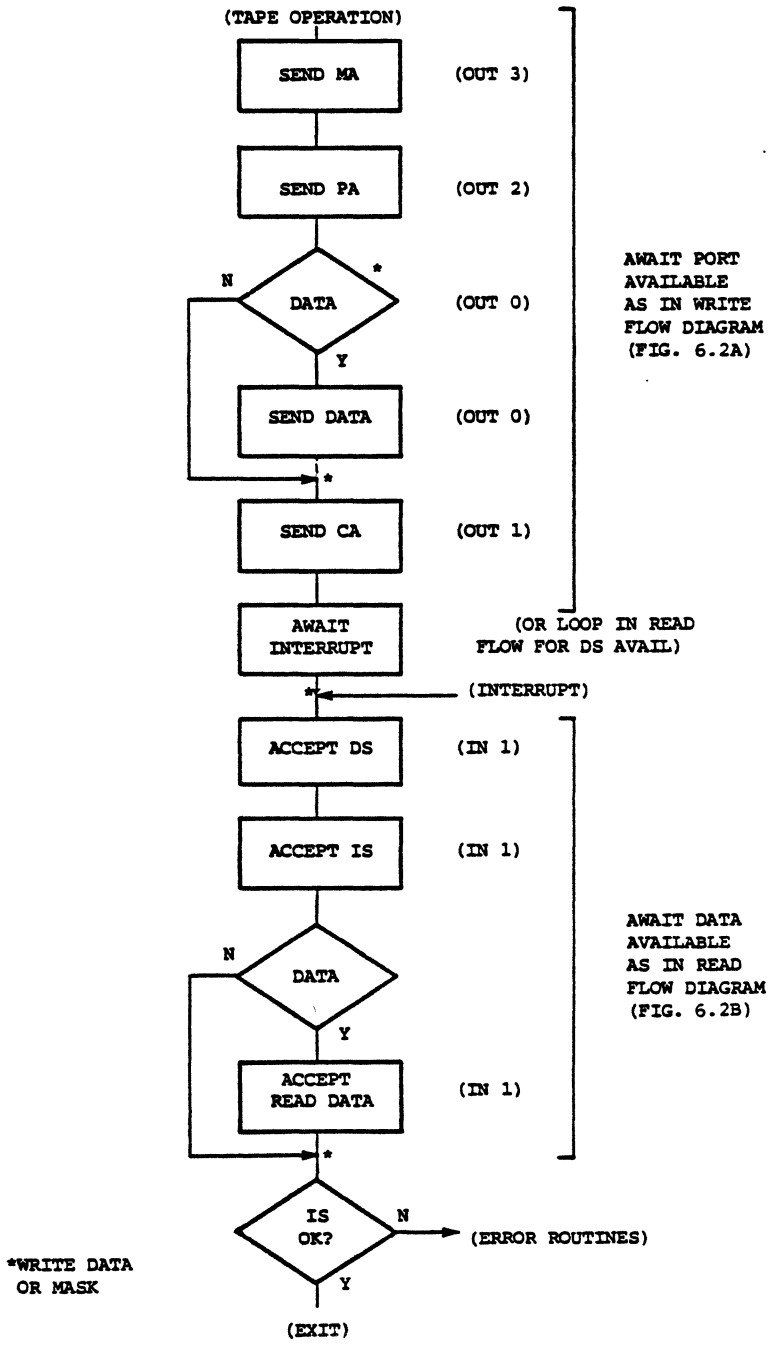


Figure 6.1 Communications Flow

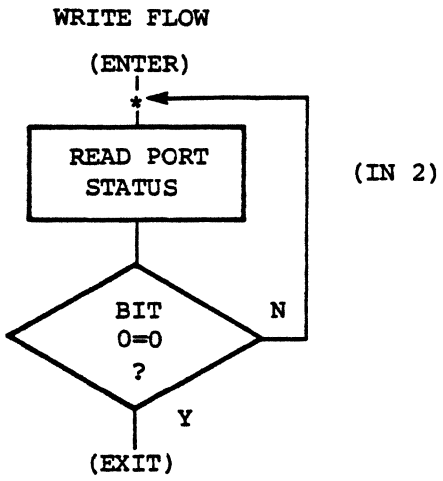


Figure 6.2A Write Flow Diagram

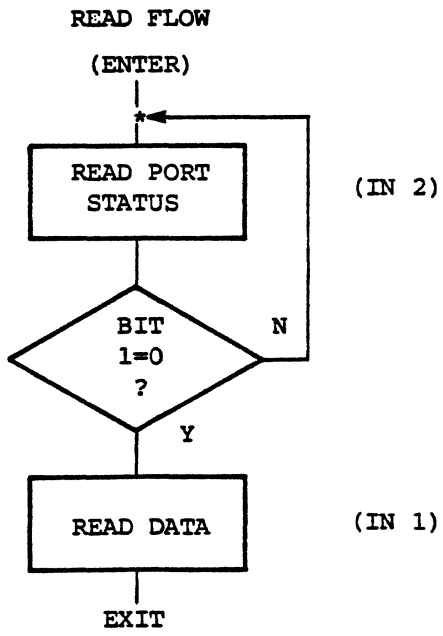
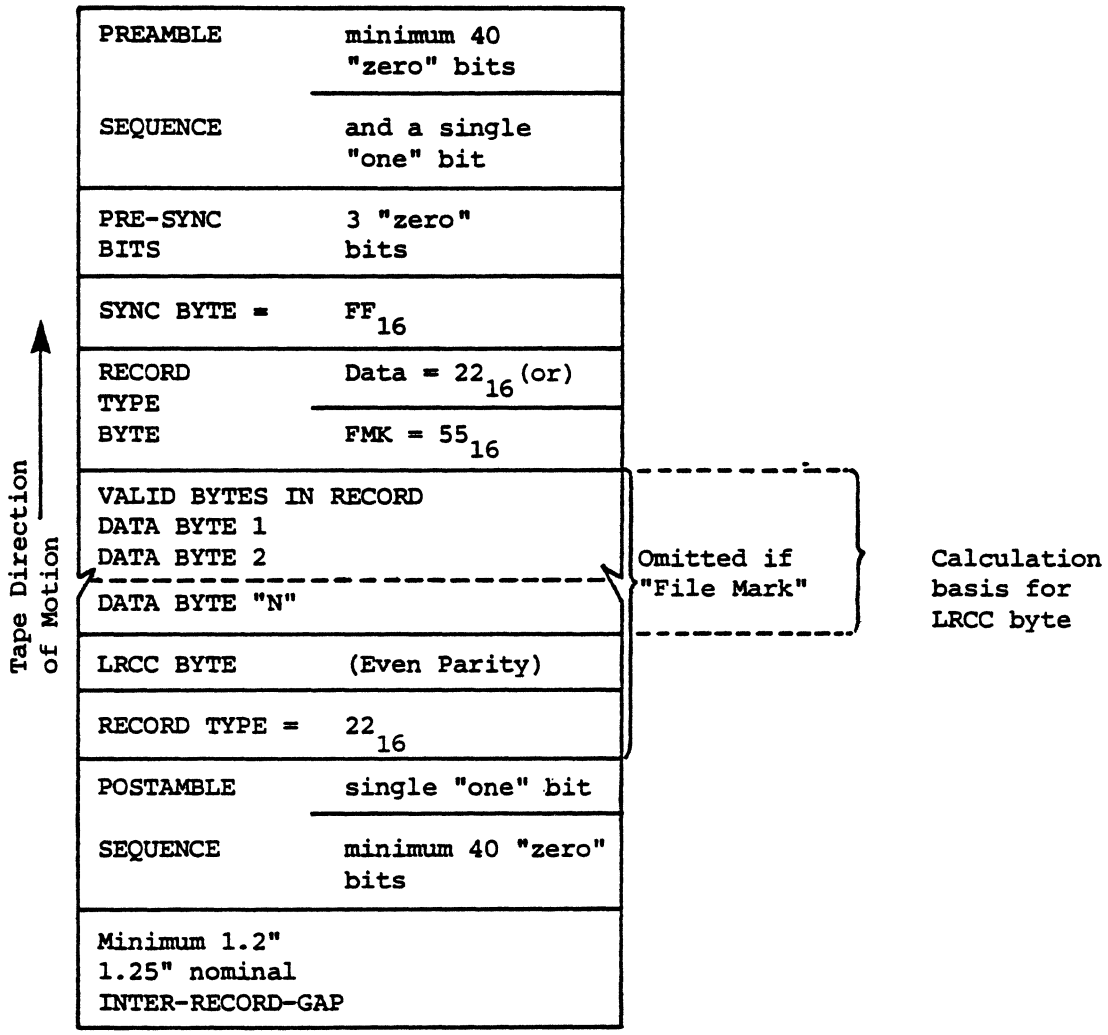


Figure 6.2B Read Flow Diagram

6.7 STORAGE DETAILS

Figure 6.3 diagrams the record structure as it is stored on the four tracks in serial format.



NOTE: All bytes consist of 8 data bits (LSB first)
1 vertical parity bit (even)

Figure 6.3 Record Structure

Table 6-C shows the formatted cartridge capacities in various configurations.

Table 6-C

Tape Storage Capacity

BYTES PER RECORD	6400 BPI RECORDS-MEGABYTES PER TAPE	
	300'	450'
500	7000-3.5	10500- 5.3
1000	5166-5.2	7749- 7.7
1024	5110-5.3	7664- 7.9
2048	3349-6.8	502--10.3
3072	2491-7.7	373 -11.5
3630	2203-8.0	330 -12.0
4096	1983-8.1	297 -12.2
8192	1092-8.9	162 -13.4
16384	575-9.4	8 -14.1
32768	295-9.7	4 -14.5

NOTE: Formatted Capacity based on normal length with 1% file mark content.

6.8 I/O CODES

The following I/O codes are used. Refer to figure 6.1 for sequencing.

		Address
OUT 3 - Set mode argument	MA	F00202
OUT 2 - Set positional argument	PA	F00203
OUT 1 - Set command argument	CA	F00200
OUT 0 - Set data or mask		F00201
IN 1 - Read interface data register		F00200
IN 2 - Read port status		F00203

Bit 0 — 1 = port buffer busy (do not output)
0 = port empty

Bit 1 — 1 = busy (do not output)
0 = data ready for input

Bits 2-5 — unspecified

Bits 7-6 — ready register number

00 = data 10 = IS
01 = DS 11 = not specified

6.9 HOST-TO-INTERFACE TRANSFER WORDS

The actual bit usage and effects of the host-to-interface transfer words (i.e. MA, PA, and CA) are as follows.

DEI CARTRIDGE TAPE CONTROL BOARD
HOST-TO-INTERFACE TRANSFER WORDS

MA	Mode Argument (OUT 3)
B7	Data mode 0 = Data operation 1 = Mask search operation
B6-B4	Data record size 000 = 1024 bytes 100 = 8192 bytes 001 = 2048 bytes 101 = 512 bytes 010 = 3072 bytes 110 = 8208 bytes 011 = 4096 bytes 111 = 3630 bytes
B3-B2	Drive selection 00 = Drive 1 10 = Drive 3 01 = Drive 2 11 = Drive 4
B1-B0	Track selection 00 = ANSI track 1 10 = ANSI track 3 01 = ANSI track 2 11 = ANSI track 4

Note: RESET default = 00 and internal accumulation pointers are initialized.

PA	Positional Argument (OUT 2)
B7-B0	Unsigned values 1-256 used by space commands.

CA	Command Argument (OUT 1)
B7	0 = Normal mode 1 = RAM diagnostic mode (Simulates a READ command in which the RAM contents are transferred as read data.)
B6	If 0, operation is relative to current tape position. If 1, operation is relative to BOT (rewind prior to command execution).
B5	Disable auto rewrite on error facility (Edit mode).
B4	Interface reset if set (allow 1ms following issue). Note: This is the only command that will not return the DS/IS upon completion.
B3-B0	The commands:

0000	No operation (Rewind if relative to BOT) (Retransmit block if not)
0001	Read
0010	Write
0011	Write file mark
0100	Forward space records*
0101	Forward space files
0110	Reverse space records*
0111	Reverse space files
1000	Send current status
1001	Off-line copy
1010	Write with range check
1011	Record search under mask*
1100-1111	High speed commands*

* Aborts if file mark detected

Write Data (OUT 0)

The length of this data block is as specified in the mode argument, MA. If the specified length exceeds the available RAM, zeros are written from the non-existant portion of the RAM. To read back this record, the read operation must be supplied with a compatible mode argument.

Only one size record should be written to a given tape to

facilitate editing and parameter definition. The maximum transfer rate is one byte every 20 microseconds.

Write Mask (OUT 0)

The length of the mask data block is determined by the number of bytes sent prior to the MASK SEARCH command (see Record Search Under Mask, section 6.11).

6.10 INTERFACE-TO-HOST TRANSFER WORDS

The interface-to-host-transfer words are obtained from port 1. This port causes an interrupt condition when ready. This condition is reset by reading the port. The meaning and usage of the interface-to-host transfer words DS and IS are explained here.

DS — Drive Status (IN 1)

All of the following bits are high true.

B7 Reserved

B6 File mark was detected

B5 Drive rewinding

B4 ON with cartridge loaded

B3 BOT (beginning of tape)

B2 EOT early warning (only file mark write operations are allowed under this condition). Thirty-six inches of tape exist between EOT early warning and the physical end of the tape.

B1 Warning flag that the drive has executed an auto rewind (i.e., power-up or remote rewind) since previous command. No relative to current position commands are accepted with this condition present unless preceded by a discrete rewind command.

B0 Write-enable (tape is not safe). This is also true if no cartridge is installed. The data cartridge has a screwdriver slot to alter this condition.

IS	Interface Status (IN 1)
B7	Reserved
B6	Data block follows (high true)
B5	Command status 00 = ok 01 = Abort without attempt* 10 = Abort with attempt** 11 = Syntax rejection or parity error
B3-B2	Current Drive (as per MA, see section 6.9)
B1-B0	Current Track (as per MA, see section 6.9)

* An abort without attempt is indicative of:
1) Reverse space at BOT or forward space at EOT
2) A write operation to a safe cartridge
3) Write data at EOT early warning
4) Any operation to a drive without a cartridge installed

** An Abort with attempt indicates possible tape motion. The following command should be relative to BOT unless care is taken to understand the exact abort criteria and tape position.

Read Data

The length of this data block is determined by the mode argument. If insufficient RAM is optioned to satisfy the specified record size, zeros from non-existant RAM are appended as record padding for each byte. Refer to cautions under **Write Data** in section 6.9.

6.11 COMMAND EXPLANATIONS

This section explains the commands referred to in section 6.9

1 - Read

This command reads a block of data, with length as per MA specification, from the tape and transmits it to the host. Read

errors should be very infrequent due to the read-after-write checking unless the tape has been mishandled or abused. Aborts could possible occur for the following reasons:

1. Dirty or mis-aligned read head

2. Faulty read circuitry

3. Tape was abused or stored near magnetic fields

4. Specified record size did not agree with writing parameter

5. Reading proceeded beyond the second consecutive file mark

6. Incompatible recording format

7. Adjacent record was edited with an inconsistent record size parameter, an out of calibration servo board, or without disabling the auto rewrite facility

A read operation will also terminate if a file mark is detected. No data is transferred in such cases. See Read Data in section 6.10 for further information.

2 - Write

Write accepts a block of data from the host, with length as per the MA value, and writes it onto the tape. The interface attempts to rewrite the record if an error is detected. Each rewrite includes backspacing and erasing 3 inches of tape. The number of retries is application sensitive.

All records are verified during writing using the read-after-write facility on the transport. If end of tape status is sensed during the write to tape, the procedure aborts and the error status is reported.

If BOT is sensed after the record is written, it is reported in the drive status while the interface status indicates command status OK (00). Both of these possibilities must be considered in the host write routines. See Write Data in section 6.9 and figure 6.4.

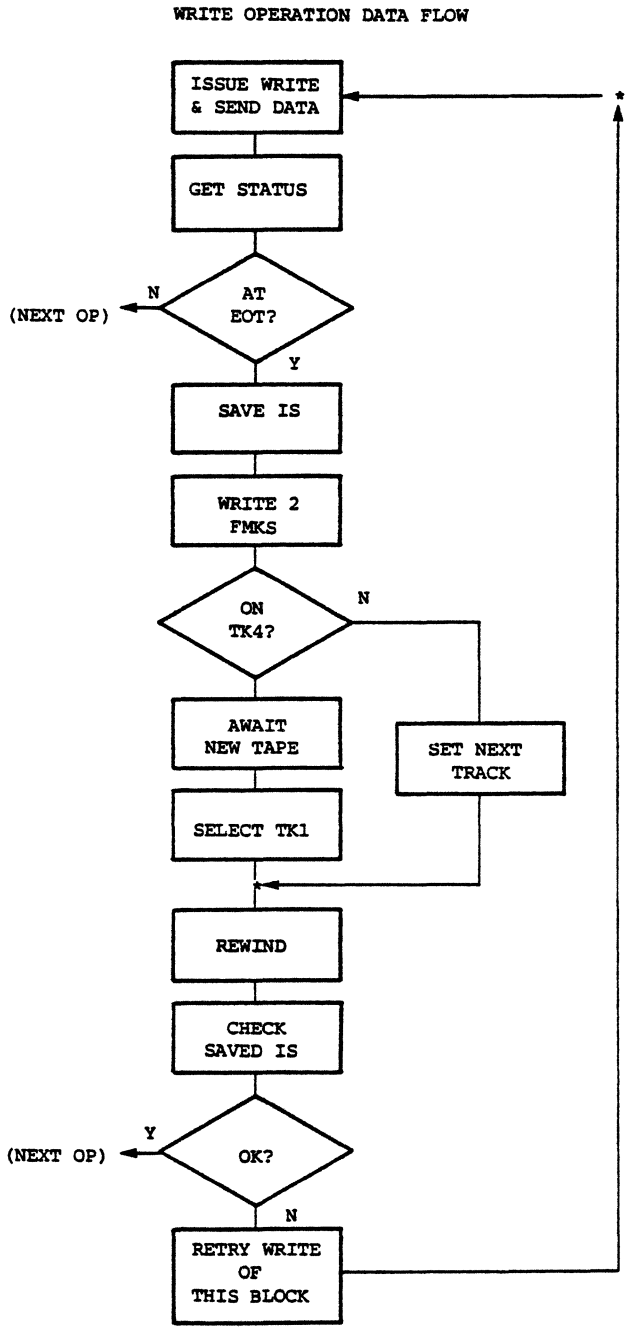


Figure 6.4 Write Operation Data Flow

3 - Write File Mark

The write file mark command writes a file mark on the tape. This mark separates files on the tape, and two consecutive file marks must appear at the end of each track. To facilitate writing file marks, they may be written after receiving end of tape status.

4 & 6 - Space Records

The space records commands space the tape in the specified direction past the number of records specified by the positional argument, PA. Note that the value of the PA is one less than the number of records spaced past. These routines abort and signal an error if a file mark is encountered.

5 & 7 - Space Files

These commands operate in the same way as the Space Records commands but count file marks instead of records. This routine does not abort on file mark detected.

8 - Send Current Status

The current drive status, DS, and the interface status, IS, are returned after the interface receives a ready condition from the selected drive. This command is useful when issued relative to BOT as the statuses are not returned until the tape has completed its rewind operation.

9 - Off-line Copy

This command may be issued to dynamically back-up recorded information. Drive 1 is always the source and drive 2 the destination. The copy firmware requires that the tapes conform to the following format:

1. All data records must be the same length and of a length specifiable by the mode argument, MA. This length is determined from track 1 record 1.

2. All tracks must be logically terminated by two contiguous file marks.

3. The destination cartridge must have a recording capacity equal to or greater than the source cartridge.

10 - Write with Range Check

This function operates in the same manner as a normal write command, except that a syntax rejection message is returned in the IS if the number of data bytes set prior to this command differs from the value indicated by the MA.

11 - Record Search Under Mask

A mask of up to 512 bytes is associated with this command. The mask is transmitted as per write data. The interface then searches the tape for a record whose initial data bytes match the mask. If a mask byte is

$$3F_{16},$$

then the corresponding record byte is ignored (i.e., a match is assumed). When a matching record is found, it is returned in its entirety, as per read data. This command aborts with attempt if a file mark is detected. Note that bit 8 of the MA must be true for the sent data to be accepted as mask rather than as write data.

12 through 15 - High Speed Commands

These commands operate identically with commands 4 through 7 except when the PA value is greater than zero. In that case, they are executed at 90 ips.

6.12 I/O SIGNALS

6.12.1 Input Data Signals

The input signals from the controller to the drive are as follows:

Unit Addressing

Up to eight (8) drives can exist on a single bus. Each drive is individually addressable by the following:

SLG- Select Gate: When true, enables selection per the select or address code. It is used to prevent unwanted transient selection during changes in the select address.

SL4-, SL2-, SL1- Unit Select address in the form of a binary number (true low).

The address is decoded by the drive as follows:

LOGICAL ADDRESS	SL4-	SL2-	SL1-
1	H	H	L
2	H	L	H
3	H	L	L
4	L	H	H
5	L	H	L
6	L	L	H
7	L	L	L
8	H	H	H

The drive is equipped with a unit address switch to provide customer selection of unit addresses. The switch nomenclature numerically corresponds to the logical address as shown in the table above.

The addressing function proceeds and remains during any other drive input and output function except where noted.

Motion

Tape motion is commanded by the following signals:

FWD-: When true, causes the tape to move in a forward direction.

REV-: When true, causes the tape to move in a reverse direction.

HSP-: When true, causes the tape to move at high speed in the direction selected by either FWD- or REV-.

Tape motion proceeds until the command signals go false or:

1. In forward, where an EOT hole is encountered, motion stops.
2. In reverse, where a set of BOT holes is encountered, motion stops.
3. If both directions are commanded simultaneously, the tape stops.
4. A rewind command is received which overrules other motion commands.
5. The Internal Ready signal is not true and motion stops.
6. When in High Speed motion, the drive drops to low speed when the upper "load point" hole is sensed in reverse or the upper "early warning" hole is sensed in forward.

RWD-: When true, causes the tape to be positioned to beginning of tape at high speed. The drive must be selected to start a Rewind sequence, but may be unselected after the sequence is started.

Beginning of tape is defined as between the two innermost (toward the middle of the tape) set of upper and lower holes located at the "head end" of the tape. This location is recommended for unloading the cartridge as the data recording area is completely protected.

Rewind "overrules" all other motion signals. Successive Rewind commands do not cause the tape to be "run off".

Rewind stops if the Internal Ready signal is false.

An automatic Rewind sequence is executed when a cartridge is installed in the drive or when power is applied when a cartridge is installed.

Data Signals

The data input process requires the following:

TR2-, TR1- Track select address: a binary number, (true low)

The track selection is decoded as follows:

ANSI Track Number	TR2-	TR1-
1	H	L
2	L	H
3	L	L
4	H	H

The track selection selects all heads for a given track (i.e., erase, write, and read). The last track selection is stored within the drive even

after deselection by the controller.

WEN-: When true, enables the writing and erasing function for the selected track. The writing and erasing processes occurs only if the cartridge is in the unprotected state (not safe). This signal remains set after deselection. This signal should not be reset until the drive has stopped. This signal should be set prior to drive motion. At least 2 milliseconds should be allowed between the reset of Write Enable and the changing of the track select signals. This signal is reset by either Reverse or High Speed commands.

WDE- Write Data Enable: A control line to the drive which enables the encoding function (sending of write data strobes and the writing of data on tape). The WEN function enables both writing and erasing. WEN causes the write circuits to become active and tape to the erased. After the tape is up to speed and other conditions met, the WDE causes the drive to send data strobes and commences to record flux transitions on the tape.

The first strobe is sent <>5.2 X 10 to the -6 seconds after WDE is true.

All data which is to be written on the tape must be sent to the drive (all preambles, check characters, and postambles).

WNZ- Write Data: During the write data strobe period the state of the inpute write data line is sensed as follows:

WNZ = Low = 1, and
WNZ = High = 0

The state of WNZ is only examined during the write data strobe period. The WNZ signal must be steady .5 X 10 to -6 seconds prior to the write data strobe true period (WDS = Low) and remains during the strobe true period = 1 X 10 to -6 seconds. WNZ is permitted to change on the rising (false going) edge of WDS-.

6.12.2 Output Signals

Data Signals

DAD- Data Detected: Is false except when data have been detected. Data detected requires the receipt of a number of data transitions from the tape without an intervening period of more than 20 X 10 to -6 seconds between transitions being received. Data detected can be used to sense the presence of blocks of data at both low and high speed.

RNZ- Read Data: During the period of read data strobe, when RNZ is low (true), the data is a 'one'; when high, the data is a 'zero'. The RNZ signal remains steady at least 100 X 10 to -9 seconds prior to and after the RDS true period.

RDS- Read Data Strobe: Is low for = 1.0 X 10 to -6 seconds, indicating that RNZ can be sampled during this period.

Read data in the forward direction has the preamble removed but contains all of the postamble. Therefore, CRCC and postamble must be stripped in forward direction, and the CRCC and the preamble must be removed in the reverse direction. To output read data strobes, a series of 'zeros' and a 'one' must be sensed. A drop-out disables RDA if it exists for more than 50 X 10 to -6 seconds.

The read data threshold levels are internally set to three different levels depending on the write and motion commands. Read only threshold occurs when running at low speed without writing. Write threshold occurs when writing. Search threshold occurs when running at high speed.

The use of these threshold levels allows for the verification and establishment of the data reliability and margins during the required read-while-write check.

WDS- Write Data Strobe: The write data strobe is generated within the Encoder/Decoder and is sent

out to indicate when the drive is taking data. The low or true strobe period if = 1.0 x 10 to -6 seconds.

Status Signals

SLD-: Is true when the drive has received its proper unit address.

RDY-: Is true when a cartridge is installed, the sensor lamp is drawing current, and the +5VDC is applied to the drive.

BSY-: Is true when the drive is in an automatic rewind sequence (i.e., when a cartridge is first executing a rewind, forward, or reverse command. This signal goes true when the command is received and remains true until the motion stopps (i.e., 30 milliseconds after low speed motion has been commanded to stop and 80 milliseconds after 90 ips operation has been commanded to stop).

In the case of receipt of a non-executed or illegal command, (FWD at EOT or REV at BOT) this signal is not true, indicating the command is rejected.

FLG-: Is set and latched when an automatic sequence to position the cartridge to BOT has been executed, or a rewind has been completed. This signal is reset by subsequent receipt of a FWD command.

WND-: Is true when a write enable condition is latched within the drive.

FUP-: Is true when a cartridge is installed and it is in the unprotected state (i.e., can be written on).

LPS-: Is set and latched when the upper load point hole (the warning of beginning of tape) is passed in the reverse direction. This signal is internally

reset when the load point hole is subsequently passed in the forward direction. When this signal is true, high speed is disabled. Reverse tape motion is allowed to proceed until the BOT holes are encountered, where the drive stops and accepts only forward commands.

EWS-: Is set and latched when the upper early warning hole (the warning at end of tape) is passed in the forward direction. This signal is internally reset when the early warning hole is subsequently passed in the reverse direction. When this signal is true, high speed is disabled. Forward tape motion is allowed to proceed until the EOT hole is encountered, where the drive stops and accepts only reverse commands.

6.13 POWER AND SIGNAL PIN ASSIGNMENTS

6.13.1 Power Connections

The power pin assignments for all drive configurations are shown in table 6-D. Additional connections for WICAT supplied drives with control and Codec Boards are shown in table 6-E.

Table 6-D
Power Supply Pin Assignments

Pin #	Signal	From	Comments
1	M.N.C.	N.A.	Keying Plug
2	V24+	Power Supply	+24VDC
3	V24-	Power Supply	-24VDC
4	M.N.C.	N.A.	Keying Plug
5	SCOM	Power Supply	Servo Common
6	V5+	Power Supply	+5VDC
7	LCOM	Power Supply	Logic Common
8	CCOM	Drive	Chassis Common
9	M.N.C.	N.A.	
10	V24+	Power Supply	+24VDC
11	224-	Power Supply	-24VDC
12	M.N.C.	N.A.	Keying Plug
13	SCOM	Power Supply	Servo Common
14	V5+	Power Supply	+5VDC
15	LCOM	Power Supply	Logic Common
16	CCOM	Drive	Chassis Common

Notes:

1. Servo, Logic, and Chassis Common are tied together and to ground at one point in the power supply.

2. Chassis Common is the physical case of the drive, Servo Common returns the +/-VDC currents and Logic Common returns +5VDC.

3. Mating Connector Part Numbers:
   AMP Number 2-86256-2, connector (1 each)
   AMP Number 86016-4, contacts (16 each)
   AMP Number 86286-1, key plugs (3 each)
   or
   Cannon Number 121-7326-10843, connector (1 each)
   Cannon Number 11-0238-0091, contacts (13 each)
   Cannon Number 225-7301-003, key plugs (3 each)

4. M.N.C. Make no external connections.

Table 6-E

PIN #	SIGNAL	FROM	COMMENTS
1	V24+	Power Supply	+24VDC
2	V5+	Power Supply	+5VDC
3	LCOM	Power Supply	Logic Common
4	V24-	Power Supply	-24VDC
5	M.N.C.	N.A.	KEY
6	LCOM	Power Supply	Logic Common

Notes:

1. M.N.C. Make no external connection.

2. Logic Common is tied to corresponding signal at one place (where other commons are tied together) in the power supply.

3. Mating connector Part Numbers:
   Cannon Number: 121-7326-702, connector (1 each)
   Cannon Number: 11-0238-0091, contacts (5 each)
   Cannon Number: 225-7301-003, key (1 each)

6.13.2 Signal Pin Assignments

Table 6-F

Pin #	Signal	From	Comments
2	SLD-	Drive	Selected
4	RDY-	Drive	Ready
6	WND-	Drive	Write Enabled
8	FLG-	Drive	Flag
10	LPS-	Drive	Load Point Sensed
12	FUP-	Drive	File Unprotected
14	BSY-	Drive	Busy
16	EWS-	Drive	Early Warning Sensed
18	RWD-	Controller	Rewind
20	REV-	Controller	Reverse
22	FWD-	Controller	Forward
24	HSP-	Controller	High Speed
26	WEN-	Controller	Write Enable
28	SL1-	Controller	Unit Select 2/0
32	SL4-	Controller	Unit Select 2/2
34	SLG-	Controller	Select Gate
36	RNZ-	Drive	Read NRZ Data
38	RDS-	Drive	Read Data Strobe
40	DAD-	Drive	Data Detected
42	WDE-	Controller	Write Data Enabled
44	WNZ-	Controller	Write NRZ Data
46	TR2-	Controller	Track Select 2/1
48	WDS-	Drive	Write Data Strobe
50	TR1-	Controller	Track Select 2/0

Notes:

1. All odd numbered pins are returns.

2. Mating connector is 3M Number 3425-3000 or equivalent.

CHAPTER 7

IEEE 488 - 1975/78 INTERFACE BUS

7.1 INTRODUCTION

The IEEE 488 General Purpose Interface Bus (GPIB) is a hardware option for the 810-104-00x I/O board on a WICAT System 150, 155, or 160. WICAT implements the IEEE-488 option by interfacing a Direct Memory Access chip and the Texas Instruments TMS 9914 chip to the 488 GPIB and the multibus. The interfacing information in this chapter is pertinent to writing a device driver for the IEEE 488 and derives largely from the TMS 9914 GPIB Adapter Preliminary Data Manual, published in 1979 by Texas Instruments, Incorporated.

The TMS 9914 performs the interface between the IEEE 488 and the multibus. It communicates with the multibus via an I/O mapped 8-bit data bus and provides a 16-bit bus to interface with the IEEE 488 via buffers.

7.2 IEEE-488 STANDARDS

To understand the information in this manual, you must familiarize yourself with the terminology and abbreviations explained in the chapter's appendix B. All acronyms employ the IEEE convention of lower case for local messages and upper case for remote messages (received via the interface).

7.3 IEEE 488 BUS CONFIGURATION

The bus is a shielded cable comprising

- o Eight lines to carry data
- o Eight control (= bus management) lines

o Eight lines as signal and system grounds

7.4 GPIB CONNECTOR

Using the standard 24-pin GPIB connector you can connect external peripheral devices into a system (see Appendix A to this chapter). The devices must conform to IEEE Standard #488-1975, which describes a byte-serial, bit-parallel interface system for a programmable measuring apparatus. The cable attached to the GPIB connector must be no longer than 20 meters with no more than 15 peripheral devices connected at one time.

7.5 THEORY OF OPERATION

The GPIB comprises 16 active signal lines and is functionally divided into three component busses:

1. An eight-line **data bus**
2. A three-line **transfer bus**
3. A five-line **management bus**

The transfer rate over the data bus is a function of the slowest peripheral device taking part in a transfer at any one time. The bus operates asynchronously with a maximum transfer rate of 250 Kbytes per second. Peripheral addresses and data are sent sequentially over the data bus.

7.5.1 Peripherals (Device Application)

Peripheral devices on the GPIB are either talkers or listeners. A **talker** can send information on the data bus (one talker at a time). A **listener** can receive information sent over the data bus. Up to 14 listeners may participate simultaneously in an data transfer.

Operational Modes

The TMS 9914 handles the IEEE standard protocol automatically in talker, listener, and controller modes. The GPIB allows up to 15 instruments within a localized area to communicate with each other over a common bus. Each device has a unique address, read from external

switches at power-on. Each device responds to its address.

Message Format

The IEEE 488 bus carries three kinds of messages between the interconnected instruments.

1. **Interface Messages:** Interface messages change the configuration of the interface by addressing devices as talkers or listeners and by transferring devices from remote to local, etc. Interface messages must be accompanied by the ATN line in the active state.

2. **Device Dependent Controls:** Not defined in the IEEE standard, these messages are instructions for selecting range and functions.

3. **Data Messages:** Messages to enable readings to be taken, processed, and stored or printed. Data messages are device dependent and thus not defined in the IEEE standard.

Message Protocol

Only the controller in charge can send interface messages. Any one device operating in the talker mode can send device-dependent controls and data messages, which are received by any devices in the listener mode. Only one device in the system can be in control at any one time.

The user can assign devices to the bus as listeners and talkers by sending their unique talk or listen addresses. The user can switch devices between remote and local control. Device data are sent in byte serial bit parallel format. Any single device (the TALKER) may send device data to a number of other devices (LISTENERS).

Interface control information is sent in byte serial bit parallel format. The controller in charge, a device on the bus, may send interface control messages.

Information is received from the IEEE 488 bus and from the internal registers and is combined with the current status of the device to produce the control signals to load registers or handle the handshake or bus management lines.

7.5.2 Data Flow

I/O pins are connected to the IEEE 488 via bus transceivers. Talker and controller outputs generated on the 9914 control the direction of the data flow. The talker and controller signals are routed within the buffers so that the buffers on particular lines are controlled as required by the TMS 9914.

7.5.3 I/O Mapped Registers

Thirteen I/O mapped registers within the TMS 9914 carry out communication between the multibus and the TMS 9914; seven are write registers, and six are read registers. This communication involves passing control data to and getting status information from the device.

Address lines 1, 2, and 3 from the multibus are connected to the register select lines RS0, RS1, and RS2 and determine the particular register selected. External logic decodes the high order address lines, which causes the CE input to the 9914 to be pulled low when any one of eight consecutive addresses is selected. Thus, the internal registers appear to be situated at eight consecutive locations within the multibus I/O space. Reading from or writing to these locations transfers information between the TMS 9914 and the multibus. The respective on board addresses are shown in table 7-A (next section). Because the registers are read only or write only, reading from or writing to the same location will not access the same register within the 9914.

7.5.4 Addresses

Each device on the IEEE 488 is given a five-bit address enabling the device to be addressed as a talker or listener. You must set this address on an external DIP switch before power-on. The microprocessor reads this address and writes it to the address register as part of the initialization procedure. The TMS 9914 responds by causing a My Address interrupt and entering the required addressed state when this address is detected on the GPIB data lines.

7.5.5 Asynchronous Communication

Three control lines operate as a handshake between a source and an acceptor. Typically, the source is a talker (controller), and the acceptor is a listener. To adjust the data rate to the slowest active listener, the bus delays sending new data until each device addressed to listen has received the previous byte and is ready for the next. Asynchronous communication also ensures compatibility over a wide range of devices.

7.6 TMS 9914 ARCHITECTURE

Access each of the thirteen registers by putting the relevant address on lines RS0, RS1, and RS2 and by performing a memory read (WE=1 DBIN=1) or memory write (WE=0 DBIN=0) operation. Table 7-A shows the register addresses and use of each bit for the read registers; table 7-B does so for the write registers.

Table 7-A
9914 Read Registers

Address			Register Name	Contents							
RS0	RS1	RS2		D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	INT Status 0	INT0	INT1	B1	B0	END	SPAS	RLS	MAC
0	0	1	INT Status 1	GET	ERR	UCG	APT	DCAS	MA	SRQ	IFC
0	1	0	Address Status	REM	ATN	LLO	LPAS	TPAS	LADS	TADS	ulpa
0	1	1	Bus Status	ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN
1	0	0	Address Switch 1	edpa	dal	dat	A5	A4	A3	A2	A1
1	1	0	CMD Pass Through	DI88	DI07	DI06	DI05	DI04	DI03	DI02	DI01
1	1	1	Data In	DI08	DI07	DI06	DI05	DI04	DI03	DI02	DI01

NOTE

The address switch register is external to the 9914 and consists of a DIP switch.

Table 7-B
9914 Write Registers

Address			Register Name	CONTENTS							
RS0	RS1	RS2		D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	INT Mask 0	X	X	BI	BO	END	SPAS	RLC	MAC
0	0	1	INT Mask 1	GET	ERR	UCG	APT	DCAS	MA	SRQ	IFC
0	1	1	Auxiliary CMD	C/S	X	X	f4	f3	f2	f1	f0
1	0	0	Address Reg	edpa	dal	dat	A5	A4	A3	A2	A1
1	0	1	Serial Poll	28	RSV	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

Full descriptions of these registers follow in this chapter. Information is received from the IEEE 488 bus and from the internal registers and is combined with the current status of the device (e.g., Talker Active State, TACS) to produce the control signals to load registers or to handle the handshake or bus management lines.

7.7 REGISTERS

This section describes each register of the TMS 9914.

Read Only Interrupt Status Registers 0 and 1

D7	D6	D5	D4	D3	D2	D1	D0	
INT0	INT1	B1	B0	END	SPAS	RLC	MAC	INT REG 0
GET	ERR	UCG	APT	DCAS	MA	SRQ	IFC	INT REG 1
								Multibus

INT0 An interrupt has occurred in register 0

INT1 An interrupt has occurred in register 1

B1 The 9914 has received a data byte

B0 TMS 9914 is ready to accept the next (or first) data byte

END	An EOI has occurred with ATN = 0
IFC	The 9914 had received an interface clear
RLC	A remote/local state change has occurred
GET	9914 has received a group execute trigger command
ERR	An incomplete source handshake error
UCG	An unidentified command (also set on reception of secondary commands when the pass through next secondary (pts) feature is used) has occurred
APT	A secondary address has occurred in extended addressing mode
DCAS	Device clear active state has occurred
MA	My address (MLA or MTA) and SPMS
SRQ	A service request has occurred and the TMS 9914 is the controller in charge
SPAS	A serial poll active state has occurred with rsv set in the serial poll register

Except for INT0 and INT1, each interrupt bit is set when the corresponding events occur, regardless of the state of the respective mask bit. INT0 and INT1, however, are set only when at least one event occurs in status registers 0 or 1 and the correspnding bit in the interrupt mask register is also set.

For example:

INT0 = (B1(S) AND B1(M) OR (B0(S) AND B0(M)) OR (END(S) AND END(M)) OR (SPAS(S) AND SPAS(M)) etc.

B1(S) is the interrupt status bit B1

B1(M) is the interrupt mask bit B1

Reading each interrupt status register also clears the register. The INT0 and INT1 bits (and therefore the INT line, which is the logic nor of INT0 and INT1) are cleared only when the register causing the interrupt is read.

Hardware and software reset clears the read only interrupt status registers 0 and 1.

NOTE

The ERR bit is not implemented on TMS 9914-P devices, and the UCG is implemented as unidentified universal command (UUCG) and unidentified address command (UACG) with bits one and two respectively.

The byte for interrupt register command is diagramed below:

GET	UUCG	UACG	APT	DCAS	MA	SRQ	IFC	INT REG 1
D7	D6	D5	D4	D3	D2	D1	D0	Multibus

Write Only Interrupt Mask Registers 0 and 1

The Write Only Interrupt Mask Register looks like this:

XX	XX	B1	B0	END	SPAS	RLC	MAC	INT MASK 0
GET	ERR	ICG	APT	DCAS	MA	SRQ	IFC	INT MASK 1
D7	D6	D5	D4	D3	D2	D1	D0	Multibus

B1 Enable interrupt on byte input

B0 Enable interrupt on byte output

END Enable interrupt on EOI with ATN false

SPAS Enable interrupt on serial poll active state

RLC Enable interrupt on remote/local change

MAC Enable interrupt on my address change

GET Enable interrupt on group execute trigger

ERR Enable interrupt on incomplete source handshake

UCG Enable interrupt on unidentified command

APT Enable interrupt on address pass through

DCAS Enable interrupt on device clear active state

MA Enable interrupt on my address

SQR Enable interrupt on service request

IFC Enable interrupt on interface clear

Before an event can send an interrupt to the multibus (multibus level 3, 68000 level 4), you must set the respective mask bit to one. The mask bits remain set until a multibus write operation alters them.

GET, ERR, UCG, APT, DCAS, and MA cause an accept data state (ACDS) holdoff if they are unmasked. Consequently, the multibus can respond to the interrupt, recognize the cause by reading the status register, and take appropriate action before loading the release ACDS holdoff auxiliary command, which would complete the handshake.

Read Only Address Status Register

The Read Only Address Status Register Byte looks like this:

REM	LLO	ATN	LPAS	TPAS	LADS	TADS	ULPA
D7	D6	D5	D4	D3	D2	D1	D0

Multibus

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 TMS 9914 is in the listener primary addressed state

TPAS TMS 9914 is in the talker primary addressed state

LADS (or LACS) The device is addressed to listen

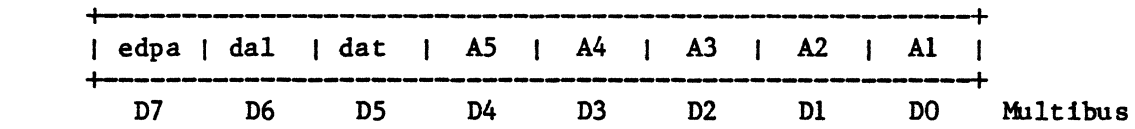
TADS (or TACS) The device is addressed to talk

ulpa This bit shows the LSB of the last address recognized by the 9914

The address status register is not a storage register in the true sense, because the data are obtained from TMS 9914 internal logic at access time.

Write Only Address Register

A diagram of the Write Only Address Byte follows:



edpa Enable dual primary addressing mode

dal Disable listen function

dat Disable talk functions

A5-A1 Device primary address

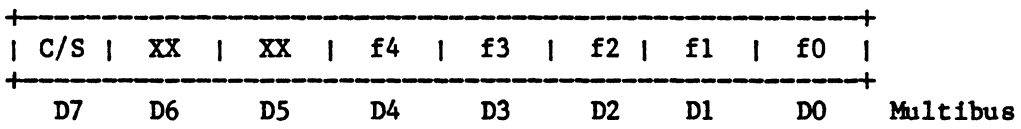
Bits A1 to A5 of the write only address register contain the primary address of the device. A power up/reset or swrst command with C/S = 1 (see auxiliary command register) renders the 9914 idle. The address switch register (typically an external DIP switch) is then read, and the contents are written into the address register.

The edpa bit enables the dual addressing mode of the 9914. Then the address comparator ignores the least significant address bit and gives two consecutive addresses to which the device responds. Inspect ulpa (in the address status register) to obtain the status of edpa

Bits dat and dal disable the talk and listen functions, respectively. Two separate devices may then use the same address if one needs only to talk and the other only to listen.

Write Only Auxiliary Command Register

The following diagram show the Write Only Auxiliary Command Register byte:



f4-f0 Auxiliary command select

C/S Clear or set operation (where applicable)

Auxiliary commands entail many special features. To carry out a function, the multibus writes data at the location corresponding to the auxiliary command register. To select the required command, load the appropriate byte as shown in table 7-C.

Table 7-C
Auxiliary Commands

C/S	F4	F3	F2	F1	F0	MNEMONIC	FUNCTION
0/1	0	0	0	0	0	swrst	Chip reset
0/1	0	0	0	0	1	dacr	Release ACDS 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 EOI ONLY
na	0	0	1	0	1	nbaf	Set new byte available false
0/1	0	0	1	1	0	fget	Force group execute trigger
0/1	0	0	1	1	1	rtl	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	1	0	stdl	Set T1 delay
0/1	1	0	1	1	0	shdw	Shadow handshake

The auxiliary commands operate in CLEAR/SET mode, pulsed mode, or either CLEAR/SET or pulsed modes. Commands operating in CLEAR/SET mode (e.g., lon and ton) require that a code be loaded with the C/S bit equal to one. The function is selected and remains selected until the code is loaded with the C/S bit equal to zero.

Bits shown in table 7-C as CLEAR/SET N/A have a pulsed mode of operation. Force EOI and release RFD holdoff are examples of this.

The force group execute trigger, fget, and the return to local, rtl, commands can operate in either mode. If fget is loaded with C/S equal to zero, a pulse appears at the trigger output of the 9914. If loaded with the C/S bit equal to one, the trigger output goes high until the command is sent again with C/S equal to zero. Similarly, sending rtl with C/S at zero resets the remote/local status bit. A REN command from the controller in charge can set the remote/-local status bit at any time. Sending rtl with C/S set to one clears the REN bit; the REN bit cannot be set again until rtl is sent with C/S at zero. In local lockout (LLO) mode rtl is ineffectual.

Auxiliary Command Definitions

The 23 auxiliary commands are listed here by function, mnemonic, byte configuration, and definition.

1. Chip Reset (swrst) 0/1 xx00000

   Writing swrst with the CLEAR/SET bit equal to one causes all inputs to be ignored and renders the 9914 chip idle. Only the interrupt status registers are cleared and the internal states of the chip are set to the following conditions:

SIDS	source idle state	CIDS	controller idle state
AIDS	acceptor idle state	LOCS	local state
TIDS	talker idle state	NPRS	negative poll response state
TPIS	talker primary idle	PPIS	parallel poll idle state
LIDS	listener idle state	SPIS	parallel poll idle state
LPIS	listener primary state		

   Following a hardware reset, swrst becomes active. You can then configure the 9914 as required, but the chip is held idle until writing the command clears the software reset, with the C/S equal to zero.

2. **Release DAC Holdoff (dacr) 0/1 XX00001**

   The Data Accepted (DAC) holdoff allows time for the host processor to respond to unrecognized commands, secondary addresses, and device trigger or device clear commands. The multibus releases the holdoff when the required action has been taken. Normally **dacr** 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 secondary address was valid, or to zero if invalid (see section on secondary addressing).

3. **Release RFD Holdoff (rhdf) na xx00010**

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

4. **Holdoff on all Data (hdfa) 0/1 xx00011**

   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 multibus using **rhdf**.

5. **Holdoff on End (hdfe) 0/1 xx00100**

   An RFD holdoff occurs when an end of data string message (EOI true with ATN false) is received over the interface. Use **rhdf** to release this holdoff.

6. **Set New Byte Available False (nbaf) na xx00101**

   If a talker is interrupted before the byte just stored in the data out register is sent over the interface, the stored byte is normally sent as soon as the ATN line is false again. If, as a result of the interrupt, this byte is no longer required, you can use **nbaf** to suppress sending the byte.

7. **Force Group Execute Trigger (fget) 0/1 xx00110**

   fget affects the state of the trigger output from the 9914.

If the C/S bit is zero, the line is pulsed high for approximately five clock cycles (1 uS at 5 MHz). If C/S is one, the trigger line goes high until fget is sent with C/S equal to zero. No interrupts or handshakes are initiated.

8. **Return to Local (rtl) 0/1 xx00111**

Provided the LLO has not been enabled, the remote/local status bit is reset and an interrupt is generated (if enabled) to inform the host processor 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 can return to remote without first clearing rtl.

9. **Force End or Identify (feoi) na xx01000**

feoi causes the EOI message to be sent with the next data byte. Then, the EOI line is reset.

10. **Listen Only (lon) 0/1 xx01001**

The listener state is activated until lon is sent with C/S set to zero.

11. **Talk Only (ton) 0/1 xx01010**

The talker state is activated until ton is sent with C/S set to zero.

NOTE

ton and lon are included for use in systems without a controller. A TMS 9914 used as a controller uses ton and lon to set itself up as a listener or talker respectively. Take care therefore to ensure that ton and lon are reset if sending UNL or OTA.

12. **Go To Standby (gts) na xx01011**

    The controller in charge issues **gts** to set the ATN line false.

13. **Take Control Synchronously (tcs) na xx01101**

    The controller in charge uses **tcs** to set the attention line true and so to gain control of the interface. If the controller is not a true listener, the shadow handshake command must be used to monitor the handshake lines so that the 9914 is in synchrony with the talker/listeners and sends ATN true only at the end of byte transfer. This ensures integrity of the data.

14. **Take Control Asynchronously (tca) na xx01100**

    The controller in charge takes control and ATN is asserted. **tca** is executed immediately, and transferring a data byte may corrupt or lose data.

15. **Request Parallel Poll (rpp) 0/1 xx01110**

    The controller in charge executes **rpp** to send the parallel poll command over the interface (in controller active state, the 9914 can assert the attention line). Reading the command pass through register to obtain the status bits and then sending **rpp** with the C/S bit set to zero completes the poll.

16. **Send Interface Clear (sic) 0/1 xx01111**

    The IFC line is set true when **sic** is sent with C/S set to one. **sic** must be sent only by the system controller and is reset (C/S = 0) after the IEEE 488 minimum time for IFC has elapsed (100 uS). The controller enters the controller active state.

17. **Send Remote Enable (sre) 0/1 xx10000**

    The system controller issues **sre** to set the REN line true and to send the remote enable message over the interface. To set

REN false, send **sre** with C/S at zero.

18. **Request Control (rqc) na xx10001**

When the TCT command has been recognized via the unidentified command pass through, the MPU sends **rqc**. The 9914 waits for the ATN line to go false and then enters the controller active state (CACS).

19. **Release Control (rlc) na xx10010**

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

20. **Disable All Interrupts (dai) 0/1 xx10011**

The INT line is disabled but the interrupt registers and any holdoffs selected are unaffected.

21. **Pass Through Next Secondary (pts) na xx10100**

Use **pts** to remote configure a parallel poll. The multibus identifies the parallel poll configure command (PPC), which passes through the 9914 as an unrecognized addressed command. **pts** is loaded, and the next byte received by the 9914 is passed through via the command pass through register. This pass entails the parallel poll enable (PPE), which is read by the multibus.

22. **Set T1 Delay (stdl) 0/1 xx10101**

The T1 delay time can be set to six clock cycles (1.2 uS at 5 MHz) if **stdl** is sent with the C/S bit at one. The T1 delay time is ten clock cycles (2 uS at 5 MHz) following a power on RESET, or if **stdl** is sent with C/S set to zero.

23. **Shadow Handshake (shdw) 0/1 xx10110**

Using **shdw**, the controller in charge can carry out the

listener handshake without participating in a data transfer. The data accepted line 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 romoved.

shdw allows **tcs** to be synchronized with the acceptor not ready state (ANRS) so that ATN can be asserted again without a data byte being lost or corrupted. The END interrupt can also be received and causes an ACDS to be generated.

Read Only Bus Status Register

The Read Only Bus Status Register Byte is diagrammed as follows:

ATN	DAV	NDAC	NRFD	EOI	SRO	IFO	REN	
D7	D6	D5	D4	D3	D2	D1	D0	Multibus

By reading the 488 bus status register, the multibus can ascertain the status of the IEEE 488 bus management lines. Information concerning this register is not obtained from storage; at read time, the information is obtained from the internal logic of the TMS 9914.

Write Only Serial Poll Register

The Write Only Serial Poll Register Byte looks like this:

S8	RSV	S6	S5	S4	S3	S2	S1	
D108	D107	D106	F105	D104	D103	D102	D101	GPIB
D7	D6	D5	D4	D3	D2	D1	D0	Multibus

This register contains the byte sent out when the controller in charge polls a device. A hardware RESET clears this register by loading the auxiliary command register with **swrst**.

Bits S1 to S6 and S8 contain device-dependent information, while bit S7 contains the service request bit. When bit 7 is set, the SRQ line becomes true. When the controller responds by polling serially a device, the SRQ returns to the passive false state

automatically, but after the service is performed, the MPU must clear the **rsv** bit. No subsequent request for service can be made until the **rsv** bit has been reset.

Read Only Command Pass Through Register

Diagrammed, the bits layout for the Read Only Command Pass Through Register looks like this:

D108	D107	D106	D105	D104	D103	D102	D101	GPIB
D7	D6	D5	D4	D3	D2	D1	D0	Multibus

No storage is contained in this register. The IEEE 488 bus is connected to the multibus as shown when an unrecognized command or a secondary address occurs. The UUCG, UACG, and APT functions are selected by means of the appropriate bits in the interrupt mask register.

Write Only Parallel Poll Register

The byte for the Write Only Parallel Poll Register is diagrammed as follows:

PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1	
D108	D107	D106	D105	D104	D103	D102	D101	GPIB
D7	D6	D5	D4	D3	D2	D1	D0	Multibus

When the controller in charge conducts a parallel poll, the contents of the parallel poll register are output. Before the polling, the multibus must load the register. Loading usually occurs during initialization. Use the pass through next secondary feature under software control to configure this register remotely. A hardware (not software) RESET clears this register by loading the register with **swrst**.

Read Only Data In Register

The following is a diagram of the Read Only Data In Register byte:

D108	D107	D106	D105	D104	D103	D102	D103	GPIB
D7	D6	D5	D4	D3	D2	D1	D0	Multibus

Addressed as a listener, the data in register transfers the data frm the GPIB to the multibus. A byte input interrupt is produced and the NRFD line is held true until the multibus reads this register. Unless a holdoff has been selected (holdoff on all data, **hdfa**, or holdoff on end, **hdfe**), the acceptor handshake function is completed when the 9914 releases the NRFD line. In the case of a ready for data (RFD) holdoff, the multibus must complete the handshake using a release holdoff, **rhdf**, auxiliary command. Reading the data in register has no effect on the data out register.

Write Only Data Out Register

The Write Only Data Out Register byte looks like this:

D108	D107	D106	D105	D104	D103	D102	D101	GPIB
D7	D6	D5	D4	D3	D2	D1	D0	Multibus

When you are operating in the talker or controller modes, use the data out register to transfer data bytes or command bytes from the multibus to the IEEE 488 bus. If the 9914 chip has previously been placed in controller active state, commands sent are accompanied by the ATN line held active true. In the talker active state (device dependent data) ATN is held false. In either of these active states the 9914 automatically carries out the source handshake. Writing into the data out register has no effect on the data in register.

A byte out (BO) interrupt when the device enters the talker mode provokes the loading of the first byte. When the byte has been sent over the GPIB, a BO interrupt is given, and the multibus can load a new byte.

After a byte has been loaded into the data out register, but before the byte has been sent over the bus, the ATN line may be taken true by the controller in charge. The controller sends

this byte immediately after the ATN line goes false unless the multibus gives the new byte available false command, nbaf.

7.8 TERMINAL ASSIGNMENTS AND FUNCTIONS

The 9914 chip has terminations on one side for the IEEE 488 GPIB lines and terminations on the other side for multibus lines. The IEEE 488 standard uses the negative logic convention for GPIB lines. The false state (0) is represented by a high voltage (>2.0V); the true state (1) is represented by a low voltage (<0.8V). The GPIB terminations of the 9914 chip agree with this negative logic convention. If, for example, data valid is true (1), the device pulls the DAV line low. These terminations, to obtain the correct signal parity, are connected to the bus via noninverting buffers.

On the multibus side of the device the terminations are in positive logic (true state (1) = high voltage: false state (0) = low voltage). This arrangement agrees with the conventional logic used by the multibus. Thus, if:

D0(MSB)							D7(LSB)
0	1	1	0	1	0	0	1

is written into the data out register, it appears on the IEEE 488 DIO lines as:

D108(MSB)							D101(LSB)
high	low	low	high	low	high	high	low

7.9 HARDWARE RESET

Following a hardware reset, usually at power-on, the controller sets **swrst** and clears the parallel poll and interrupt status registers, and the auxiliary commands (except **swrst**). The IEEE 488 state diagram logic is then set hardware reset states (table 7-D) (The final s in each mnemonic stands for state.)

Table 7-D
Hardware Reset States

SIDS	source idle	CIDS	controller idle
AIDS	acceptor idle	LOCS	local
TIDS	talker idle	NPRS	negative poll response
TPIS	talker primary idle	PPIS	parallel poll idle
LIDS	listener idle	SPIS	serial poll idle
LPIS	listener primary		

7.10 CONTROLLER FUNCTION

Entering Controller Mode

The 9914 chip enters the controller mode under local or remote control. If the device is the system controller, enter controller mode by loading the auxiliary command register with **send interface clear** (sic). Loading **sic** with the C/S bit set to zero clears the auxiliary command register after the IEEE 488 minimum time of 100 uS. The 9914 enters the controller active state automatically when an IFC has been sent to the other instruments on the bus.

With the 9914 in talker active state, obtain control by issuing the **tct** command. If unmasked, an unrecognized command group (UCG) interrupt occurs with an accept data state (ACDS) holdoff. The multibus then reads the command from the **cpt** register, recognizes it, and loads the request control command, before releasing the holdoff, with the auxiliary command **dacr**. The device becomes controller in charge when the previous controller releases the ATN line.

Sending Commands

While in the controller mode, send commands by loading them into the data out register. The 9914 chip completes the handshakes automatically by using a byte generated by a BO interrupt. The ATN line is asserted until **rlc** or **gts** put the 9914 chip in CIDS or controller standby state.

NOTE

While in controller mode, the 9914 chip does not monitor the commands it is sending.

Table 7-E lists addressed and universal commands that you can issue while you operate in controller mode.

Table 7-E
Commands Sent in Controller Mode

COMMAND	ACRONYM	DIO LINES 8 7 6 5 4 3 2 1	TYPE	NOTES
Addressed Command Group	ACG	x 0 0 0 x x x x	AC	
Device Clear	DCL	x 0 0 1 0 1 0 0	UC	
Group Execute Trigger	GET	x 0 0 0 1 0 0 0	AC	
Go To Local	GTL	x 0 0 0 0 0 0 1	AC	
Listen Address Group	LAG	x 0 1 x x x x x	AD	
Local Lockout	LLO	x 0 0 1 0 0 0 1	UC	
My Listen Address	MLA	x 0 1 L L L L L	AD	1
My Talk Address	MTA	x 1 0 T T T T T	AD	2
My Secondary Address	MSA	x 1 1 S S S S S	SE	3,4
Other Secondary Address	OSA	TAG & MTA	SE	4,5
Other Talk Address	OTA		AD	
Primary Command Group	PCG		--	6
Parallel Poll Configure	PPC	x 0 0 0 0 1 0 1	AC	7
Parallel Poll Enable	PPE	x 1 1 0 S P P P	SE	8,9
Parallel Poll Disable	PPD	x 1 1 1 D D D D	SE	8,10
Parallel Poll Unconfigure	PPU	x 0 0 1 0 1 0 1	UC	11
Secondary Command Group	SCG	x 1 1 x x x x x	SE	
Selected Device Clear	SDC	x 0 0 0 0 1 0 0	AC	
Serial Poll Disable	SPD	x _ _ 1 1 0 0 1	UC	
Serial Poll Enable	SPE	x 0 0 1 1 0 0 0	UC	
Take Control	TCT	x 0 0 0 1 0 0 1	AC	12
Talk Address Group	TAG	x 1 0 x x x x x	AD	
Universal Address Group	UAG	x 0 0 1 x x x x	UC	
Unlisten	UNL	x 0 1 1 1 1 1 1	AD	
Untalk	UNT	x 1 0 1 1 1 1 1	AD	

7.11 TALKER FUNCTION

Listed here are the events that circumscribe talker active state.

1. The device is talker addressed.
2. The byte out (BO) interrupt prompts the multibus to load the first data byte.
3. The controller is not asserting the ATN line.
4. Data bytes loaded into the data out register are transferred over the bus.
5. The bytes are received by all listeners
6. The 9914 automatically carries out the handshake protocol.
7. If enabled and unmasked, the BO interrupt is generated when each byte has been accepted by all the active listeners on the bus.
8. Generation of a second BO interrupt tells the multibus that the 9914 chip is ready to accept the next data byte.

Local Implementation

To send data or device-dependent control information over the bus, a device must place the 9914 in the talker addressed state. To implement the talker function locally, the multibus loads the auxiliary command register with the ton command. Enabling this function locally normally happens in a system without a controller or when the controller addresses itself to talk.

Diagrammed, the byte for local implementation looks like this:

AUXILIARY COMMAND REGISTER

1	x	x	0	1	0	1	0	ton

D7 D0

[C/S is one to set the function]

GPIB Lines

x	1	0	1	1	1	1	1	UNTALK

D108 D101

Remote Implementation

Remote enabling of the talker function occurs when the controller in charge places on the system bus my talk address (MTA). (See secondary addressing section if APT mask bit is set.) The remote enabling byte looks like this:

GPIB Lines

x	1	0	T	T	T	T	T	MTA
D108							D101	

7.12 LISTENER FUNCTION

The following events circumscribe the listener addressed state:

1. The 9914 carries out the handshake protocol with the talker and generates a byte interrupt (BI) when the data byte is received.

2. The NRFD line is held true on the bus until the multibus has read the byte in register, or, until you have issued the rhdf command.

At any one time, the system can accommodate a number of active listeners. Enabling the listener function, as described in this section, affects only the device in question, not all devices on the system.

Local Implementation

To place the 9914 in the listener addressed state (LADS), load the auxiliary command register with the lon command. Enabling this function locally normally occurs in a system without a controller or when the controller addresses itself to listen. Diagrammed, the bits layout for local implementation of lon looks like this:

AUXILIARY COMMAND REGISTER

1	x	x	0	1	0	0	1	lon

Alternatively, the listener function is enabled when the

controller in charge sends **my listen address** (MLA) for the device. (If APT mask is set, read about enabling secondary addressing in the next section.) Here is the GPIB line bit diagram of this alternative for enabling the listener function:

GPIB Lines								
x	0	1	L	L	L	L	L	MLA

Clearing Listeners

Because simultaneous active listeners are possible, you must clear a given listener if that listener's device is not required to receive the data. To clear a listener, use the **unlisten command** (UNL). A diagram of the bits for UNL follows:

GPIB Lines								
x	0	1	1	1	1	1	1	UNL

7.13 ADDRESSING

The 9914 can operate with primary or secondary addressing.

Primary Addressing

If the address pass through (APT) bit in the interrupt mask register is set to zero, the 9914 responds to primary addresses only. The controller enters listener (LADS) or talker (TADS) state immediately when the **my address** command occurs on the bus. Either state is maintained until the **unlisten, untalk,** or **other talk address** command occurs. Mixing of primary and secondary addressing is disallowed. The primary addressed state is cleared when a new primary command is received by the device.

Secondary Addressing

When secondary addressing is required, the APT mask bit is set to one during initialization. When **my address** is received, the 9914 chip enters the **primary addressed state** (i.e., TPAS or LPAS) and refrains from entering LADS or TADS. If the next command received by the 9914 is the secondary address, an APT interrupt and an ACDS holdoff are caused. The multibus then reads the

address from the command pass through register. If the multibus recognizes the secondary address as valid for the device, it sends dacr with the C/S bit set to release the ACDS holdoff. The 9914 then enters LADS or TADS according to which primary address was originally received.

If the secondary address applies to another device (i.e., is invalid), the multibus loads dacr with the C/S bit set to zero. The 9914 chip then completes the handshake, but remains in the primary addressed state until it receives another primary command. Thus, further secondary addresses can be received without the need for the 9914 to reassert the primary address.

7.14 HANDLING INTERRUPTS

Sixteen events can cause an interrupt. Each event has a corresponding bit in one of the interrupt registers, as diagrammed here:

INT0	INT1	BI	BO	END	SPAS	RLC	MAC	INT REG 0
GET	ERR	UCG	APT	DCAS	MA	SRQ	IFC	INT REG 1

When one of these events occurs, the corresponding bit in one of these registers is set. Furthermore, if interrupts are enabled and the corresponding mask bit is also set, a one is written to either INT0 or INT1 (depending on which register is affected), and the INT line to the multibus is pulled low. The multibus's response to the interrupt entails first reading the contents of interrupt registers 0 and 1 to find the cause, then carrying out the appropriate service routine. The registers are cleared when read.

If reading an interrupt register coincides with an interrupt, the associated bit is stored. When the read cycle completes, the bit is introduced into the corresponding register.

Disable Interrupts

If a polling system is required in which the status registers are periodically read by the multibus, the interrupts are not used. You can disable interrupts without affecting the holdoffs on unrecognized addresses or commands. Do so by loading the disable all interrupts (dai) command into the auxiliary command register. Here is the byte diagram for dai.

AUXILIARY COMMAND REGISTER

1	x	x	1	0	1	0	0	dai
D7							D0	

The interrupt status bits are still set, but the INT line is held high regardless of the value in the mask register.

7.15 SERVICE REQUESTS

The SRQ function enables the instrument to carry out a task autonomously and signal the controller when further action is needed. Writing a one into the **rsv** bit in the serial poll register sets the service request output from the 9914. The controller responds with a serial poll on the requesting device; thereafter, the SRQ line is reset to false. The local multibus must reset the **rsv** bit when the service has been carried out. This reset is required before you can request further service.

The remaining seven bits in the serial poll register can be loaded with the code when the SRQ is made. You can choose the format for this and might want to use these bits to indicate the condition of the device and the reason for requesting service.

7.16 GPIB SIGNAL DEFINITIONS

Data Bus. The data bus encompasses eight bidirectional active-low signal lines, D101 through D108. One byte is sent over the bus at a time. D101 is the least significant bit, D108 the most significant bit. Each byte represents either a primary or secondary peripheral address, a control word, or a data byte. Data bytes can be in ASCII format with or without parity, or they can be formatted in machine-dependent binary code.

Management Bus. The management bus is a group of five signal lines used to control data transfers over the data bus. Definitions of the five signals follow:

1. **Attention (ATN)**

   ATN goes active when peripherals are being assigned as listeners and talkers. Only peripheral addresses and control

messages can be sent over the data bus when ATN is active low. After ATN goes high, only listeners and talkers can take part in the data transfer.

2. **Service Request (SRQ)**

   Any peripheral device on the GPIB can request the attention of the controller by setting SRQ active low. The controller responds by setting ATN active low and executing a serial poll to see which device is requesting service.

3. **Interface Clear (IFC)**

   The controller activates IFC whenever the controller wants to place all interface circuitry in a predetermined quiescent state.

4. **Remote Enable (REN)**

   The controller activates REN whenever the system is operating under program control. REN causes all peripherals on GPIB to ignore their front panel controls and operate under remote control via signals and control messages received over the GPIB.

5. **End of Identify (EOI)**

   A talker uses the EOI signal to indicate the end of a data transfer. The talker activates EOI as the last byte is sent. With EOI active, a listening controller assumes that a byte received is the last byte in the transmission. When the controller is talking, it always activates EOI as the last byte is sent.

Transfer Bus. Each time a data byte is sent over the data bus, the talker and the listeners execute a handshake sequence over the transfer bus. Definitions of the transfer bus signal lines follow:

1. **Not Ready for Data (NRFD)**

   An active low NRFD signal line indicates that one or more assigned listeners are not ready to receive the next data byte. When all assigned listeners for a particular data transfer have released NRFD, the NRFD line goes inactive high. This tells the talker to place the next data byte on the data bus.

2. **Data Valid (DAV)**

   The talker activates DAV shortly after the talker places a valid data byte on the data bus. An active low DAV signal tells each listener to capture the byte currently on the data bus. When NRFD is active low, the talker cannot activate DAV.

3. **Date Not Accepted (NDAC)**

   Each listener holds the NDAC signal line active low until the listener captures the data byte. NDAC then goes inactive high, which tells the talker to remove the byte from the data bus.

GPIB Data Formats. Any series of bit patterns (alphanumeric) can be sent over the GPIB.

Transferring ASCII Data. ASCII numeric data can be sent in either standard (free) format or scientific format and you must send the most significant digit first. Valid ASCII characters are zero through nine, E, e, +, -, and decimal point. ASCII character strings can be sent as any sequence of valid ASCII characters. Strike the return key or activate the EOI signal line of the management bus to terminate all ASCII data transfers.

7.17 MEMORY MAP

Table 7-F is a memory map of the addresses of the IEEE 488 associated devices and registers. The base I/O address is F000XX.

Table 7-F
IEEE 488 Memory Map

HEX ADDRESS	CIRCUIT FUNCTION	CMD TYPE	DESCRIPTION
C0	488 Controller	R	Read Interrupt Status on Register 0
		W	Write Interrupt Mask to Register 0
C2		R	Read Interrupt Status on Register 1
		W	Write Interrupt Mask to Register 1
C4		R	Read Address Status Register
		W	
C6		R	Read Bus Status Register
		W	Write Auxiliary Command Register
C8		R	N/A
		W	Write Address Register
CA		R	
		W	Serial Poll Register
CC		R	Command Pass Through Register
		W	Parallel Poll Register
CE		R	Data In Register
		W	Data Out Register
D0	Addressable	R	N/A
	Latch	W	LEDs and Parallel Port Direction
D2	Test Switches	R	Sw1-Sw8 for Test/Configuration
		W	N/A
D4	EOP Clear	R	N/A
		W	
D6		R	Not used
		W	DMA End of Process Mask Set
D8		R	Not used
		W	
DA		R	Not used
		W	
DC		R	Not used
		W	
DE	DMA Extended	R	N/A
	Address	W	Set up ARD10-ADR17 for DMA Transfer
E0	DMA	R	Current Address CHAN 0
	Controller	W	Base and Current Address CHAN 0
E2		R	Current Word Count CHAN 0
		W	Base and Current Word Count CHAN 0
E4		R	Current Address CHAN 1
		W	Base and Current Address CHAN 1
E6		R	Current Word Count CHAN 1
		W	Base and Current Word Count CHAN 1
E8		R	Current Address CHAN 2
		W	Base and Current Address CHAN 2

Table 7-F (continued)

HEX ADDRESS	CIRCUIT FUNCTION	CMD TYPE	DESCRIPTION
EA		R	Current Word Count CHAN 2
		W	Base and Current Word Count CHAN 2
EC		R	Current Address CHAN 3
		W	Base and Current Address CHAN 3
EE		R	Current Word Count CHAN 3
		W	Base and Current Word Count CHAN 3
F0		R	Read Status Register
		W	Write Command Register
F2		R	Illegal
		W	Write Request Register
F4		R	Illegal
		W	Write Single Mask Register Bit
F6		R	Illegal
		W	Write Mode Register
F8		R	Illegal
		W	Clear Byte Pointer F/F
FA		R	Read Temporary Register
		W	Master Clear
FC		R	Illegal
		W	Illegal
FE		R	Illegal
		W	Write All Mask Register Bits

7.18 DIRECT MEMORY ACCESS (DMA)

In order to improve the speed of the transfer when a TMS 9914 is used, a direct memory access controller (DMA) has been added to the circuit. A discussion of the AMD 9517 (DMA) follows.

If a byte input or output occurs, the TMS 9914 sets the ACCRQ low (regardless of whether DMA access is required). If the DMAC responds by setting ACCGR low, it has set up the memory address for the data out register. Alternatively, if DBIN is low, the byte from the data in register is written into memory at the location given by the DMAC.

Note that the TMS 9914 has no chip enable when responding to an Access Granted signal and does not use the register selected lines RS0-RS2.

When the ACCGR is received by the TMS 9914, it resets the ACCRQ line until the next byte input or output occurs. Note that the

DBIN line is inverted in the DMA mode. When ACCGR is low (DMA access) DBIN=0 indicates a memory write from the TMS 9914 outputs to device RAM. When ACCGR is high, a multibus access is occurring, and a DBIN=0 indicates a write to the TMS 9914 registers.

Direct memory access uses the AMD 9517A (DMA) device. WICAT uses this device with the limitations and enhancements discussed here.

7.18.1 DMA Limitations

The WICAT 9517A (DMA) support circuitry does not allow the 9517 to perform memory to memory DMA transfers. The circuitry also does not allow the 9517A to work in the compressed timing mode. The cascade mode is also not used.

The 9517A is a four channel DMA. Our circuit uses the DMA only to support the 9914. This means that only one channel, channel 0, is used. Note that the 9517 needs to be programmed such that DREQ and DACK are sense active low.

7.18.2 DMA Enhancements

The 9517A has 16-bit addressing. This limits its transfers to 64 Kbytes. An additional 8-bit address latch has been added for the upper address lines of the multibus. The on board address of this latch is DE, as shown in table 7-F. Positive true data written into the latch are negative true addresses ARDI0-ARDI7 during a DMA cycle.

When a channel's word count goes to zero, the 9517A pulse EOP goes low to provide the multibus with a completion signal. This End of Process (EOP) interrupt sets its TC bit in the status register and resets its request bit.

It was necessary to add a latch and an enable to the signal in order to use the EOP signal as an interrupt. If EOP is wanted as an interrupt, it is necessary to write 00 (hex) to the latch at on board address D6, as shown in table 7-F. Clear the interrupt by writing to on board address D4, as in table 7-F.

7.18.3 DMA Operation

The 9517 is designed to operate in two major cycles called Idle and Active. Each device cycle is made up of a number of states. The 9517 can take seven separate states, each a full clock period long.

State 1 (S1) is the inactive state. It occurs when the 9517 has no valie DMA requests pending. While in S1, the DMA controller is inactive but may be programed by the multibus to a Program Condition.

State 0 (S0), the first state of DMA service, occurs when the 9517 has requested a hold but the multibus has not yet returned an acknowledge. An acknowledge from the multibus signals that transfers may begin.

S1, S2, S3, and S4 are the working states of the DMA service.

7.18.4 DMA Idle Cycle

When no channel is requesting service, the 9517 enters the idle cycle and performs S1 states. In this cycle, the 9517 samples the DREQ lines every clock cycle to determine if any channel is requesting DMA service.

The device also samples CS for any attempt by the multibus to write or read the internal registers of the 9517. When CS and HACK are low, the 9517 enters Program Condition. The multibus can then establish, change, or inspect the internal definition of the part by reading from or writing to the internal registers.

Address lines A1-A4 are inputs to the device and select which registers are read or written. The IOR and IOW lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip-flop generates an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit address and word count registers. The flip-flop is reset by Master Clear or Reset, or by a software command line.

Special software commands can be executed by the 9517 in the Program Condition. These commands are decoded as sets of addresses when both CS and IOW are active and do not make use of the data bus. Functions include Clear First/Last Flip-Flop and Master Clear.

7.18.5 DMA Active Cycle

When the 9517 is in the Idle cycle and a channel requests a DMA service, the device outputs a HREQ to the multibus and enters the Active cycle. DMA service takes place in one of four modes of the active cycle.

Single Transfer Mode

In single transfer mode, the 9517 makes a one-byte transfer during each HREQ/HACK handshake. When DREQ goes active, HREQ also goes active. After the multibus responds by driving HACK active, a one-byte transfer takes place. HREQ goes inactive following the transfer, the word count is decremented and the address is either incremented or decremented. When the word count goes to zero, a Terminal Count (TC) causes an Autoinitialize if the channel has been programmed to do so.

To perform a single transfer, DREQ must be held active only until the corresponding DACK goes active. If DREQ is held continuously active, HREQ goes inactive following each transfer, then goes active again and a new byte transfer is made following each rising edge of HACK.

Block Transfer Mode

In Block Transfer, the 9517 continues making transfers until a TC (caused by the word count going to zero) or an external End of Process (EOP) is encountered. DREQ need be held active only until DACK becomes active. An autoinitialize occurs at the end of the service if the channel has been programmed for it.

Demand Transfer Mode

In Demand Transfer, the device continues making transfers until a TC or external EOP is encountered or until DREQ goes inactive. Thus, the device requesting service may discontinue transfers by bringing DREQ inactive. Service may be resumed by asserting an active DREQ once again.

During the time between services, when the multibus is allowed to operate, the intermediate values of address and word count may be read from the 9517 Current Address and Current Word Count registers. Autoinitialization only occurs following a TC or EOP at the end of service. Following Autoinitialization, an active-going DREQ edge is

required to initiate a new DMA service.

Cascade Mode

Not used in this design.

7.18.6 DMA Transfer Types

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write, and Verify. Write transfers move data from an I/O device to the memory by activating IOR and MEMW. Read transfers move data from memory to an I/O device by activating MEMR and IOW. Verify transfers are pseudo transfers; the 9517 operates as in Read or Write transfers generating addresses, responding to EOP, etc., however, the memory and I/O control lines remain inactive.

Memory-to-Memory

This function is not possible in the present design.

Autoinitialize

By programming a bit in the Mode register, a channel may be set up for an Autoinitialize operation. During autoinitialization, the original values of the Current Address and Current Work Count registers are automatically restored from the Base Address and Base Word Count registers of that channel following EOP. The base registers are loaded simultaneously with the current registers by the multibus and remain unchanged throughout the DMA service. The mask bit is not set by EOP when the channel is in Autoinitialize. Following Autoinitialize, the channel is ready to repeat its service without multibus intervention.

Priority

The 9517 has two types of priority encoding available as software selectable options. Fixed priority is used in this application.

Compressed Timing

This function is not possible in the present design.

7.18.7 DMA Register Description

Current Address Register

Each channel has a 16-bit Current Address register. It holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. The register is written or read by the multibus in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialization takes place only after an EOP.

Current Word Count Register

Each channel has a 16-bit Current Word Count Register. It should be programmed with, and returns on a multibus read, a value one less than the number of words to be transferred.

The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, a TC is generated.

This register is loaded or read in successive 8-bit bytes by the multibus in Program Condition. Following the end of a DMA service, it may also be initialized by an Autoinitialize back to its original value. Autoinitialize can occur only when an EOP occurs. Note that the contents of the Word Count register is FFFF (hex) following an internally generated EOP.

Base Address and Base Word Count Registers

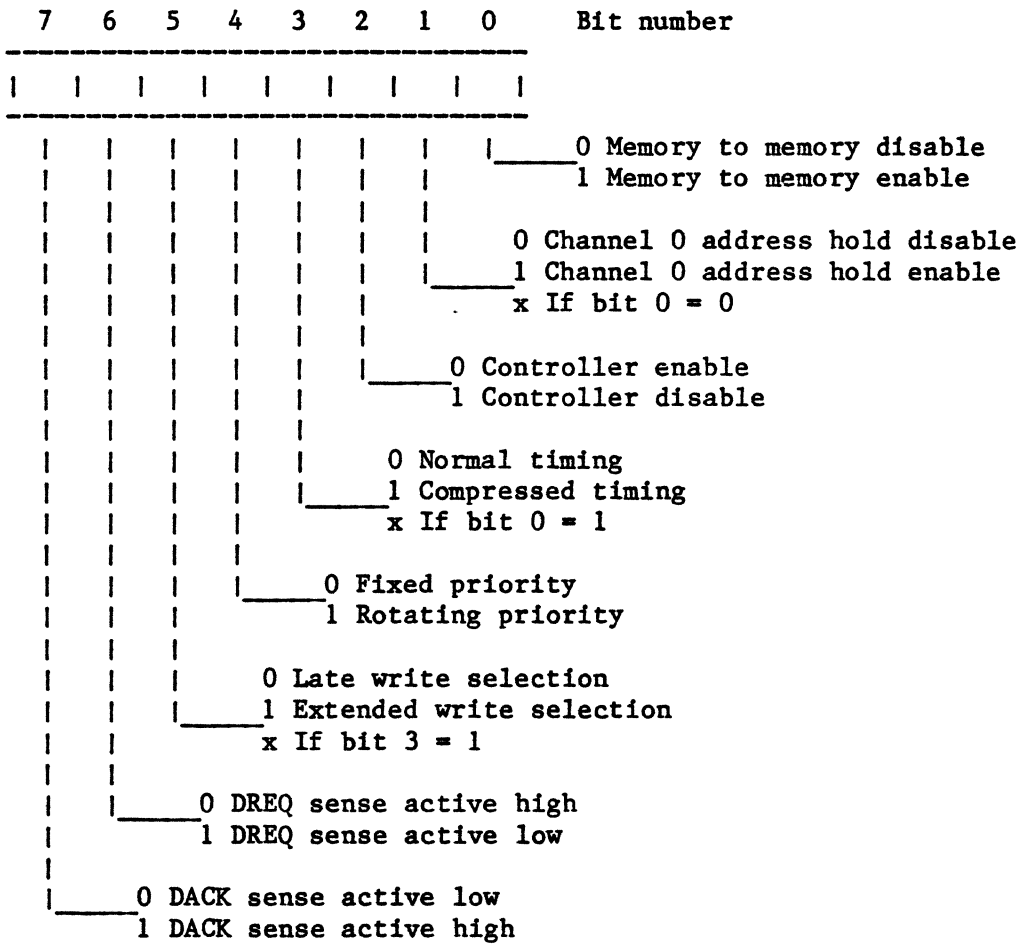
Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original values of their associated current registers. During Autoinitialize, these values are used to restore the current registers to their original values.

The base registers are written simultaneously with their

corresponding current register in 8-bit bytes during DMA programming by the multibus. Accordingly, writing to these registers when intermediate values are in the current registers overwrites the intermediate values. The base registers cannot be read by the multibus.

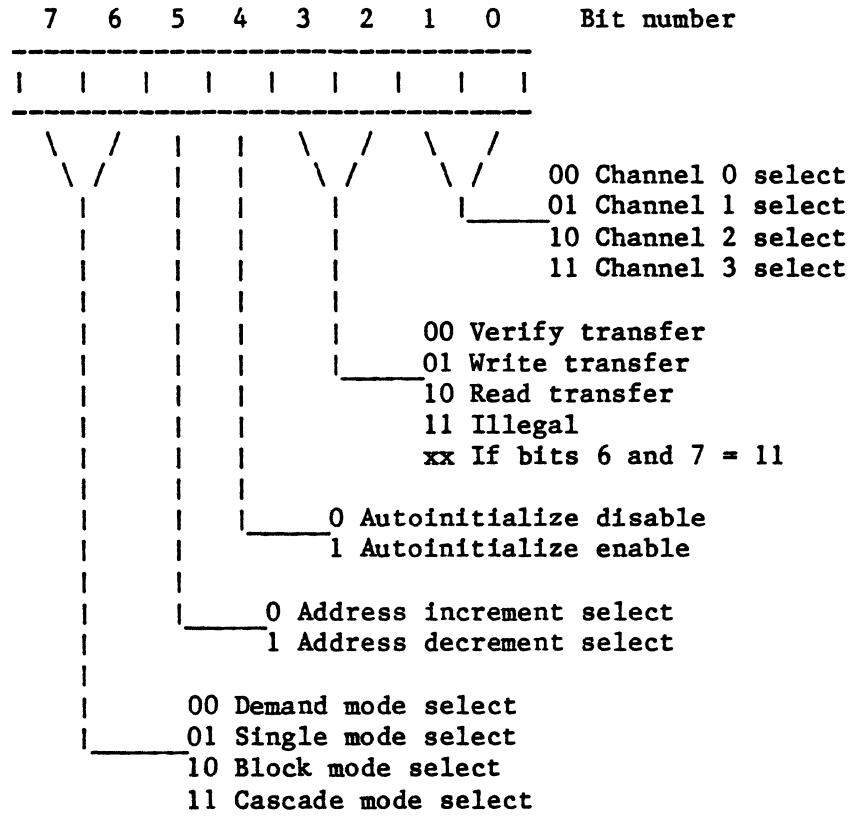
Command Register

This 8-bit register controls the operation of the 9517. It is programmed by the multibus in the Program condition and is cleared by Reset. The following table lists the function of the command bits.



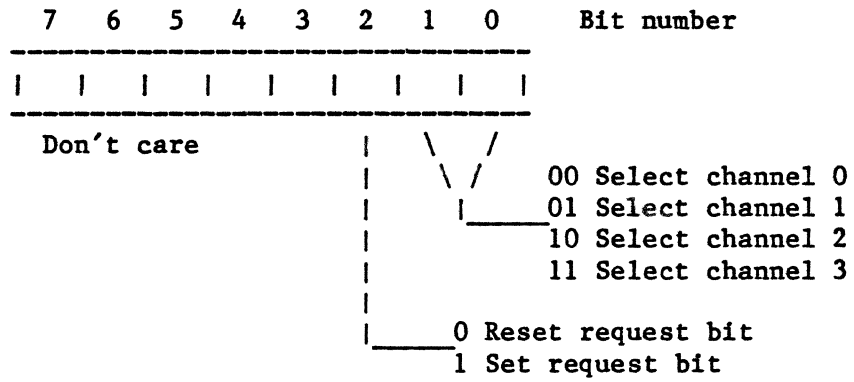
Mode Register

Each channel has a 6-bit mode register associated with it. When the register is being written to by the multibus in Program Condition, bits 0 and 1 determine which channel Mode register is to be written.



Request Register

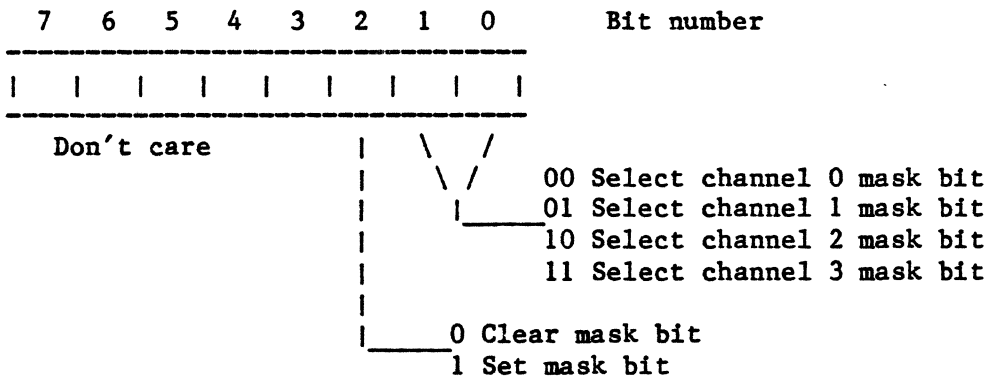
The 9517 can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are nonmaskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external EOP. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word.



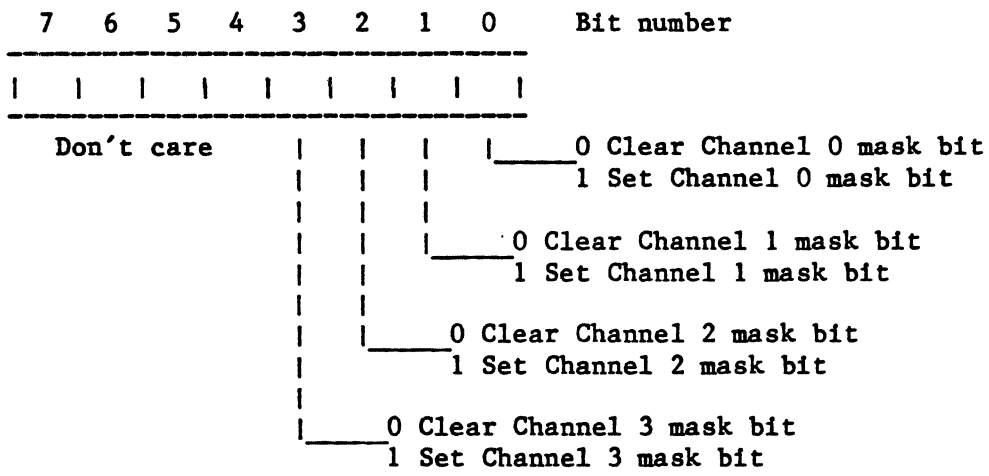
Software requests are serviced only if the channel is in Block mode.

Mask Register

Each channel has a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an EOPO if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register.

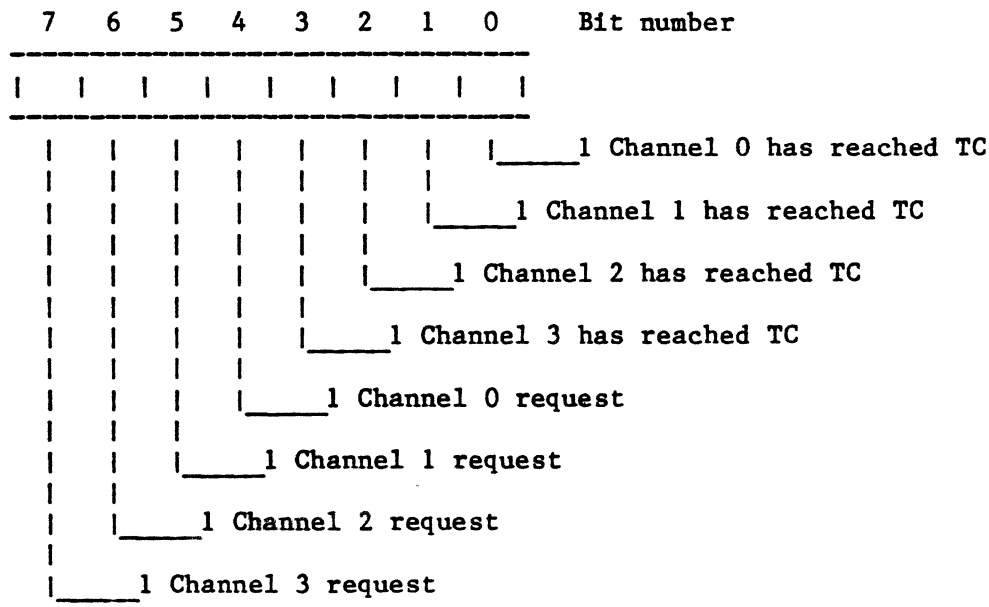


All four bits of the Mask Register may also be written with a single command.



Status Register

The Status register may be read out of the 9517 by the multibus. It indicates which channels have reached a terminal count and which channels have pending DMA requests. Bits 0-3 are set each time a TC is reached by that channel, including after each Autoinitialization. These bits are cleared by Reset and each Status Read. Bits 4-7 are set whenever their corresponding channel is requesting service.



Temporary Register

The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the multibus in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands

There are two special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. They are:

1. **Clear First/Last Flip/Flop**

   This command may be issued prior to writing or reading 9517 address or word count information. This initializes the flip-flop to a known state so subsequent accesses to register contents by the multibus address lower and upper bytes in the correct sequence.

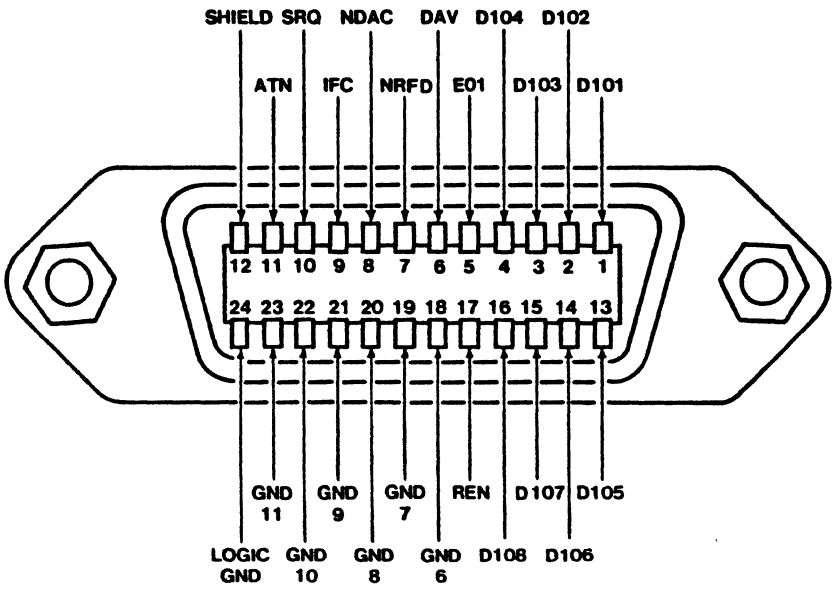
2. **Master Clear**

   This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip/Flop registers are cleared and the Mask register is set. The 9517 enters the Idle cycle.

Table 7-F lists the address codes for the software commands and the register addresses.

7.19 APPENDIX A

THE IEEE 488 INTERFACE CONNECTOR



7.20 APPENDIX B

DEFINITION OF TERMS AND ACRONYMS

ACCGR	Access granted. DMA signal from DMAC to TMS 9914.
ACCRQ	Access Request. DMA signal from TMS 9914 to DMAC.
AIDS	Acceptor Idle State. part of acceptor handshake.
APT	Address Pass Through.
ATN	Attention Line. Part IEEE bus. The Controller In Charge pulls this line true (low) to send commands over the bus.
BI	Byte In. Indication that the TMS 9914 has completed an acceptor handshake.
BO	Byte Out. Indication that the TMS 9914 has completed a source handshake.
CIDS	Controller Idle State. Part of IEEE state diagram.
Controller in Charge	The currently active controller (but not necessarily the system controller).
DAV	Data Valid. IEEE 488 handshake line.
DCAS 3	Device Clear Active State. Part of IEEE state diagram.
DMA	Direct Memory Access.
DMAC	Direct Memory Access Controller.
Dual Addressing	A mode of operation of the TMS 9914 in which it responds to two consecutive addressors.
EOI	End or Identify. IEEE bus line. Signifies end of data block if ATN is false or requests a parallel poll if ATN is true.
END	Status bit set in TMS 9914 when EOI is true with ATN false.

ERR	ERROR. Incomplete source handshake. Typically when 9914 is talker and no listeners are on the bus.
GET	Group Execute Trigger.
GPIB	General Purpose Interface Bus.
Host Processor	The microprocessor controlling the operation of the instrument or device of which the TMS 9914 is part.
IFC	Interface Clear. IEEE bus line. Sets all devices on the bus to a known quiescent state.
LADS	Listener Addressed State. Part of the IEEE state diagram. The TMS 9914 has recognized its listener address on the bus or has been locally addressed to listen via the lon auxiliary command and is ready to carry out an acceptor handshake.
LIDS	Listener Idle State. Part of the IEEE state diagram.
LISTENER	Any device forming part of the GPIB interface that can receive data from the DIO lines.
LLO	Local Lockout. A conditin in which the device should respond to local control (e.g., from the front panel).
LPAS	Listener Primary Addressed Se. Part of the IEEE state diagram.
MA	My Address. The address of a particular instrument or device. Usually defined position of the address switch at power can be either the talker or listener address.
MAC	My Address Change. Status bit with the TMS 9914 that idicates change of addess status.
MPU	Microprocessor. Refers to the processor controlling the instrument or device containing the TMS 9914.
NDAC	Not Data Accepted. On of the three bus handshake lines indicating the source that the current data byte has not been latched

	from the bus.
NRFD	Not Ready For Data. A bus handshake that becomes false when all active listeners are ready to read the next data byte.
NPRS	Negative Poll Response State. Part of the IEEE state diagram.
PPIS	Parallel Poll Idle State. Part of the IEEE state diagram.
PPO Subset	The parallel poll subset into which TMS 9914 powers-up, giving no initial parallel poll capability.
REN	Remote Enable. A IEEE bus line indicating to all devices that they should respond to remote messages until sent back to local control.
RS2-RS0	Register select lines from the least significant microprocessor address lines to the TMS 9914.
SIDS	Source Idle State. Part of the IEEE state diagram.
SPAS	Serial Poll Active State. Part of the IEEE state diagram. The TMS 9914 is taking part in a serial poll.
SPIS	Serial Poll Idle State. Part of the IEEE state diagram. The TMS 9914 is not taking part in a serial poll.
SRQ	Service Request. An IEEE bus line held true by a device requiring service by the Controller In Charge.
System Controller	The instrument or device in control of the overall interface configuration. The only device able to send IFC and REN true on the bus.
TADS	Talker Addressed State. Part of the IEEE state diagram. The TMS 9914 has recognized its talk address from the bus or has been locally addressed to TALK via an auxiliary command.

Talker	Any instrument or device capable of transmitting data bytes over the IEEE 488 bus.
TIDS	Talker Idle State. Part of the IEEE state diagram.
TPAS	Talker Primary Addressed State. Part of the IEEE state diagram.
UCG	Unidentified Command Group.