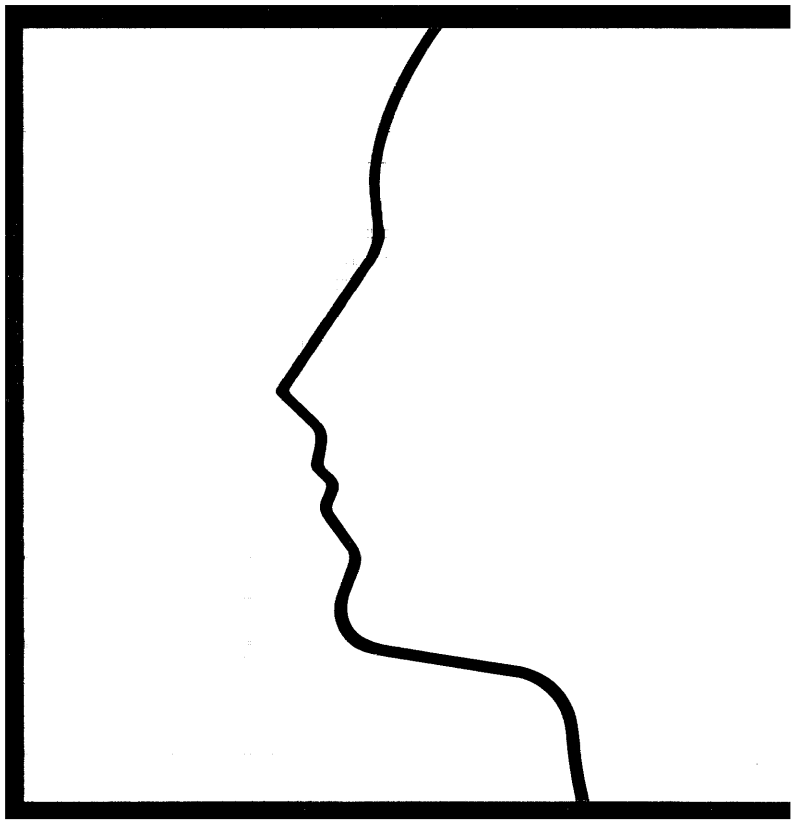
TEXAS INSTRUMENTS

EXPLORER™

SYSTEM INTERFACE

GENERAL DESCRIPTION



WARNING: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

EXPLORER™ SYSTEM INTERFACE GENERAL DESCRIPTION

MANUAL REVISION HISTORY

Explorer™ System Interface General Description (2243145-0001) Rev. A

Original Issue .....February 1985

Revision A .....October 1985

The computers, as well as the programs that TI has created to use with them, are tools that can help people better manage the information used in their business; but tools—including TI computers—cannot replace sound judgment nor make the manager's business decisions.

Consequently, TI cannot warrant that its systems are suitable for any specific customer application. The manager must rely on judgment of what is best for his or her business.

Printed in U.S.A

THE EXPLORER™ SYSTEM SOFTWARE MANUALS

Mastering the Explorer Environment

Explorer Technical Summary	2243189-0001
Explorer Operations Guide	2243190-0001
Explorer Zmacs Editor Tutorial	2243191-0001
Explorer Glossary	2243134-0001
Explorer Communications User's Guide	2243206-0001
Explorer Diagnostics	2533554-0001
Explorer Master Index to Software Manuals	2243198-0001
Explorer System Software Installation	2243205-0001

Programming With the Explorer

Explorer Programming Primer	2243199-0001
Common LISP, The Language, by Guy L. Steele, Jr.	2537175-0001
Explorer Lisp Reference	2243201-0001
Explorer Zmacs Editor Reference	2243192-0001
Explorer Programming Concepts and Tools	2243130-0001
Explorer Window System Reference	2243200-0001
Explorer Command Interface Toolkit User's Guide	2243197-0001

Explorer Toolkits

Explorer Natural Language Menu System User's Guide	2243202-0001
Explorer Relational Table Management System User's Guide	2243203-0001
Explorer Graphics Toolkit User's Guide	2243195-0001
Explorer Grasper User's Guide	2243135-0001
Explorer Prolog Toolkit User's Guide	2243204-0001
Programming in Prolog, by Clocksin and Mellish	2249985-0001
Explorer Color Graphics User's Guide, Support for the Raster Technologies Model One	2537157-0001
Explorer TCP/IP User's Guide	2537150-0001

System Software Internals

Explorer System Software Design Notes	2243208-0001

THE EXPLORER™ SYSTEM HARDWARE MANUALS

System Level Hardware Publications

Explorer Unpacking and Inventory	2243216-0001
Explorer 7-Slot System Installation	2243140-0001
Explorer System Field Maintenance	2243141-0001

System Enclosure Hardware Publications

Explorer 7-Slot System Enclosure General Description	2243143-0001
Explorer 2-Megabyte Memory General Description	2243147-0001
Explorer Memory General Description	2533592-0001
Explorer Processor General Description	2243144-0001
Explorer Display Unit General Description	2243151-0001
Explorer System Interface General Description	2243145-0001

Mass Storage Hardware Publications

Explorer Mass Storage Enclosure General Description	2243148-0001
Explorer Winchester Disk Formatter (ADAPTEC) Supplement to Explorer Mass Storage Enclosure General Description	2243149-0001
Explorer Winchester Disk Drive (Maxtor) Supplement to Explorer Mass Storage Enclosure General Description	2243150-0001
Explorer Cartridge Tape Drive (Cipher) Supplement to Explorer Mass Storage Enclosure General Description	2243166-0001
Explorer Cable Interconnect Board (2236120-0001) Supplement to Explorer Mass Storage Enclosure General Description	2243177-0001
Explorer NuBus™ Peripheral Interface General Description (NUPI board)	2243146-0001

Mass Storage Hardware Vendor Publications

Series 540 Cartridge Tape Drive Product Description, Cipher Data Products, Inc., Bulletin Number 01-311-0284-1K (¼-inch tape drive)	2249997-0001
MT01 Tape Controller Technical Manual, Emulex Corporation, part number MT0151001 (formatter for the ¼-inch tape drive)	2243182-0001
XT-1000 Service Manual, 5¼-inch Fixed Disk Drive, Maxtor Corporation, part number 20005 (5¼-inch Winchester disk drive, 112 megabytes)	2249999-0001
ACB-5500 Winchester Disk Controller User's Manual, Adaptec, Inc., (formatter for the 5¼-inch Winchester disk drive)	2249933-0001

Optional Equipment Hardware Publications

Explorer NuBus Ethernet® Controller General Description	2243161-0001
Model 855 Printer Operator's Manual	2225911-0001
Model 855 Printer Technical Reference Manual	2232822-0001
Model 855/856 Printer Maintenance Manual	2225914-0001

ABOUT THIS MANUAL

Introduction

This manual describes the Texas Instruments (TI) Explorer system interface board and the auxiliary fiber-optic adapter board used in the Explorer computer system. The information in this manual serves as a standalone descriptive document for the two boards; the *Explorer Field Maintenance Manual* contains the maintenance information for the two boards.

This manual is written for original equipment manufacturers (OEMs), system designers, field maintenance people, and TI customer representatives (CRs).

Contents of This Manual

The manual is divided into four sections and one appendix as follows:

Section 1: Introduction — Provides a general description and a functional overview of the system interface board and the fiber-optic adapter board, lists their important features, and describes how the boards function as a part of an Explorer system.

Section 2: Installation — Provides unpacking and installation procedures for the boards.

Section 3: Operation — Provides instructions for conducting a self-test of the system interface board and for interpreting fault indicator responses to the self-tests.

Section 4: System Design and Programming — Provides an overall view of the operation of each board keyed to block diagrams. This overview description is followed by a more detailed functional description. Additionally, programming information is provided in this section.

Appendix A: Memory Maps — Provides detailed memory maps of major devices on the system interface board.

Related Documents

The following publications contain related reference information. A complete list of the Explorer hardware and software documents is in the frontispiece of this manual.

Document Title	Part Number
Explorer 7-Slot System Installation	2243140-0001
Explorer 7-Slot System Enclosure General Description	2243143-0001
Explorer Display Unit General Description	2243151-0001
Explorer Operations Guide	2243190-0001
NuBus Specification	2242825-0001

The following document provides additional information concerning the serial communications controller used on the system interface board.

Z8030/Z8530 SCC Serial Communications Controller Technical Manual	Zilog, Inc. 1315 Dell Ave. Campbell, Ca. 95008

CONTENTS

	Paragraph	Title	Page
1		**Introduction**	
	1.1	General	1-3
	1.2	Explorer Computer System Overview	1-5
	1.2.1	Explorer Computer System	1-5
	1.2.2	Bus Interfaces	1-7
	1.2.3	Memory	1-8
	1.2.4	Circuit Board Slot Assignments	1-8
	1.2.5	Board Address Configuration	1-8
	1.3	Board Physical and Functional Features	1-9
	1.3.1	SI Board Features	1-9
	1.3.2	Fiber-Optic Board Features	1-11
2		**Installation**	
	2.1	Introduction	2-3
	2.2	Unpacking Procedures	2-3
	2.2.1	SI Board Unpacking Procedure	2-3
	2.2.2	Fiber-Optic Board Unpacking Procedure	2-4
	2.3	Installation and Removal Procedures	2-5
	2.3.1	SI Board Installation Procedure	2-5
	2.3.2	SI Board Removal Procedure	2-6
	2.3.3	Fiber-Optic Board Installation Procedure	2-7
	2.3.4	Fiber-Optic Board Removal Procedure	2-8
3		**Operation**	
	3.1	Introduction	3-3
	3.2	Board Self-Test Procedure	3-3
4		**System Design and Programming**	
	4.1	Introduction	4-3
	4.2	SI Board Physical and Functional Features	4-3
	4.3	Memory Map	4-5
	4.4	SI Board Functional Description	4-5
	4.4.1	NuBus Interface	4-5
	4.4.1.1	NuBus Time-Out Logic	4-10
	4.4.1.2	NuBus Master Clock	4-10
	4.4.1.3	NuBus Acknowledge Generator	4-11
	4.4.2	SI Board Bus Structure	4-11
	4.4.3	Configuration ROM	4-12

Paragraph	Title	Page
4.4.4	Configuration Register	4-15
4.4.5	Event Generator	4-16
4.4.5.1	Event Generator Programming	4-17
4.4.5.2	Event Generator Initialization	4-18
4.4.6	Nonvolatile RAM	4-18
4.4.7	Real-Time Clock	4-20
4.4.7.1	Real-Time Counter	4-20
4.4.7.2	Real-Time Clock Programming	4-20
4.4.8	Programmable Interval Timers	4-23
4.4.8.1	Interval Timer Hardware	4-24
4.4.8.2	Interval Timer Programming	4-25
4.4.9	Parallel Printer Port	4-31
4.4.9.1	Parallel Printer Port Interface Signals	4-32
4.4.9.2	Parallel Printer Port Programming	4-33
4.4.10	Bit-Mapped Graphics Display Controller	4-35
4.4.10.1	Bit-Mapped Memory Functional Description	4-35
4.4.10.2	Logical Operation Unit	4-39
4.4.10.3	Shift Register Latch, Shift Registers, and Video Generator	4-39
4.4.10.4	CRT Controller	4-39
4.4.10.5	NuBus Interface	4-39
4.4.10.6	Local Bus Interface	4-39
4.4.10.7	Bit-Mapped Graphics Display Controller Programming	4-41
4.4.11	Mouse/Keyboard/Audio Interface	4-49
4.4.11.1	Mouse/Keyboard/Audio Interface Address Decoders	4-49
4.4.11.2	Mouse Motion and Position Logic	4-49
4.4.11.3	Keyswitch Detector	4-52
4.4.11.4	Interrupt Enable and Monitor Control Registers	4-52
4.4.11.5	Interrupt Handshake Controller	4-53
4.4.11.6	Sound Control Register	4-55
4.4.11.7	Speech Register	4-56
4.4.11.8	Voice Register and Voice Interrupt Generator	4-57
4.4.11.9	Diagnostic Register	4-57
4.4.12	Keyboard Asynchronous Communication Interface	4-60
4.4.12.1	USART Operation	4-60
4.4.12.2	USART Programming	4-61
4.4.13	Special Chord Detector	4-66
4.4.14	Fiber-Optic Interface	4-67
4.4.14.1	Channel A (SI to Monitor) Description	4-68
4.4.14.2	Channel B (Monitor to SI) Description	4-74
4.4.15	RS-232C Serial Data Port	4-77
4.4.15.1	Port Input and Output Signals	4-81
4.4.15.2	RS-232C Port Programming	4-83
4.4.15.3	Typical Programming Example	4-102
4.4.16	Board Initialization Procedure	4-102
4.5	Fiber-Optic Board Functional Description	4-103
4.5.1	General Description	4-103
4.5.2	RS-232C Interface	4-104

Paragraph	Title	Page
4.5.3	Parallel Printer Interface	4-104
4.5.4	Fiber-Optic Logic and Interface	4-104
4.5.4.1	TTL Signal-Level Interface	4-105
4.5.4.2	Optical Interfaces	4-105

	Appendix	Title	Page
Appendix	A	Memory Maps	A-1

	Figure	Title	Page
Figures	1-1	System Interface Board and Fiber-Optic Board	1-4
	1-2	7-Slot System Enclosure	1-6
	1-3	Explorer System Block Diagram	1-7
	3-1	Example of Fault LED Locations	3-4
	4-1	System Interface Board Block Diagram	4-4
	4-2	SI Board Functional Block Diagram	4-7
	4-3	NuBus Clock Timing Diagram	4-11
	4-4	Configuration Register Bit Assignments	4-15
	4-5	Nonvolatile RAM Memory Map	4-19
	4-6	Control Byte Format	4-26
	4-7	Typical Interval Timer Initialization at Power-Up	4-29
	4-8	Starting the Square-Wave Rate Generator	4-30
	4-9	Starting the Interval Timers	4-31
	4-10	Bit-Mapped Graphics Display Controller Block Diagram	4-36
	4-11	Bit-Mapped Memory Addressing Scheme	4-44
	4-12	Video Attributes Register	4-47
	4-13	Video Test Register	4-49
	4-14	Motion/Keyswitch Data Register Bit Assignments	4-51
	4-15	Mouse Position Register Bit Assignments	4-52
	4-16	Interrupt Enable and Monitor Control Registers	4-54
	4-17	Sound Control Register Bit Assignments	4-56
	4-18	Speech Register Bit Assignments	4-57
	4-19	Voice Register Bit Assignments	4-58
	4-20	Diagnostic Control Field Bit Assignments	4-58
	4-21	Diagnostic Data Field Bit Assignments	4-59
	4-22	USART Mode Instruction Control Byte Format	4-62
	4-23	USART Command Instruction Control Byte Format	4-64
	4-24	USART Status Byte Format	4-65
	4-25	USART Transmit and Receive Data Byte Format	4-66
	4-26	Simplified Raster-Scan Display	4-69
	4-27	Manchester Encoding	4-71
	4-28	Simplified Channel A Format and Timing	4-72

Figure	Title	Page
4-29	Bit Assignments for Channel A Multiplexed Data	4-73
4-30	Channel A Transmit Block Diagram	4-75
4-31	Channel B (Monitor-to-SI) Data Format	4-76
4-32	Channel B Receive Block Diagram	4-77
4-33	Z8530 Input/Output Connections	4-79
4-34	Read Register 0 Bit Assignments	4-88
4-35	Read Register 1 Bit Assignments	4-88
4-36	Read Register 2 Bit Assignments — Channel A Only	4-89
4-37	Read Register 2 Bit Assignments — Channel B Only	4-90
4-38	Read Register 3 Bit Assignments	4-90
4-39	Read Register 10 Bit Assignments	4-91
4-40	Read Register 12 Bit Assignments	4-91
4-41	Read Register 13 Bit Assignments	4-92
4-42	Read Register 15 Bit Assignments	4-92
4-43	Write Register 0 Bit Assignments	4-93
4-44	Write Register 1 Bit Assignments	4-94
4-45	Write Register 2 Bit Assignments	4-94
4-46	Write Register 3 Bit Assignments	4-95
4-47	Write Register 4 Bit Assignments	4-95
4-48	Write Register 5 Bit Assignments	4-96
4-49	Write Register 6 Bit Assignments	4-97
4-50	Write Register 7 Bit Assignments	4-97
4-51	Write Register 9 Bit Assignments	4-98
4-52	Write Register 10 Bit Assignments	4-98
4-53	Write Register 11 Bit Assignments	4-99
4-54	Write Register 12 Bit Assignments	4-100
4-55	Write Register 13 Bit Assignments	4-100
4-56	Write Register 14 Bit Assignments	4-101
4-57	Write Register 15 Bit Assignments	4-101
4-58	Connector P3 and Connecting Cable Pin Assignments	4-104
4-59	Connector P4 and Connecting Cable Pin Assignments	4-106

Table	Title	Page
Tables		
1-1	SI Board Characteristics	1-10
1-2	Fiber-Optic Board Characteristics	1-12
4-1	SI Board Device Base Addresses	4-6
4-2	NuBus Signal Functions	4-9
4-3	Configuration ROM Contents	4-12
4-4	Event Causes and Register File Storage Locations	4-17
4-5	Real-Time Clock Register Addresses and Functions	4-21
4-6	Timing Register and RAM Register Format	4-21
4-7	Interrupt Control Register and Interrupt Status Register Format	4-22
4-8	Interval Timer Register Addresses and Functions	4-27
4-9	Parallel Printer Port Signals	4-32

Table	Title	Page
4-10	Parallel Printer Port Register 0 Bit Functions	4-34
4-11	Parallel Printer Port Register 1 Bit Functions	4-35
4-12	Local Bus Signal Functions	4-40
4-13	CRT Controller Register Functions	4-43
4-14	Logical Functions	4-45
4-15	Address Map for Mouse, Audio, and Diagnostic Registers	4-50
4-16	Addresses and Functions of USART Registers	4-62
4-17	Monitor Timing	4-69
4-18	RS-232C Port Input and Output Signals	4-81
4-19	Summary of Z8530 Read Register Functions	4-84
4-20	Summary of Z8530 Write Register Functions	4-85
4-21	RS-232C Register Addresses	4-86
4-22	Baud Rate Generator Divisors	4-87
4-23	SI Board and Backplane Signals	4-107

Example	Title	Page
Example 4-1	Read-Modify-Write AND Operation	4-46

1 INTRODUCTION

Highlights of This Section

- Explorer computer system overview
- System interface board features and characteristics
- Fiber-optic board features and characteristics

General

1.1 This section contains a general description and a functional overview of the TI Explorer system interface (SI) board and the auxiliary TI Explorer fiber-optic adapter (driver/receiver) board (see Figure 1-1).

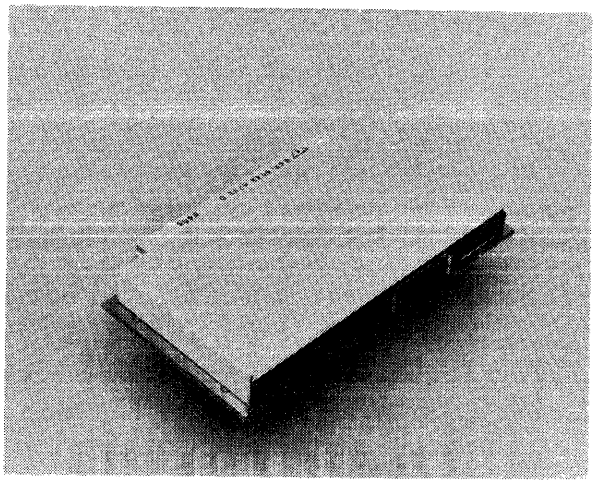
The SI board installs into an Explorer computer system to provide a wide variety of user and system service functions on a single logic board. The SI board is designed in the standard three-high Eurocard format and is installed into one slot of the chassis backplane via three 96-pin DIN connectors at the bottom of the board. The SI board provides the following:

- Two programmable interval timers
- Nonvolatile random-access memory (RAM)
- Bit-mapped graphics controller
- Real-time clock
- Fiber-optic channel interface
- Parallel printer port
- RS-232C interface port
- Accessory functions for the Explorer processor

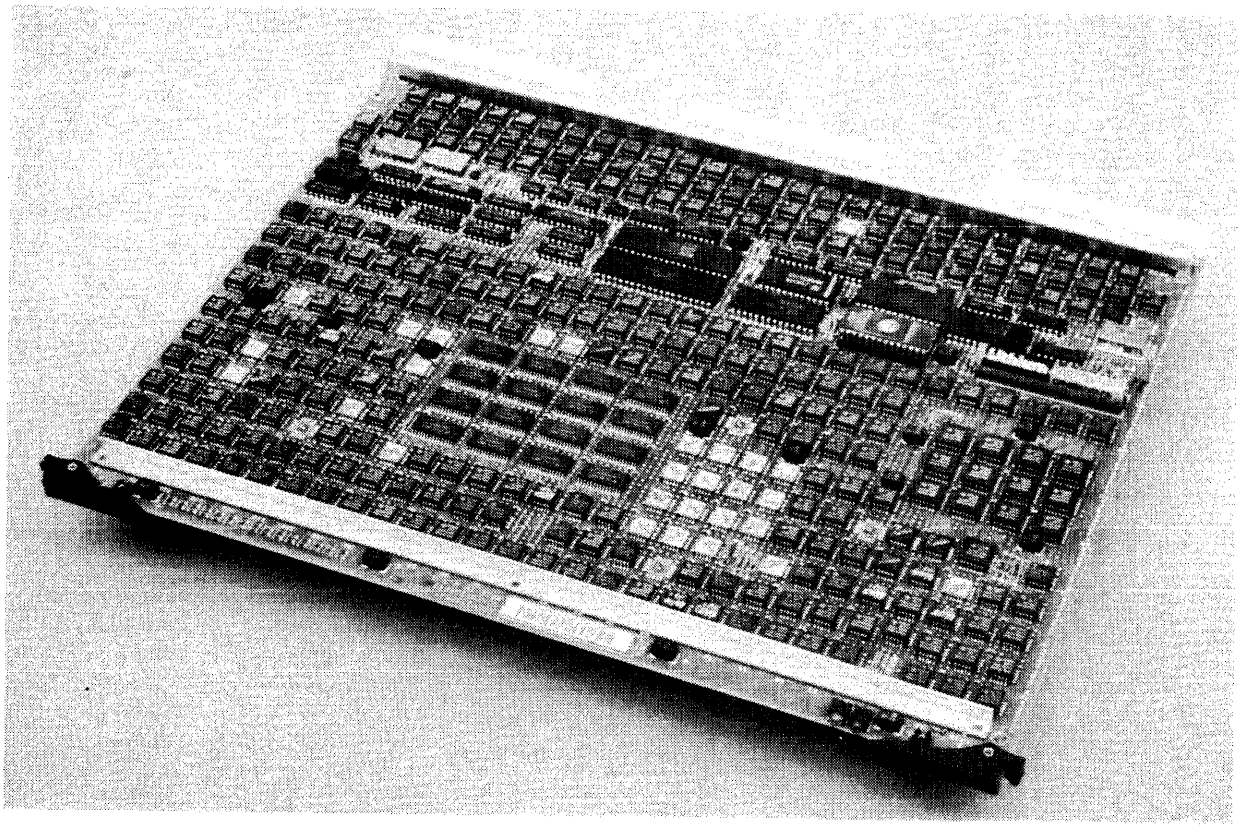
The fiber-optic adapter board, referred to in this manual as the *fiber-optic board*, is used with the SI board to provide single-channel TTL-to-optical signal conversion and single-channel optical-to-TTL signal conversion. In this capacity, the fiber-optic board forms one part of a duplex fiber-optic communication link between the computer system enclosure and the monitor.

The fiber-optic board also serves as a passive adapter to extend the RS-232C and parallel printer port signals of the SI board to corresponding input/output (I/O) cable connectors. The fiber-optic board mounts behind the chassis backplane on the back side of an extended-pin DIN connector at the chassis slot occupied by the SI board. The size and shape of the fiber-optic board is consistent with that of standard Explorer I/O adapter boards. There are four connectors on the board: one for the backplane, one for fiber-optic I/O, and two for conventional electrical cables to I/O devices.

Figure 1-1 System Interface Board and Fiber-Optic Board



FIBER-OPTIC ADAPTER BOARD



SYSTEM INTERFACE BOARD

Explorer Computer System Overview

1.2 This overview describes the major functions and operation of the SI board and the fiber-optic board as part of an Explorer system. A computer system that contains an Explorer processor is more specifically referred to as an Explorer computer system. The following paragraphs describe an Explorer system that has the Explorer processor, the SI board, and other circuit boards installed in a 7-slot system enclosure.

Explorer Computer System

1.2.1 The Explorer system is an advanced, single-user workstation designed to process high-level, stack-oriented Lisp-like languages. The Explorer system features include fast processing, large memory capacity, and high-quality graphics. This system provides extensive support for the development of large-scale, complex programs and for research in new technologies, including artificial languages. Figure 1-2 shows a basic configuration of an Explorer in a 7-slot system enclosure. The configuration shown consists of:

- A free-standing system enclosure that has a seven-slot backplane into which the Explorer processor, the SI and fiber-optic boards, and up to five other circuit boards are installed. The system dc power supply is also housed within the system enclosure. A power on/off switch is the only operating control on the outside of the enclosure.

- A desktop display unit that includes a high-resolution cathode-ray tube (CRT) display monitor, a detached keyboard, a microphone/headphone connection, and an optical mouse for operator control.

- A compact, desktop, mass-storage enclosure with formatter logic and one or two mass-storage devices.

The enlarged view of the 7-slot system enclosure, Figure 1-2, shows the placement of the processor and other circuit boards in the enclosure.

The basic Explorer system block diagram (see Figure 1-3) shows the major operating elements within the system enclosure and their local bus and NuBus interfaces with each other. Operating features of the SI and fiber-optic boards are described in the following paragraphs.

Figure 1-2 7-Slot System Enclosure

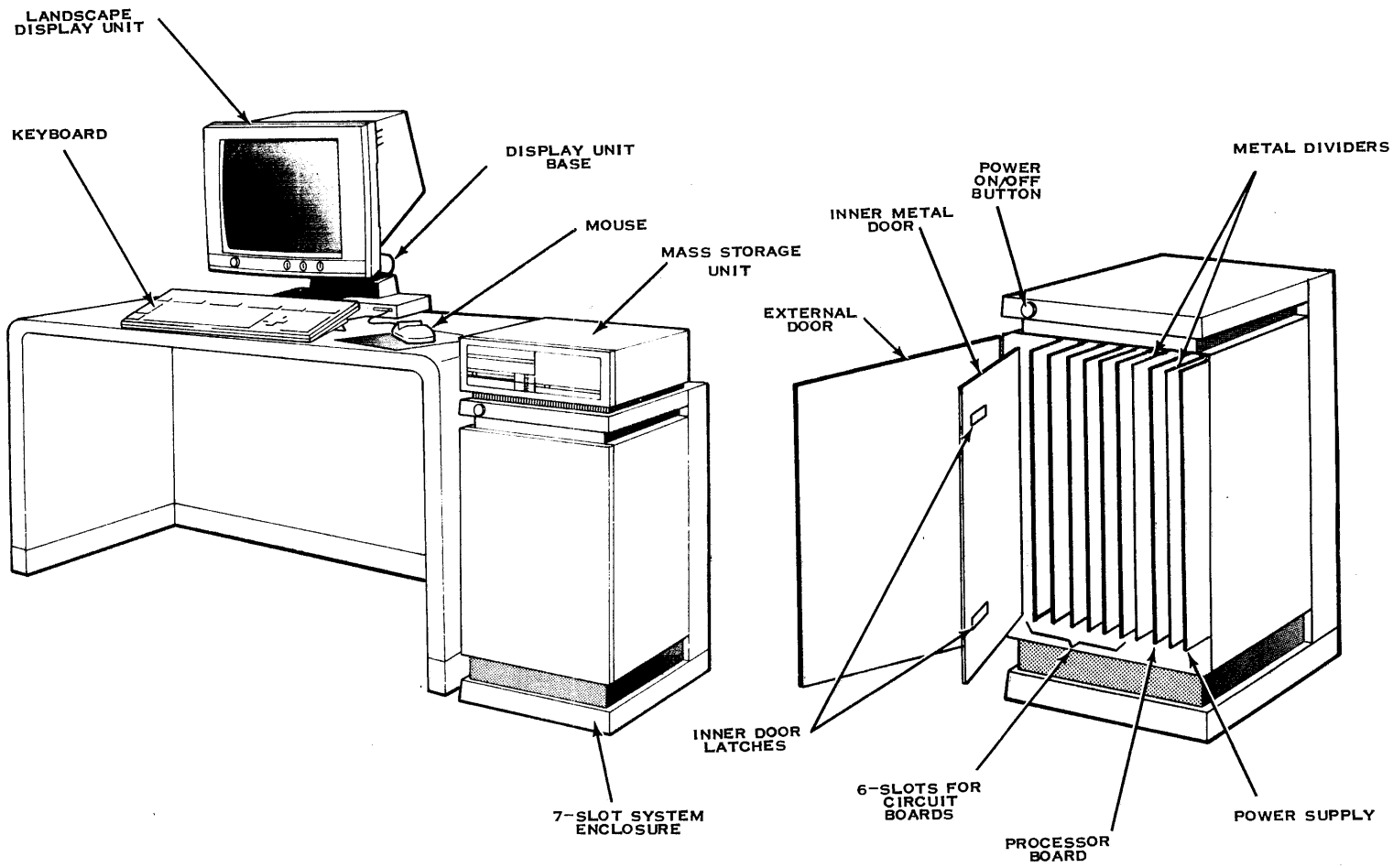
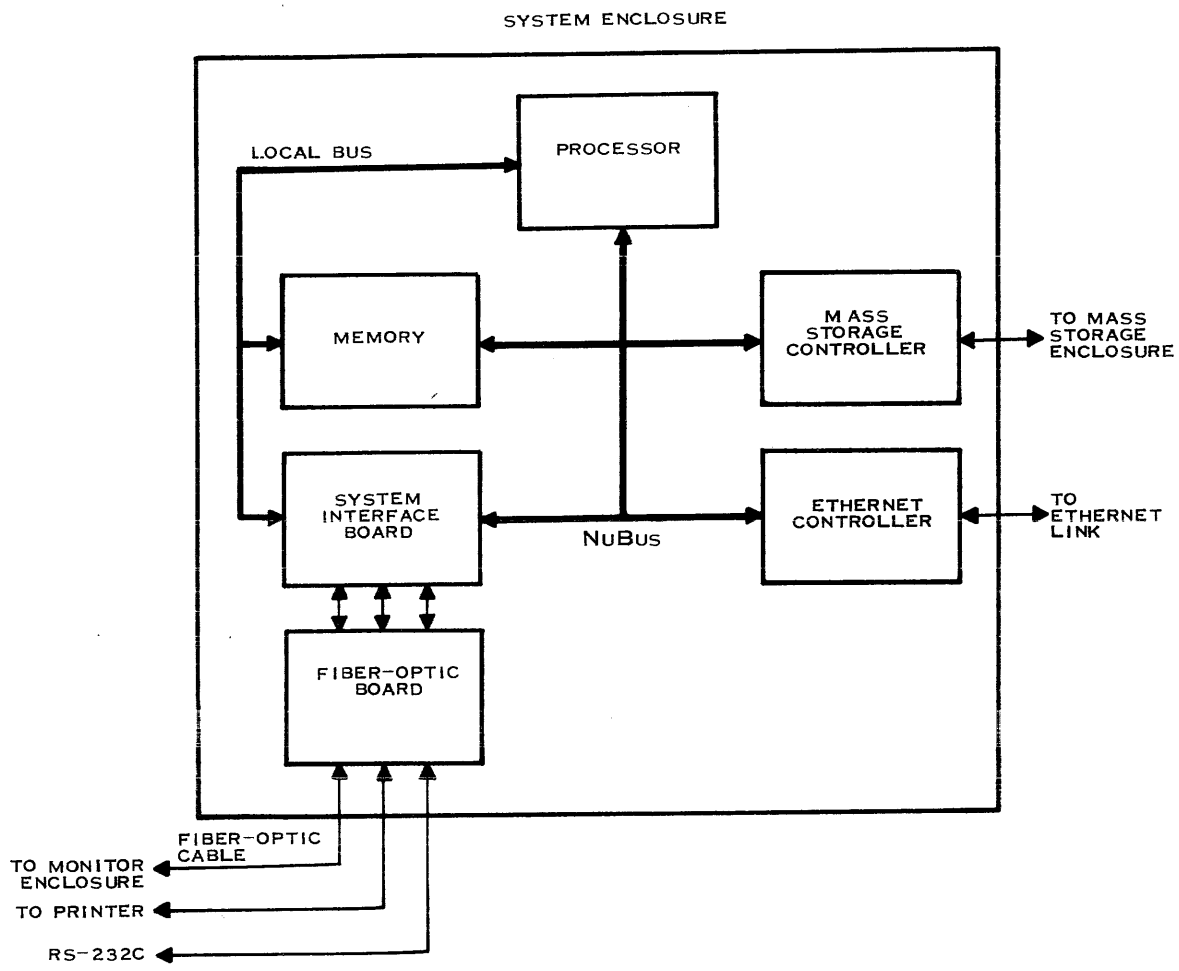


Figure 1-3 Explorer System Block Diagram



Bus Interfaces

1.2.2 The NuBus data transfer bus is a 32-bit, synchronous bus that has the address and data lines multiplexed onto the same lines. A simple communications protocol allows circuit boards connected to the NuBus to be either master or slave devices. All control, data, address, and power lines are brought to one 96-pin DIN connector. On the enclosure backplane, connector P1 is reserved in all slots for the NuBus. Refer to the *NuBus Specification* for detailed NuBus information.

The local bus also is a high-speed, synchronous bus that has separate lines provided for the 32-bit address codes and the 32-bit data lines. All control, data, address, and power lines for the local bus are brought to one 96-pin DIN connector. The Explorer processor is the master device on the local bus; all other circuit boards on the local bus are slave devices.

Memory

1.2.3 Memory is accessed by the processor over the local bus. A memory board has a capacity of two megabytes of memory when 64K-bit (where K equals 1024) dynamic random-access memory (DRAM) chips are used. The memory board communicates with other components of the system over the NuBus and the local bus. One memory board, when enabled by a slot-dependent local bus signal from the processor, becomes a NuBus master device to act as the local-bus-to-NuBus transfer port for the processor.

Circuit Board Slot Assignments

1.2.4 Electrically, some circuit boards can be installed into one of several board slots in the enclosure backplane. However, there are some physical limitations that must be observed. For example, the Explorer processor imposes a heavier heat load than other Explorer logic boards. Because of this heat loading, the processor must be installed in a board slot that has special airflow ducting. This is slot 6 of the 7-slot system enclosure. The memory board that is to serve as the NuBus master for the processor must be installed in an available board slot that has the enabling signal from the processor wired to its local bus connector. This is slot 4 of the 7-slot system enclosure. The SI board is installed into board slot 5 of the 7-slot system enclosure. This slot selection is to allow the NuBus clock, generated by the SI board, to be propagated from as near the end of the bus as possible. The *Explorer 7-Slot System Enclosure General Description* contains further information about board slot assignments.

Board Address Configuration

1.2.5 In general, a circuit board can be inserted into any slot of a system enclosure backplane without jumpers or switches on the board to link it to a specific location. The slots themselves are identified by a hard-wired identification code (hexadecimal 0 through F) on the backplane. This identification code becomes part of the board address configuration when the board is inserted into a slot. The board slot address configuration is hexadecimal FSXXXXXX where S represents the slot identification code and X represents the hexadecimal digit.

A configuration read-only memory (ROM) is provided on each circuit board. Along with other information, this configuration ROM contains a unique identifying name for the board of which it is a part. This name is addressed along with the slot identification code during power-up so that the processor knows which type of circuit board is in each slot during all succeeding NuBus and local bus operations. Configuration ROM contents for the SI board are described in detail in Section 4.

Each backplane slot is allocated 16 megabytes of address space in the range of FS000000 through FSFFFFFF. The total address space that is reserved and mapped for the 16 possible slots on the NuBus is 256 megabytes. This is only one-sixteenth of the total 4-gigabyte address space capability of the 32-bit NuBus and local bus. The unused address space, from 00000000 to F0000000, is uncommitted and can be used as required.

Board Physical and Functional Features

1.3 Physical and functional features of the SI board and the fiber-optic board are provided in the following paragraphs.

SI Board Features

1.3.1 Features of the SI board hardware that support user and Explorer computer system service functions include:

- An interface to the monitor, keyboard, mouse, and speech and sound control circuits through a fiber-optic link (the mouse acts as a quick pointing device over the full range of the video display)
- Graphics control logic
- Parallel printer and RS-232C ports
- System resources such as the interval timers and real-time clock

The fiber-optic link transmits the display image and the keyboard and mouse data. The keyboard interface provides a special chord detector to monitor keyboard input for keystroke sequences that can initiate a computer system reboot. Sound-control circuits provide programmed control of the sound generator in the display monitor.

The graphics controller uses a one-megabit, bit-mapped memory to store the high-resolution display image. The bit-mapped memory is implemented with high-speed random access memory (RAM) for instantaneous image update. The video display is completely refreshed 60 times per second to minimize flicker.

Resources provided to the system include clocks, timers, nonvolatile memory, and power failure event logic. A battery-powered time and date clock makes it unnecessary to reinitialize the system clock on power-up. Other interval timers can be used to issue events at programmable intervals. The nonvolatile memory maintains critical data through power-off, power failure, and board removal conditions. The SI board monitors the power failure warning line from the power supply and can issue power failure warnings as an event to two predesignated addresses upon power failure. Table 1-1 lists general characteristics for the SI board.

Table 1-1 **SI Board Characteristics**

Item	*Characteristics*
Temperature:	
Operating	4 to 45 degrees C (40 to 113 degrees F)
Nonoperating	−40 to 65 degrees C (−40 to 149 degrees F)
Gradient	Less than 10 degrees C per hour
Relative humidity (noncondensing):	
Operating	8 to 80 percent
Nonoperating	5 to 95 percent
Altitude:	
Operating	−300 to 3000 meters (−990 to 10 000 feet)
Nonoperating	−300 to 12 000 meters (−990 to 40 000 feet)
Mechanical shock (in 3 axes):	
Operating	15 g
Nonoperating	25 g
Mechanical vibration (in 3 axes):	
Operating	0.5 g
Nonoperating	1.0 g
Dimensions:	
Width	366.7 mm (14.437 in)
Depth	280 mm (11.024 in)
Thickness (with components)	18.11 mm (0.714 in) maximum
Electromagnetic emission	The SI board is a component of an Explorer system and is tested to meet applicable FCC and VDE electromagnetic interference requirements at the system level.

Table 1-1 **SI Board Characteristics (Continued)**

Item	*Characteristics*	
Power usage:		
Voltage	***Current***	***Power***
+5.0 Vdc	10.5 A (typical)	52.5 watts (typical)
+12.0 Vdc*	0.38 A (typical)	4.6 watts (typical)
−12.0 Vdc	0.36 A (typical)	4.3 watts (typical)

NOTE:

* This voltage does not include +12 Vdc power supplied to the voltage regulator or the fiber-optic adapter board. Add 0.020 amps (typical).

Fiber-Optic Board Features

1.3.2 The features of the fiber-optic board are as follows:

- Provides an electro-optical driver and a receiver for digital communication on separate fibers of a duplex fiber-optic cable
- Provides TTL-compatible signals to the SI board
- Accepts TTL-compatible signals from the SI board
- Serves as a physical adapter and extender between a DIN electrical connector at the rear of the backplane and the RS-232C and parallel printer cable connectors.

Characteristics of the fiber-optic board are shown in Table 1-2.

Table 1-2 Fiber-Optic Board Characteristics

Item	*Characteristics*
Temperature:	
Operating	4 to 45 degrees C (40 to 113 degrees F)
Nonoperating	−40 to 65 degrees C (−40 to 149 degrees F)
Gradient	Less than 10 degrees C per hour
Relative humidity (noncondensing):	
Operating	8 to 80 percent
Nonoperating	5 to 95 percent
Altitude:	
Operating	−300 to 3000 meters (−990 to 10 000 feet)
Nonoperating	−300 to 12 000 meters (−990 to 40 000 feet)
Mechanical shock (in 3 axes):	
Operating	15 g
Nonoperating	25 g
Mechanical vibration (in 3 axes):	
Operating	0.5 g
Nonoperating	1.0 g
Dimensions:	
Height	124.5 mm (4.9 in)
Width	71.9 mm (2.83 in)
Thickness (with components)	16.5 mm (0.65 in) maximum
Electromagnetic emission	The fiber-optic board is a component of an Explorer system and is tested to meet applicable FCC and VDE electromagnetic interference requirements at the system level.

Table 1-2 **Fiber-Optic Board Characteristics (Continued)**

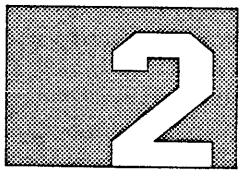
Item	*Characteristics*
Optical signaling (baud) rate:	
Transmit channel (channel A)	50 MHz maximum, board inherent (limited to 40 MHz on receive end by detector in monitor)
Receive channel (channel B)	10 MHz, limited by on-board detector
Optical power peak wavelength:	
Transmit and receive	820 nanometers (nominal)
Optical rise/fall time:	
Transmit	3.0 nanoseconds (typical)
Receive	35 nanoseconds (typical)
Optical transmit power	1400 microwatts (typical) (measured at 820 nanometers over a 250 micrometer spot)
Receiver optical overload power	30 microwatts (maximum)
Power usage:	

Voltage	*Current*	*Power*
+5.0 Vdc	85 mA (typical)	425 mW (typical)
+12.0 Vdc*	12 mA (typical)	144 mW (typical)

Note:

* The +12 Vdc power is supplied to the fiber-optic board by the SI board.

INSTALLATION



Highlights of This Section

- SI and fiber-optic board unpacking procedures
- SI and fiber-optic board installation procedures
- SI and fiber-optic board removal procedures

Introduction

2.1 This section contains unpacking, installation, and removal instructions for the system interface (SI) board and the fiber-optic board.

Unpacking Procedures

2.2 Unpacking procedures for the SI board and for the fiber-optic board are in the following paragraphs.

SI Board Unpacking Procedure

2.2.1 The SI board contains a factory-installed, 3-volt, lithium battery. Under operating conditions, this lithium battery is used as a backup battery to prevent loss of data in the nonvolatile RAM and in the real-time clock under power-off or power-failure conditions or when the board is removed from its board slot.

The following caution regarding static control is applicable when the SI board is not installed.

CAUTION: The SI board contains static-sensitive electronic components. To avoid damage to these components, make sure that you are properly grounded before handling the board.

The recommended grounding method is to use a static-control system composed of a static-control floor or a table mat and a static-control wrist strap. These are commercially available. If you do not have a static-control system, you can discharge any static charge by touching a grounded object prior to handling the board. Then, as an additional safety measure, put the board on a grounded work surface after removing it from its static-protective package or from the system enclosure.

Before transporting or storing the printed circuit board, return it to its static-protective package or to the system enclosure.

Observe the following caution when handling the SI board.

CAUTION: To prevent accidental discharge of the battery, do not lay the SI board on top of a static-protective bag, a chassis, or any other conductive surface. (The exterior of the static-protective bag is electrically conductive.) Even momentary contact with a conductive surface can destroy the data in the nonvolatile RAM or discharge the backup battery. Place the SI board in the backplane of the computer system.

If you need to return the SI board for service, place the board inside its static-protective bag for shipment.

Perform the following procedure to unpack the SI board:

1. Check for any documents or written instructions attached to the exterior of the SI board packing container. Read and follow the instructions.

2. Open the packing container that contains the SI board, and carefully remove the packing material.

3. Remove the SI board from the static-protective bag, keeping the bag intact. Be sure to follow the static-sensitive caution when handling the board outside of the static-protective bag.

4. Check the SI board for any physical damage that may have occurred during shipping. If the board is damaged, follow your normal procedure for reporting a damaged shipment.

Fiber-Optic Board Unpacking Procedure

2.2.2 The fiber-optic board does not contain static-sensitive components and may or may not be shipped in a static-protective bag. Perform the following procedure to unpack the fiber-optic board:

1. Check for any documents or written instructions attached to the exterior of the fiber-optic board packing container. Read and follow the instructions.

2. Open the packing container that contains the fiber-optic board. Carefully remove the packing material from the fiber-optic board and remove the board from the packing container.

3. Check the fiber-optic board for any physical damage that may have occurred during shipping. If the board is damaged, follow your normal procedure for reporting a damaged shipment.

Installation and Removal Procedures

2.3 Refer to the general description manual for your system enclosure for the details about slot utilization and for the exact locations of the operating controls on the enclosure. The system enclosure manual also contains specific instructions for opening the front and rear panels to gain access to the inside of the system enclosure.

WARNING: Do not operate the computer system when normally closed doors or panels of the system enclosure are opened. Under normal conditions, interlocks prevent power from being applied when these doors and panels are not in place. Do not bypass or otherwise tamper with the interlocks. Lethal voltages are exposed if this precaution is not observed.

CAUTION: The board slots require high-airflow ducting for cooling air. Proper airflow occurs only with the doors and panels in place during operation. If the enclosure cooling system is disturbed, the resultant heat buildup can damage the circuit boards.

SI Board Installation Procedure

2.3.1 Install the SI board into the system enclosure as follows:

NOTE: You do not need to set or check any jumpers or switches prior to installing the SI board.

---

1. Set the power switch, located at the top left corner of the system enclosure, to the off (out) position.
2. Open the front exterior door of the system enclosure.
3. Release the latches on the internal metal door of the 7-slot system enclosure and open this door.

CAUTION: These additional precautions should be taken when handling the SI board.

The exterior of the static-protective bag is electrically conductive. Do not place the SI board, which has a factory-installed lithium battery, on a static-protective bag, a chassis, or any other conductive surface. Failure to observe this can result in discharge of the lithium battery and/or damage to electronic components. In the case of a board that has been removed from an operating system, failure to observe this precaution can result in the loss of data in the nonvolatile RAM and the real-time clock.

4. With connector P1 to the top and the red fault light-emitting diode (LED) to the bottom, carefully slide the SI board into the connectors on the backplane at the slot 5 position. When you are facing the front of the Explorer system, the slot at the far left is slot 0.

5. Use the two injector/ejector tabs at the top and bottom of the board to lock the board into place in the slot. Ensure that the injector tabs are completely closed for proper board seating.

6. Close and latch the internal door.

7. Close and lock the front door of the system enclosure.

8. Perform the self-test procedure described in Section 3 of this manual.

SI Board Removal Procedure

2.3.2 Remove the SI board from the system enclosure as follows:

1. Observe the precautions for handling an SI board.

2. Set the power switch on the system enclosure to the off (out) position.

3. Open the front exterior door of the system enclosure.

4. Release the latches on the internal metal door of the 7-slot system enclosure and open this door.

5. Remove the SI board from slot 5 of the system enclosure by using the injector/ejector tabs located at the top and bottom edge of the board to unlock the board. Carefully slide the board out of its position in the backplane and place it on a nonconductive surface or into the static-protective bag.

6. Close and latch the internal door and the outer front door of the system enclosure.

Fiber-Optic Board Installation Procedure

2.3.3 Install the fiber-optic board into the system enclosure as follows:

NOTE: You do not need to set or check any jumpers or switches prior to installing the fiber-optic board.

---

1. Set the power switch on the system enclosure to the off (out) position.

2. Open the rear door of the enclosure.

3. Locate the 96-pin reverse DIN connector on the backplane at the connector P3 position of the board slot in which the SI board is installed. This is slot 5 in the 7-slot chassis. When facing the rear of the system enclosure, slot 0 is to the far right. Carefully slide the 96-pin P1 connector end of the fiber-optic board into the board guides and into the 96-pin reverse DIN connector.

   NOTE: If your computer system includes both RS-232C and parallel printer cable connections, install the 40-pin printer cable connector to connector P4 before installing the smaller RS-232C cable connector. The RS-232C cable connector interferes with the installation and removal of the larger connector.

   ---

4. Refer to the system enclosure general description manual for procedures for installing peripheral cables. Install cables to the fiber-optic board connectors as follows:

   a. Connect the parallel printer cable to connector P4.

   b. Connect the RS-232C cable to connector P3.

   c. Connect the fiber-optic cable to connector P2.

5. Close and lock the rear door of the enclosure.

6. Perform the self-test procedure described in Section 3 of this manual.

Fiber-Optic Board Removal Procedure

2.3.4 Remove the fiber-optic board from the system enclosure as follows:

1. Set the power switch on the enclosure to the off (out) position.
2. Open the rear door of the enclosure.
3. Locate the 96-pin reverse DIN connector on the backplane at connector P3 position of slot 5 where the SI board is installed.
4. Refer to the system enclosure general description manual for procedures for removing peripheral cables. Remove peripheral cables from connectors P4, P3, and P2 of the fiber-optic board as applicable.
5. Carefully remove the 96-pin P1 connector of the fiber-optic board from the 96-pin reverse DIN connector at the chassis backplane and remove the fiber-optic board from the system enclosure.
6. Close and lock the rear door of the system enclosure.

OPERATION 3

Highlights of This Section

This section provides instructions for conducting a self-test of the SI board after it is installed in the system enclosure.

Introduction

3.1 Self-testing of the Explorer computer system occurs automatically at a system initialization (power-up), a microcode restart (warm boot), or a power-down/power-up restart. You can force a self-test after initial power-up either by turning the system power off and then on or by entering a command from the keyboard, as described in the following procedure.

All boards go through a self-test any time the system is booted. The Explorer processor board controls the initiation of all self-tests. During this sequence, the SI board runs a self-test. The fault LED at the edge of the board lights while the self-test is in progress, which takes about 15 seconds. If the fault LED stays on longer than 15 seconds, additional testing of the board is needed. Refer to the *Explorer System Field Maintenance* manual for corrective action procedures.

Board Self-Test Procedure

3.2 Perform the following self-test procedure after installing the SI board in the system enclosure.

1. Open the outer front door of the system enclosure so that the fault LEDs for the logic boards are visible through slots in the internal metal door. Do not open this metal door.

2. To begin the self-test when the power is initially off (that is, with the power switch out), set the power switch on the enclosure to the on (in) position. To initiate a self-test after initial power-up, force a cold reset of the system by either:

   a. Turning the enclosure power off and then on; or

   b. Entering the cold boot keystroke chord from the keyboard by simultaneously pressing the META-CTRL-META-CTRL-ABORT keys. All boards perform a self-test any time that the system is cold booted.

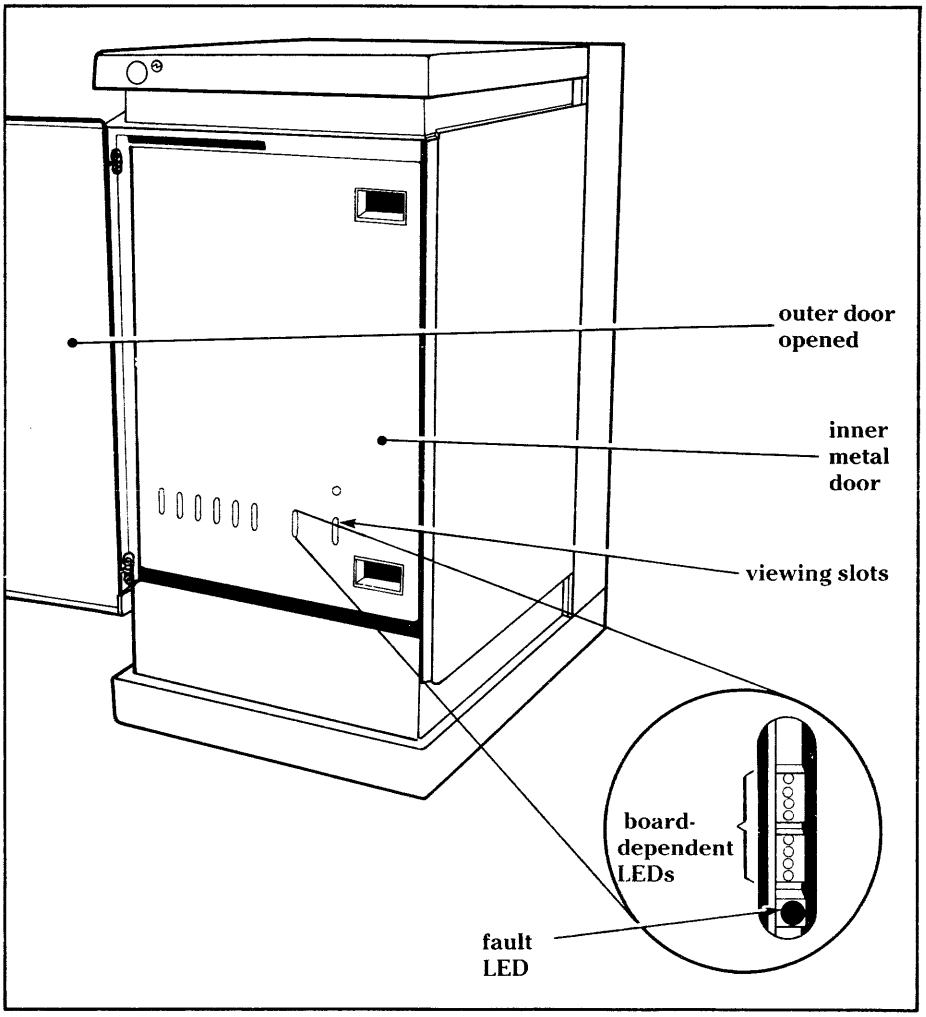
3. During the self-testing, the system prints the name of each test and the result on the monitor. Check the fault LEDs, which are located near the bottom of the circuit boards installed in the enclosure board slots (see Figure 3-1), as viewed from the front of the enclosure. For the SI board in slot 5, the yellow fault LED is the monitor fault LED and the red fault LED is the SI board fault LED.

4. Check that the red fault LED on the SI board in the enclosure turns on to indicate that the self-test for the board is in progress, which takes about 15 seconds.

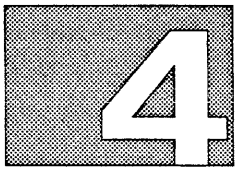
5. If no fault is detected by the self-test, the red fault LED on the processor turns off, which indicates that the processor is good. If the yellow fault LED turns off, both the monitor and the fiber-optic board are functioning properly. However, if the yellow fault LED remains lit, you must perform further testing to determine if the fault is caused by the fiber-optic board or by the monitor. The system then performs additional self-tests by running the self-test microcode on other boards in the enclosure.

6. If the SI board is faulty, replace the board and repeat this self-test procedure. Refer to Section 2 of this manual for board installation and removal procedures.

Figure 3-1 **Example of Fault LED Locations**



SYSTEM DESIGN AND PROGRAMMING



Highlights of This Section

- Detailed functional description of the SI board
- Detailed functional description of the fiber-optic board

Introduction

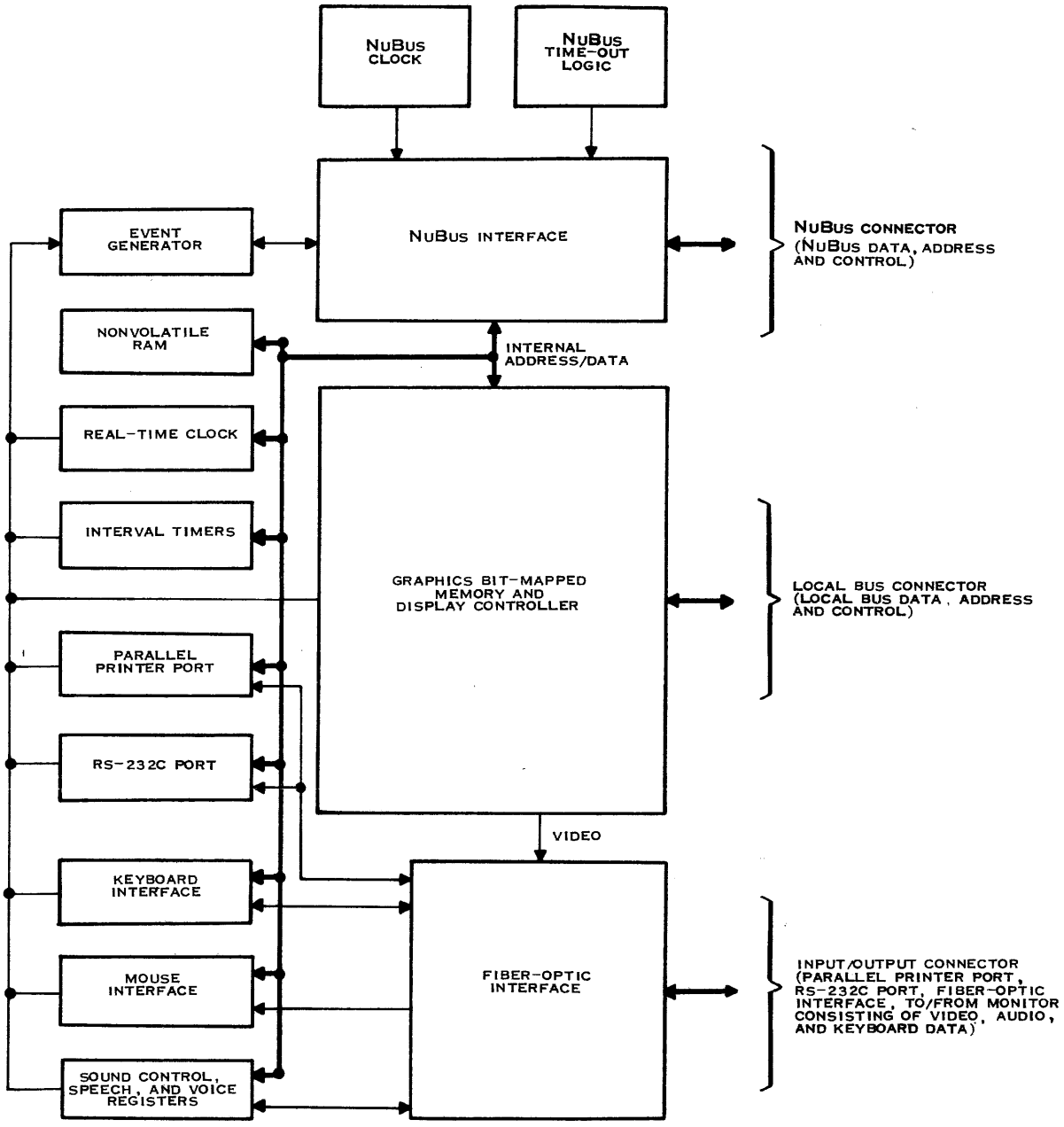
4.1 This section provides functional descriptions of the SI board and the fiber-optic board. A description of the features and overall operation of each board, keyed to a basic block diagram, is followed by a more detailed functional description. Programming and SI board initialization information is also included in this section.

SI Board Physical and Functional Features

4.2 Important functional features of the SI board and their interaction (as shown in the major blocks of Figure 4-1 include:

- 10-megahertz NuBus clock — Provides NuBus clock timing for all NuBus operations. (The clock can be disabled to allow an alternate clock to provide timing for the NuBus.)
- Nonvolatile RAM — Uses battery power to maintain critical data intact in a 2-kilobyte (2048 bytes) RAM through power-off and power-failure conditions.
- Real-time clock — Provides time and date information in a real-time clock circuit to issue events at programmable intervals. Battery backup power keeps the clock running through power-off and power-failure conditions.
- Two programmable interval timers — Allow short-interval timing or longer-interval cascade timing that is independent of the real-time clock.
- Fiber-optic interface — Provides data multiplexing and demultiplexing functions between the serially formatted optic link and the parallel data paths of the SI board.
- Graphics bit-mapped memory and display controller — Provides a display data transfer to a high-resolution monochrome video display.
- Mouse interface — Provides mouse position data, mouse motion events, and mouse keyswitch events for system processing. The mouse acts as a quick pointing device over the full range of the video display.
- Sound control logic — Provides program control of the sound/noise generator in the monitor. The use of parity ensures that an error in transmission does not produce an irritating sound error at the monitor.
- Digital speech output and voice input paths — Allow full voice input and speech output at the display monitor. The display monitor and the SI board provide the data paths, but an optional speech/voice processor board is required to complete the speech synthesis/voice analysis process.

Figure 4-1 System Interface Board Block Diagram



- Keyboard interface — Accepts keyswitch make, break, and special-purpose codes from the keyboard and sends test and other commands to the keyboard.

- Special chord detector — Monitors the incoming keyboard data stream for a code sequence associated with the keyboard-controlled system boot. The chord detector forces a computer system reset upon recognition of the boot code. (The chord detector is not shown in Figure 4-1.)

- RS-232C and parallel printer ports — Provide standard connections to external hard-copy and communication devices.

- Power-failure event logic — Monitors the power-failure warning line from the power supply and generates power-failure events to either of the two predefined event addresses. (The power-failure event logic is not shown in Figure 4-1.)

- Event generator — Monitors other circuit interrupts on the SI board and, when enabled, responds to circuit interrupts by transmitting events to NuBus event addresses.

Memory Map

4.3 All major devices on the SI board are mapped to memory locations within the NuBus address space that is allotted to the board. Each memory-mapped device is assigned a base address in memory and as many additional contiguous memory locations from that base address as needed. The content and significance of the base address and the additional memory locations for each device are described in detail in the following paragraphs describing the device.

For convenience, Table 4-1 provides a summary of the hexadecimal base addresses of the complete NuBus memory map for the SI board devices.

SI Board Functional Description

4.4 The following paragraphs contain operational and programming information for the SI board. The description is keyed to Figure 4-2, System Interface Board Functional Block Diagram. After programming and initialization procedures for the individual devices on the SI board are described, the paragraph titled Board Initialization Procedures provides an SI board initialization procedure that must be followed to properly initialize the board from power-up or after a full system reset.

NuBus Interface

4.4.1 The NuBus interface provides the buffering, decoding, multiplexing, and timing signals that are required to properly connect the system NuBus to the internal bus (I bus) of the SI board. The I bus includes data, address, and control lines to interconnect the various bus-oriented functions located on the SI board. Your system enclosure manual describes the signal lines, timing, and operation of the NuBus in detail. Table 4-2 lists and briefly describes the NuBus signals.

Table 4-1 SI Board Device Base Addresses

Device	*NuBus Base Address (Hexadecimal)*
Graphics control space	FSE00000 through FSE7FFFF
Graphics bit-mapped memory:	
Displayed	FSE80000 through FSE993FF
Nondisplayed	FSE99400 through FSE9FFFF
Event generator	FSF00000
Parallel printer port	FSF10000
Mouse registers	FSF20000
Real-time clock	FSF80000
Interval timers	FSF90000
Nonvolatile RAM	FSFA0000
RS-232C port	FSFB0000
Keyboard USART	FSFC0000
Configuration ROM	FSFFE000 through FSFFFFFF

NOTE:

S in the hexadecimal address represents the electrical slot identification code, hexadecimal 0 through F.

The NuBus interface allows the SI board to operate as either a master or slave device. When the SI board is operating as a slave device, other system boards can read from or write to addressable locations on the SI board. When operating as a master device, the SI board can arbitrate for and gain control of the NuBus, and write to any location in system memory. The SI board acts as a NuBus master only to post events. All other SI board NuBus operations are slave operations.

NOTE: The SI board does not support NuBus block transfer operations or parity.

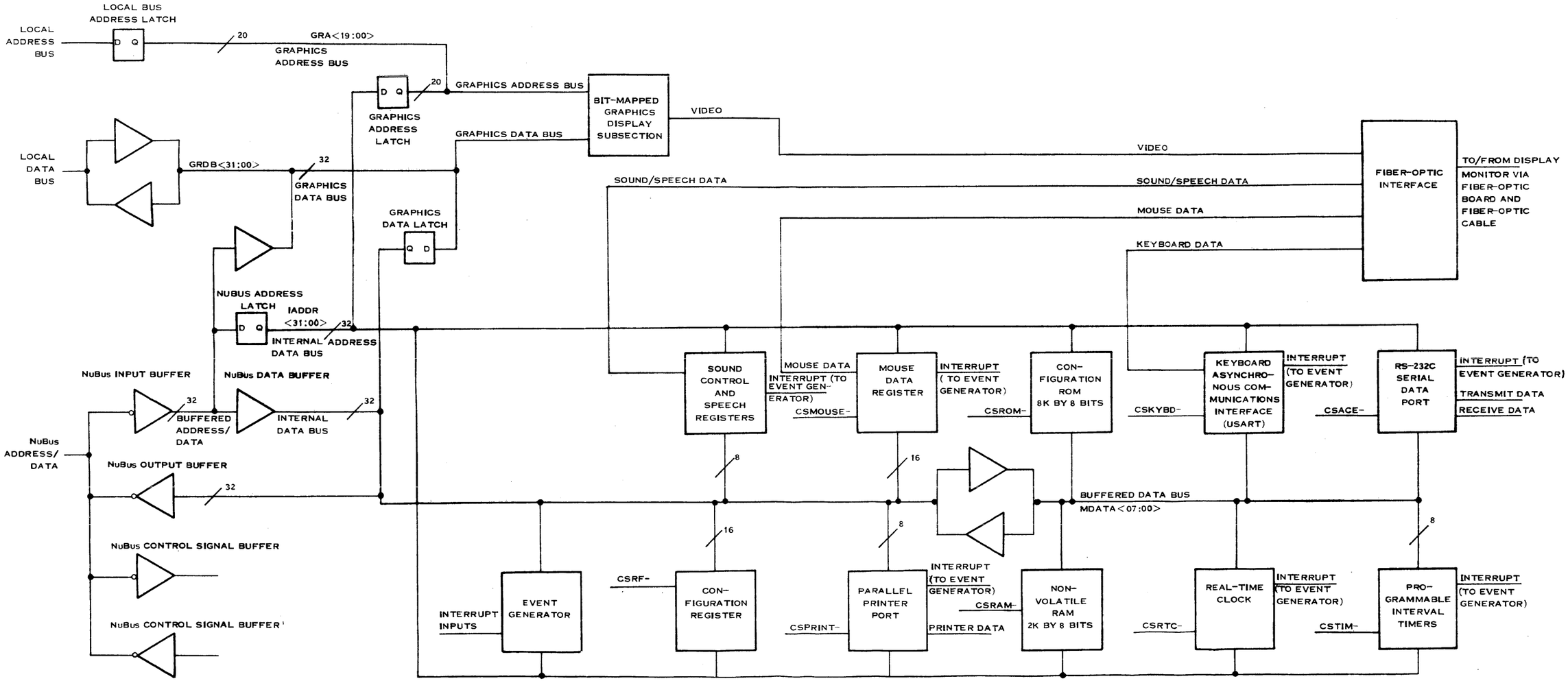

LOCAL BUS ADDRESS LATCH
LOCAL ADDRESS BUS
D Q
20
GRA<19:00>
GRAPHICS ADDRESS BUS
D Q
20
GRAPHICS ADDRESS BUS
GRAPHICS ADDRESS LATCH
BIT-MAPPED GRAPHICS DISPLAY SUBSECTION
VIDEO
GRAPHICS DATA BUS
LOCAL DATA BUS
GRDB<31:00>
32
GRAPHICS DATA BUS
GRAPHICS DATA LATCH
Q D
VIDEO
SOUND/SPEECH DATA
SOUND/SPEECH DATA
MOUSE DATA
KEYBOARD DATA
FIBER-OPTIC INTERFACE
TO/FROM DISPLAY MONITOR VIA FIBER-OPTIC BOARD AND FIBER-OPTIC CABLE
NUBUS ADDRESS LATCH
D Q
IADDR <31:00>
32
INTERNAL ADDRESS DATA BUS
NuBus INPUT BUFFER
NuBus DATA BUFFER
32
BUFFERED ADDRESS/ DATA
32
INTERNAL DATA BUS
NuBus ADDRESS/ DATA
NuBus OUTPUT BUFFER
32
NuBus CONTROL SIGNAL BUFFER
NuBus CONTROL SIGNAL BUFFER'
SOUND CONTROL AND SPEECH REGISTERS
MOUSE DATA INTERRUPT (TO EVENT GENERATOR)
CSMOUSE-
MOUSE DATA REGISTER
INTERRUPT (TO EVENT GENERATOR)
CSROM-
CON-FIGURATION ROM 8K BY 8 BITS
CSKYBD-
KEYBOARD ASYNCHRONOUS COMMUNICATIONS INTERFACE (USART)
INTERRUPT (TO EVENT GENERATOR)
CSACE-
RS-232C SERIAL DATA PORT
INTERRUPT (TO EVENT GENERATOR)
TRANSMIT DATA
RECEIVE DATA
8
16
BUFFERED DATA BUS
MDATA<07:00>
16
8
8
INTERRUPT INPUTS
EVENT GENERATOR
CSRF-
CON-FIGURATION REGISTER
CSPRINT-
PARALLEL PRINTER PORT
INTERRUPT (TO EVENT GENERATOR)
CSRAM-
PRINTER DATA
NON-VOLATILE RAM 2K BY 8 BITS
CSRTC-
REAL-TIME CLOCK
INTERRUPT (TO EVENT GENERATOR)
CSTIM-
PRO-GRAMMABLE INTERVAL TIMERS
INTERRUPT (TO EVENT GENERATOR)

Table 4-2 NuBus Signal Functions

Signal Signature	*Function*
Control Lines:	
CLK–	CLK– is the NuBus clock. CLK– synchronizes bus arbitration and data transfers between system modules on the NuBus. The clock signal has a nominal frequency of 10 megahertz with a 75 percent duty cycle. In general, signals are changed at the rising edge of CLK– and tested at the falling edge.
START–	Transfer start (START–) is driven for only one clock period by the current NuBus master at the beginning of a data transfer operation (transaction). START– indicates to the slaves on the NuBus that the address/data signal lines are carrying a valid address.
ACK–	Transfer acknowledge (ACK–) is driven for only one clock period by the addressed slave device on the NuBus. ACK– is asserted to indicate the completion of a transaction.
TM0–, TM1–	TM0 and TM1 are transfer mode signals asserted by the current NuBus master during start cycles to indicate the type of bus operation being initiated. The signals are also driven by NuBus slaves during acknowledge cycles to indicate the type of acknowledgment as shown below:

TM1–	*TM0–*	*Type of Acknowledgment*
L	L	Bus transfer complete
L	H	Error
H	L	Bus time-out error
H	H	Try again later

RESET–	RESET– has two functions, each of which is determined by its duration. When asserted for a single clock period, RESET– causes a NuBus interface initialization for all boards on the bus. When RESET– is asserted for more than one clock period, all boards on the bus are returned to their initial power-up state.
AD0–, AD1–	AD0– and AD1– carry address information during the start cycle of a byte transaction and carry control information during the start cycle of a nonbyte transfer.
Address/Data Lines:	
AD00– through AD31–	AD–< 31:00> are NuBus address/data signals that are multiplexed to carry a 32-bit address at the start of a cycle and 32 bits of data during the remainder of the cycle. These signals can be generated by any master on the NuBus.
Bus Arbitration Lines:	
RQST–	RQST– is a bus request signal that can be asserted by any NuBus master that wants control of the bus.

Table 4-2 NuBus Signal Functions (Continued)

Signal Signature	*Function*
ARB0– through ARB3–	ARB–< 03:00> form a binary arbitration code and are driven by contenders for the bus. The distributed arbitration logic samples the signals to establish which of the vying devices becomes the next master on the NuBus.
Card Slot Identification Lines:	
ID0– through ID3–	ID–< 03:00> are binary-coded slot identification signals that are hard-wired into each slot so that each installed board can be linked to its physical (slot) location.
Bus Parity Lines:	
SP–, SPV–	SP– transmits system parity information between NuBus devices that implement NuBus parity checking. SPV– is a system parity valid signal that indicates that SP– is being used. These signals are both wired high on the SI board and are not used.

NuBus Time-Out Logic

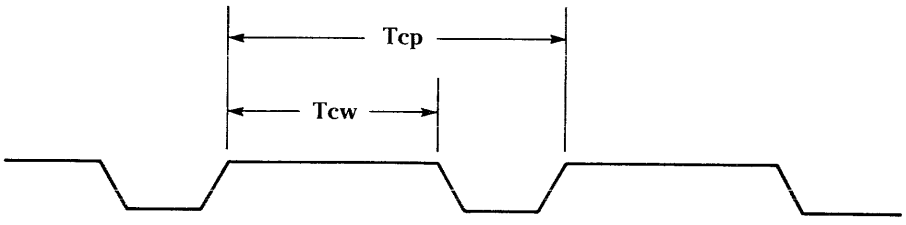
4.4.1.1 The SI board provides the NuBus time-out logic that monitors the NuBus and terminates any NuBus cycle that exceeds a predetermined number of NuBus clock periods, regardless of the master originating the cycle. The number of clock periods is fixed at 256. With a 10-megahertz NuBus clock, the time-out period is 25.6 microseconds. At the beginning of each NuBus cycle that begins with the assertion of the transfer start signal, a timer commences counting clock periods until either a transfer acknowledge signal is detected on the NuBus or the terminal count of 256 is reached. If the terminal count is reached without a transfer acknowledge, the NuBus time-out logic generates the transfer acknowledge signal and asserts the transfer mode signal TM0– to indicate a NuBus time-out error. If a transfer acknowledge signal occurs within the specified number of clock cycles, the timer resets and remains dormant until the next NuBus cycle begins.

NuBus Master Clock

4.4.1.2 A master NuBus clock circuit on the SI board generates the NuBus clock signal for all boards installed in the chassis. The clock circuit generates a 10-megahertz rectangular wave with a 75/25 duty cycle as shown in Figure 4-3. The nominal clock period is 100 nanoseconds.

Other boards have the ability to drive the NuBus clock lines. The SI board NuBus clock is applied to the backplane through a 0-ohm resister in series with the NuBus clock line driver output. This 0-ohm resister must be removed if the SI board is not to be the source for the NuBus clock.

Figure 4-3 NuBus Clock Timing Diagram



NOTE:
Tcp equals clock period
Tcw equals clock width.

NuBus Acknowledge Generator

4.4.1.3 The acknowledge generator provides a transfer acknowledge signal to the NuBus for each device on the SI board that can be read from or written to. This transfer acknowledge signal (ACK–) occurs upon completion of the device access time following the start of a NuBus cycle. The generated transfer acknowledge signal indicates to the NuBus that the SI board device has acknowledged the cycle.

SI Board Bus Structure

4.4.2 The NuBus accesses the SI board through the I bus on the SI board. As shown in Figure 4-2 (SI Board Functional Block Diagram), the I bus uses a 32-bit data bus, a 32-bit address bus, and various control lines to interconnect SI board devices. Signal signatures for the data bus and address bus are IDATA< 31:00> and IADDR< 31:00>, respectively. Board devices use the I bus for both on-board and NuBus data transactions. The I bus can accommodate byte, halfword, and full-word data transfers.

The buffered data bus, MDATA< 07:00>, is a buffered version of the least significant byte of the internal data bus. The buffered data bus interconnects metal-oxide semiconductor (MOS) devices on the SI board.

The graphics data bus, GRDB< 31:00>, and the graphics address bus, GRA< 19:00>, serve as the interface to the bit-mapped graphics subsection of the SI board. The local bus interface to the SI board is to the bit-mapped graphics subsection only.

Configuration ROM

4.4.3 A configuration ROM on the SI board is accessible from the NuBus to comply with Explorer system conventions. This ROM contains such information as the board serial number, cyclic redundancy check (CRC) signature, configuration ROM size, vendor identification, board type, board part number, revision level, and offsets to configuration registers and flag registers. The configuration ROM is organized into 8192 locations of 8 bits each. The ROM has 8 kilobytes of storage and resides at the highest address of control space allocated to the SI board with the upper 64 locations required for configuration data. The remaining locations extending from NuBus addresses FSFFE000 through FSFFFEFC (hexadecimal) are available for SI board diagnostic code. Any diagnostic code stored in the ROM must be downloaded to system memory for execution by the host processor. Table 4-3 provides a list of information that is contained in the configuration ROM. Addresses of the list are from the most significant address to the least significant address.

Table 4-3 Configuration ROM Contents

Description	*NuBus Address*	*Byte Size*	*Contents in Hexadecimal and Comments*
	MSB to LSB		Most significant byte to the least significant byte
Serial number	FSFFFFFC through FSFFFFDC	9	1. Mixed data format 2. Can be added as a last minute operation to an EPROM or PROM 3. Corresponds to bar code markings
Spare	FSFFFFD8	1	1. Contents FF
Revision level	FSFFFFD4 through FSFFFFC0	6	1. ASCII data format
CRC signature	FSFFFFBC through FSFFFFB8	2	1. Binary data format 2. Placed at end of covered data to ease calculation 3. Excludes serial number and revision number
ROM size	FSFFFFB4	1	1. Contents 0D 2. Binary data format 3. Equals Log2 of ROM size in bytes
Vendor ID	FSFFFFB0 through FSFFFFA4	4	1. Contents TIAU (TI Austin) 2. ASCII data format

Table 4-3 Configuration ROM Contents (Continued)

Description	*NuBus Address*	*Byte Size*	*Contents in Hexadecimal and Comments*
Board type	FSFFFFA0 through FSFFFF84	8	1. Contents SIB 00, 00, 00, 00, 00 2. ASCII data format (bytes 0 through 2); binary data format (bytes 3 through 7)
Part number	FSFFFF80 through FSFFFF44	16	1. Contents 00002236645-0001 (formerly 00002236590-0001) 2. ASCII data format PPPPPPPPPPP-DDDD 3. Sixteen characters, left justified
Configuration register offset	FSFFFF40 through FSFFFF38	3	1. Contents 40, 00, F0 2. The byte offset is from the start of the control space on this board to the first byte (lowest address) of the configuration register. ADDR=> FS000000+> offset
Device driver offset	FSFFFF34 through FSFFFF2C	3	1. Contents 00, 80, FF (none) 2. The byte is offset from the start of control space on this board to the first byte (lowest address) of the device driver routine. ADDR=> FS000000+> offset
Diagnostic offset	FSFFFF28 through FSFFFF20	3	1. Contents FF, FF, FF (none) 2. The byte is offset from the start of the control space on this board to the first byte (lowest address) of the diagnostic code. ADDR=> FS000000+> offset
Flag register offset	FSFFFF1C through FSFFFF14	3	1. Contents FF, FF, FF 2. The byte is offset from the start of the control space on this board to the first byte (lowest address) of the flag register. ADDR=> FS000000+> offset
ROM flags (PFWP, master, NuBus, self-test)	FSFFFF10	1	1. Contents 00 2. Binary data format
Layout byte	FSFFFF0C	1	1. Contents 02, which indicates that the ROM layout conforms to this document 2. Binary data format 3. The number increments for each revision of this document that affects the configuration ROM

Table 4-3 Configuration ROM Contents (Continued)

Description	*NuBus Address*	*Byte Size*	*Contents in Hexadecimal and Comments*
Test time	FSFFFF08	1	1. Contents FF implies no self-test 2. Binary data format 3. Equals Log2 of self-test time (seconds)
Reserved	FSFFFF04	1	1. Contents FF (set to > 000000 or > FFFFFF if not used)
Resource type	FSFFFF00	1	1. Contents 28 2. Binary data format
Diagnostic self-test area	FSFFFEFC through FSFFE000	1984	1. Diagnostic self-test code

NOTES:

The ROM is accessed by byte only, not words. Valid header addresses are FSFFFF00, –04, –08, ... , –F4, –F8, –FC. ROM data is valid only on the least significant (LS) byte of the NuBus.

The binary fields are stored such that the logically highest NuBus address of each field contains the most significant (MS) byte while the lowest contains the LS byte.

ASCII fields are stored as strings, with the first (most significant) character at the lowest address. Characters are stored one per word in byte 0, with contiguous word addresses.

Any field containing > 00..00 or > FF..FF indicates that the field is invalid, with the exception of the CRC (00 and FF are valid signatures).

Any unused field should be left unprogrammed.

The S in the NuBus address represents the slot number ID.

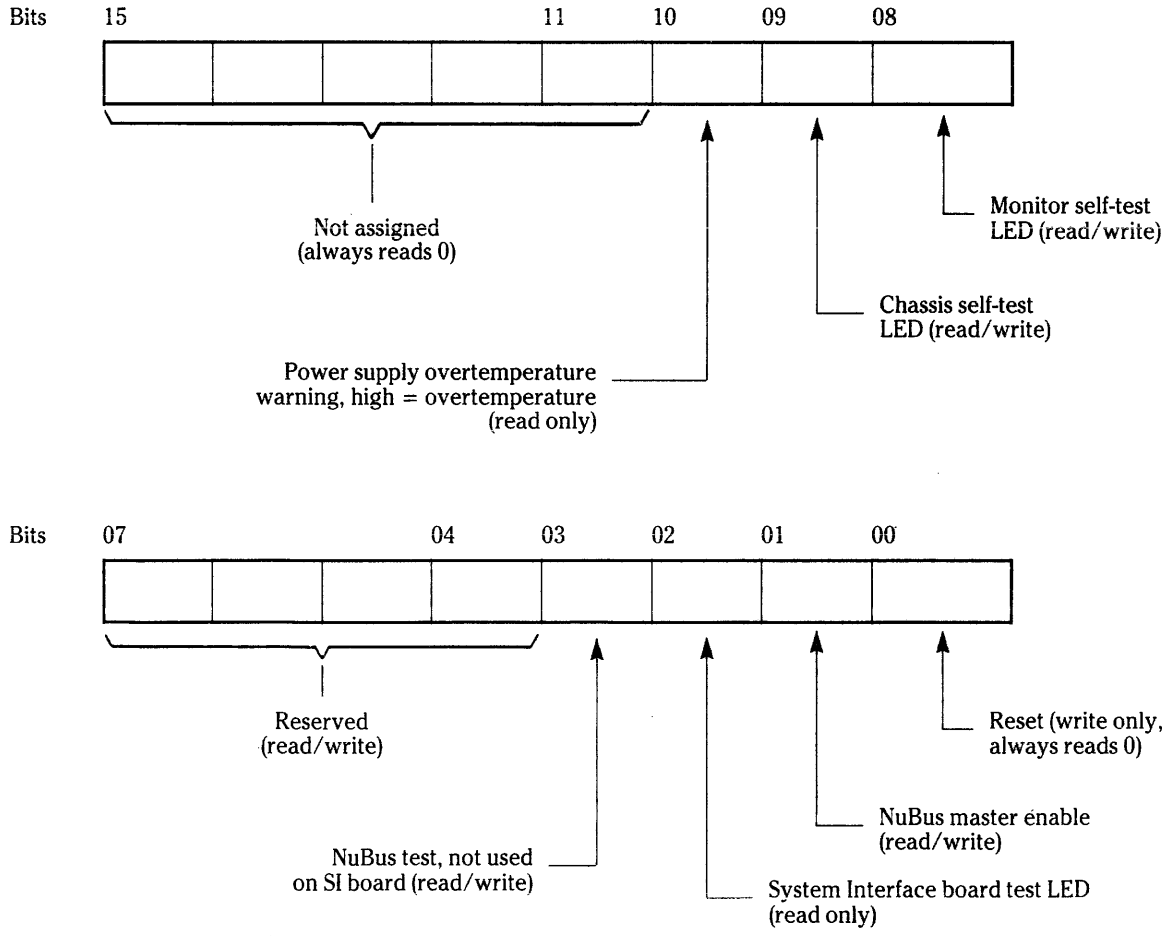
Configuration Register

4.4.4 The configuration register conforms to standard configuration register formats and requirements of Explorer systems. The SI board configuration register is a 16-bit register that resides at NuBus address hexadecimal FSF00040.

NOTE: The configuration register address is subject to change. Any program reference to the configuration register should go through the offset pointer stored in a reserved location of the configuration ROM.

Standard Explorer system bit assignments and special bit assignments for the SI board configuration register are shown in Figure 4-4.

Figure 4-4 Configuration Register Bit Assignments



The host processor must control configuration register bits 02, 08, and 09 because the SI board does not have processing capability for independent self-test and light-emitting diode (LED) test lamp control. All of these LED indicators light at power-up, and the processor must extinguish each of these LEDs at the successful completion of the applicable self-test. The chassis test LED is not implemented in the 7-slot chassis.

The chassis self-test bit (bit 09) provides for a system enclosure front panel fault indicator. If a fault indicator is implemented on the system enclosure, a driver on the SI board controls the indicator through a dedicated line in the backplane. However, current versions of the Explorer do not have a front panel fault indicator.

Bit 0 of the configuration register is the SI board reset bit. Writing a 1 to this bit resets the entire board except for the NuBus interface. NuBus acknowledge is returned after the reset has completed its cycle.

The event generator on the SI board has NuBus master capability, so the host processor must enable the master operation enable bit as part of the recovery from power-up or reset. The master operation enable bit should not be enabled until after the event addresses are programmed. As an absolute minimum, the overtemperature and power failure warning event addresses must be programmed before the master operation enable bit for the SI board is enabled.

Event Generator

4.4.5 The event generator monitors other circuit devices on the SI board for generated interrupts. It also monitors the computer system power supply for overtemperature and power failure warning signals. When enabled, the event generator responds to circuit interrupts by posting an event to any preprogrammed NuBus word address (any modulo-4 byte address, hexadecimal 00000000 through FFFFFFFC). The event generator posts the event by writing a hexadecimal FF byte to the preprogrammed address upon receipt of the interrupt. In summary, the event generator performs the following functions:

- Stores event addresses provided by the host processor
- Polls each of the monitored circuit devices and power supply signals for an interrupt condition
- Responds to an interrupt condition by:
  - Becoming a NuBus master and acquiring access to the NuBus
  - Writing a hexadecimal FF byte to a preprogrammed address within the NuBus memory space that corresponds to the interrupting device
  - Acknowledging the interrupt to the interrupting device

There is no priority encoding for the interrupting devices, except for the power failure warning signal, because each device is polled in sequence. If a power failure warning signal occurs, the event generator immediately posts the event, regardless of other inputs.

The preprogrammed addresses for posting of events are stored in a 16-location, 32-bit event generator register file. Table 4-4 shows the interrupting devices, the interrupting causes, and the memory locations in the event generator register file in which the preprogrammed addresses for posting of events are stored.

Event Generator Programming

4.4.5.1 Each of the events listed in Table 4-4 has an associated vector address where the event is to be posted. The vector addresses are stored in the 16-location event generator register file at memory locations beginning with address hexadecimal FSF00000 through FSF0003C. A write operation to each of these locations stores the address where the event is to be posted when it occurs. You can store or alter the address at any time by performing a write operation of four bytes, two halfwords, or one full word.

Table 4-4 Event Causes and Register File Storage Locations

Device	*Cause for Interrupt*	*Memory Location (Hexadecimal)*
Real-time clock	Time interrupt	FSF00000
Interval timer (short)	Interval elapsed	FSF00004
Interval timer (long)	Interval elapsed	FSF00008
RS-232C serial port	Status interrupt	FSF0000C
Parallel printer port	Printer acknowledge	FSF00010
Graphics controller	Command acknowledge	FSF00014
Keyboard USART	Ready to transmit/receive	FSF00018
Power supply	Overtemperature	FSF0001C
Keyboard chord reset	Operator entry	FSF00020
Mouse interface	Mouse motion detected	FSF00024
Mouse interface	Mouse keyswitch change	FSF00028
Voice interface	Voice data present	FSF0002C
Sound interface	Sound parity error	FSF00030
Fiber-optic data link*	Fiber-optic link warning	FSF00034
Power supply	Power failure warning	FSF00038
Power supply	Power failure warning	FSF0003C

NOTE:

* The fiber-optic link warning is only available on part number 2236645-0001. On earlier SI boards, such as part number 2236590-0001, there are three power-failure warning interrupts and no fiber-optic link warning.

Event Generator Initialization

4.4.5.2 After power is applied to the system and a reset pulse is generated, the event generator is latched in a standby mode so that no events can be generated. After the event vector addresses have been written into the event generator register file, the enabling of the SI board as a NuBus master device through the configuration register allows the event generator to begin polling sources and posting events as required. Writing a 1 to the NuBus master enable bit in the configuration register enables event posting. Writing a 0 to the NuBus master enable bit disables all event posting from the event generator.

The event generator should be enabled early in the SI board initialization sequence. However, some of the programmable interrupting devices can come up in a random state and generate spurious events. It is important that these devices be initialized before the event generator is enabled to begin posting of events. Also, the SI board must be reset through the configuration register after initialization of these devices. The recommended sequence for SI board initialization is as follows:

1. Write a control byte to the short-term interval timer
2. Write a control byte to the long-term interval timer
3. Disable and clear interrupts from the real-time clock
4. Reset the SI board
5. Load event addresses into the register file
6. Enable the event generator for posting of events

Event addresses stored in the event generator register file are volatile and must be reloaded after each power-up or reset. Also, event addresses must be word addresses (that is, modulo-4 byte addresses) or the event byte is not interpreted properly by the central processor.

Nonvolatile RAM

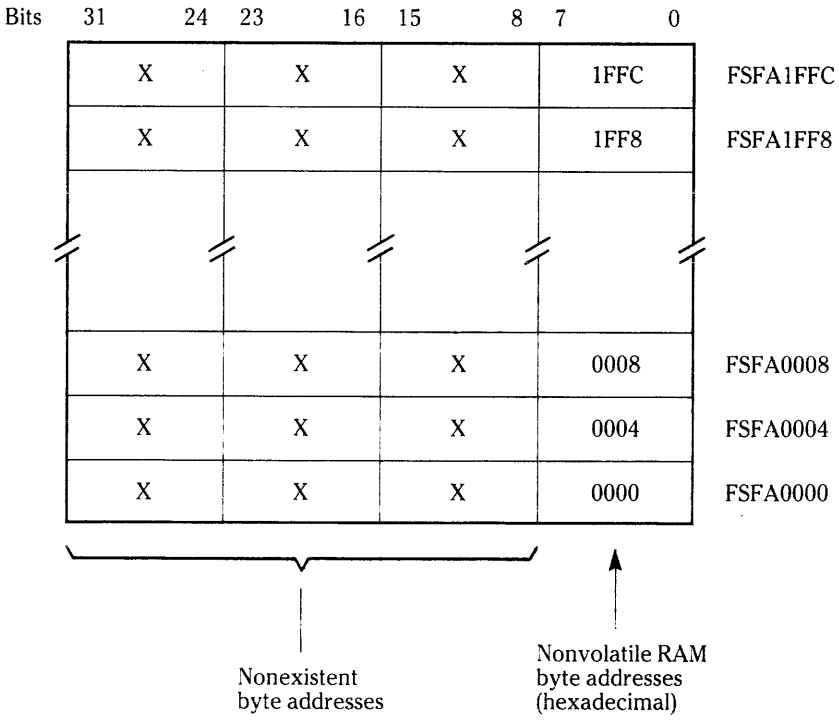
4.4.6 The nonvolatile RAM on the SI board stores computer system operational parameters before and during a power failure to facilitate recovery from the power failure. The RAM is also addressed by software to store environmental data such as an overtemperature condition detected in the power supply before a power failure. The nonvolatile RAM is organized into 2048 locations of 8 bits each. As shown on the block diagram, the nonvolatile RAM is addressed for read and write operations by the internal address bus. Data to and from the RAM is transferred over the 8-bit buffered data bus.

In normal operation, the RAM is powered by 5 volts provided by the system power supply. A factory-installed 3-volt lithium battery provides backup power to the RAM. No battery current is drawn during normal power supply operation. If the power supply voltage drops to below about 4.5 volts, the nonvolatile RAM switches to a write-protect mode. As the power supply voltage decays further to about 3 volts, the RAM switches over to battery power. This is the data retention mode and no read or write requests to the RAM are honored. Assuming the backup battery voltage is good, the nonvolatile RAM returns to normal operation when the power supply voltage recovers to about 4.75 volts.

No data is lost when the transition from normal to backup power occurs. The backup battery is capable of powering the RAM in the data retention mode for at least one year after normal system power has been been removed. If the backup battery voltage drops below about 2.2 volts, the first write operation after recovery of normal system power is locked out. Self-test reads a location, modifies the data, and then reads the data again. The data remaining unmodified indicates that the battery is low and a battery voltage-low message is originated.

Figure 4-5 is a memory map of the nonvolatile RAM that shows the relationship between NuBus addresses and RAM addresses. Note that RAM addresses correspond to NuBus word addresses though only one byte of data is stored per address location. The nonvolatile RAM NuBus address space extends from hexadecimal FSFA0000 through FSFA1FFC.

Figure 4-5 **Nonvolatile RAM Memory Map**



Real-Time Clock

4.4.7 The real-time clock includes an addressable real-time counter; 56 bits of RAM with a comparator to compare the real-time counter to the RAM data; and an interrupt output. The time base for the real-time clock is a 32 768-hertz crystal oscillator. The real-time clock shares the backup battery with the nonvolatile RAM and can retain data for at least one year using backup power. As shown in Figure 4-2, the real-time clock is addressed for read and write operations by the internal address bus. Data to and from the real-time clock is transferred over the 8-bit buffered data bus.

Real-Time Counter

4.4.7.1 The real-time clock has a set of eight timing registers that contain time-of-day and date information. The host processor can program the real-time clock for the current time, the day of the week, the day of the month, and the month in a 24-hour clock format. The real-time clock can generate an interrupt to the event generator under any of the following conditions:

- 10-hertz (100 millisecond) interval
- 1-hertz (1 second) interval
- 1-minute interval
- 1-hour interval
- 1-day interval
- 1-week interval
- 1-month interval
- At any specific time and/or date programmed into the real-time clock

Real-Time Clock Programming

4.4.7.2 As shown in Table 4-5, the real-time clock consists of 24 eight-bit registers (including the 8 timing registers) that are addressed at NuBus addresses hexadecimal FSF80000 through FSF8005C. Registers 22 and 23, the last two registers, are not implemented.

Timing Each timing register (registers 0 through 7) contains one or two 4-bit digits representing the current time count. Each digit represents a binary-coded decimal (BCD) number as defined in Table 4-6. The most significant (tens) BCD number is read from the D7 (most significant bit) through D4 outputs of the timing register. The least significant (units) BCD number is read from the D3 (most significant bit) through D0 outputs of the timing register. RAM registers (registers 8 through 15) have the same format as their corresponding timing registers. Table 4-6 indicates the maximum BCD number that each digit can reach during normal counting.

Table 4-5 **Real-Time Clock Register Addresses and Functions**

Register Address (Hexadecimal)	*Register Number*	*Function*
FSF80000	0	100-nanoseconds counter
FSF80004	1	10-milliseconds and 100-milliseconds counter
FSF80008	2	Seconds counter
FSF8000C	3	Minutes counter
FSF80010	4	Hours counter
FSF80014	5	Day-of-the-week counter
FSF80018	6	Day-of-the-month counter
FSF8001C	7	Month counter
FSF80020	8	RAM — 100-nanoseconds unit
FSF80024	9	RAM — 10-milliseconds and 100 milliseconds units
FSF80028	10	RAM — seconds
FSF8002C	11	RAM — minutes
FSF80030	12	RAM — hours
FSF80034	13	RAM — day of the week
FSF80038	14	RAM — day of the month
FSF8003C	15	RAM — months
FSF80040	16	Interrupt status
FSF80044	17	Interrupt control
FSF80048	18	Counters reset
FSF8004C	19	RAM reset
FSF80050	20	Status bit
FSF80054	21	GO command

Table 4-6 **Timing Register and RAM Register Format**

Register Number	*Function*	*Maximum Binary-Coded Decimal Tens (D7-D4)*	*Units (D3-D0)*
0 and 8	100 nanoseconds	9	Not used
1 and 9	10 milliseconds (units)		9
	100 milliseconds (tens)	9	
2 and 10	Seconds	5	9
3 and 11	Minutes	5	9
4 and 12	Hours	2	9
5 and 13	Day of week	0	7
6 and 14	Day of month	3	9
7 and 15	Month	1	9

Interrupts The interrupt output of the real-time clock is programmed by writing interrupt enable information to the write-only interrupt control register (register number 17). Table 4-7 shows the function of each bit in the interrupt control register. Interrupts are enabled by setting the desired bit to one and disabled by setting the bit to zero. Note that more than one interrupt can be enabled at the same time. As shown in the table, bit D0 is the compare bit. When this bit is set to one, an interrupt is generated when the current value in the timing registers (registers 0 through 7) is equal to the value programmed into the RAM registers (registers 8 through 15). The RAM registers are programmed using the same format as is used for the timing registers.

Interrupt Status Register After one or more enabling bits have been set in the interrupt control register, interrupts are generated as programmed. An interrupt generates a high at the interrupt output of the real-time clock and causes a bit to be set in the interrupt status register (register 16) that corresponds to the enabling bit in the interrupt control register (see Table 4-7). The source of the interrupt can be determined by reading the interrupt status register. Reading this register clears and resets the interrupt output of the real-time clock.

Counter and RAM Reset Registers The timing registers can be reset by writing all ones (hexadecimal FF) to the counter reset register (register number 18). The RAM registers can be reset by writing all ones (hexadecimal FF) to the RAM reset register (register number 19).

Status Bit Register Bit D0 in the status bit register (register 20) is a status bit that indicates that data read from the timing registers may be invalid. The 1-kilohertz input to the millisecond counter has a pulse width of 61 microseconds. The status bit is set if a read operation of the real-time clock is made during this 61 microseconds. This tells the user that the clock is rippling through the real-time clock and data that was read may be invalid. The status bit should be read every time a counter is read or a series of counters are read. If the status bit is set, the read operation should be repeated. Reading the status bit register resets the status bit.

Table 4-7 Interrupt Control Register and Interrupt Status Register Format

Bit	*Function*
D0	Compare
D1	10 hertz
D2	1 hertz
D3	1 minute
D4	1 hour
D5	1 day
D6	1 week
D7	1 month

GO Command Register A write pulse to the GO command register (register 21) resets the 1-millisecond, 10-milliseconds, 100-milliseconds, seconds, and minutes counters. This command is used to start the clock at a precise time. Data on the bus is ignored during the write operation. If the seconds counter is at a value greater than 40 when the GO command is issued, the minute counter increments; otherwise the counter does not increment. The GO command is not necessary to start the clock, but it is a convenient way to start the clock precisely at a given time.

Programmable Interval Timers

4.4.8 The interval timing device as implemented on the SI board uses three 16-bit counters to provide two programmable interval timers within a single integrated circuit. One timer is a short-term interval timer with a programmable interval range of 1 microsecond to 65.536 milliseconds with 1 microsecond resolution. The other, a long-term interval timer, has a programmable range of 2 microseconds to 1.19 hours. The interval timers are addressed for read and write operations by the internal address bus. Data to and from the interval timers is transferred over the 8-bit buffered data bus. Interrupts generated by the interval timers are applied to the SI board event generator.

A 1-megahertz clock derived from the NuBus clock drives counter 0 to provide the short-term interval timer range. The long-term interval timer consists of separately programmable rate generator and counter circuits. Counter 1, also driven by the 1-megahertz clock, provides a rate generator output. The output of counter 1 drives the long-term interval counter, counter 2. The division ratio programmed into the rate generator determines the resolution and the range of the long-term interval timer. Rate generator output periods can be programmed to vary from 2 microseconds (0.5 megahertz) to 65.536 milliseconds (15.259 hertz). The counter of the long-term interval timer can be programmed for delays that vary from the rate generator period multiplied by one (2 microseconds) to the rate generator period multiplied by 65 536 (1.19 hours). In summary, the programming of both counters 1 and 2 determines the minimum to maximum interval range of the long-term interval counter.

NOTE: In practice, the interval timers should not be programmed for intervals of less than 20 microseconds. Accuracy of the programmed interval deteriorates when load times for the interval counter hardware and NuBus access times for generating and servicing the timer-expired event become a significant part of the programmed interval.

Interval Timer Hardware

4.4.8.1 The interval timing device is a versatile general-purpose timer that includes three independent 16-bit counters with separately-selected operating modes. A control register for each counter stores a byte of data that specifies the operating mode, the count type (binary-coded decimal or binary), and the load/read sequence for its associated counter. Because the control register specifies a sequence for loading or reading the content of the counter, a control word must precede a counter load operation. Also, the counter load must follow the sequence specified by the control word. The load operation is not complete until one rising edge and one falling edge of the 1-megahertz clock occurs after the write operation.

Outputs from the interval timers are undefined until the counters are initialized with operating mode information. Therefore, the control loading sequence for the interval timers must occur before the event generator on the SI board is enabled.

Data is transferred to and from the counters in counter load and counter read operations over the 8-bit buffered data bus, which corresponds to the eight least significant bits of the internal data bus of the SI board. Since counter read operations require a control byte to specify the order of the byte readback, it is necessary to complete a specified read operation before attempting any other timer operation.

A counter latching feature allows any of the counters to be read without disturbing ongoing counting operations. In this type of read, a special code causes the contents of a counter to be dumped to a storage register for readback.

The programmable interval timers are capable of five modes of operation, but only two of these modes are used. Counter 1 operates in the square-wave rate generator mode (mode 3) to provide the rate generator input to counter 2, the long-term interval counter. Counters 0 and 2 both operate in the interrupt on terminal count mode (mode 0). These two modes of operation are described in the following paragraphs.

Mode 0 Operation (Interrupt on Terminal Count) Both programmable interval timer outputs (counters 0 and 2) operate in the interrupt on terminal count mode (mode 0). The interrupt outputs of the counters are undefined until a control byte is loaded into the associated control register that selects the counter and the mode of operation. Mode 0 operation forces the interrupt output of the counter to its inactive state. The output remains inactive until the count is loaded into the 16-bit counter and decremented to zero. The control byte also sets up the counter to accept and load the count either into the most significant byte position of the counter, into the least significant byte position of the counter, or into the least significant byte position followed by the loading of the most significant byte position. The loading of the counter must occur in accordance with instructions provided in the control byte. A loading of all zeros in the 16-bit counters causes a maximum count. The loading of the counter does not have to occur immediately after the control byte is loaded into the control register. The delay can be indefinite as long as system power remains applied.

After the counter is loaded, it begins counting down. The counter issues an interrupt to the event generator when the counter contents decrements to zero. The interrupt remains active until either a new control byte clears the interrupt or the counter is reloaded.

NOTE: The event generator does not generate multiple events based on an interrupt that remains active. The event generator produces a single event each time a counter interrupt is issued.

A counter can be reloaded without loading a new control byte into its control register. However, the loading of the counter must follow the sequence as required by existing control byte information in the control register. A reload of the counter clears the existing interrupt and starts a new decrement of the counter.

Mode 3 Operation (Square-Wave Rate Generator) Counter 1, which drives the clock input of the long-term interval counter, operates in mode 3 as a programmable-period clock generator. The output waveform is programmed in terms of periods of the 1-megahertz clock input to the counter. When an even value of N is loaded into the counter, the output waveform is high for half the time the counter is decrementing and low for the other half. The result is a square wave with a period equal to the time required for the number of 1-megahertz input counts loaded into the counter to decrement to zero. The input count automatically reloads at the end of the counter period and the cycle repeats indefinitely. If the load value N is odd, the output waveform is high for (N + 1)/2 clock periods and then low for (N − 1)/2 clock periods. The duty cycle is not 50 percent, but the total period is accurate. Again, the load value automatically reloads at the end of each counter period.

An external load does not affect the output waveform until the current period is complete. This assures a predictable response. The rate generator does not count until after the first load operation. In this manner the load operation serves to synchronize the counter to software operations.

Interval Timer Programming

4.4.8.2 The complete functional operation of the interval timing device is controlled by system software. First, a control byte sent to the control register by the host processor initializes each counter with the desired control information after which time the loading or reading of a counter can occur. Each of these operations is discussed separately. Additionally, examples are provided for typical programming of the control registers and counters, both at power-up initialization and during regular operation.

Control Register Programming A control byte written to the control register is used to initialize a selected counter with a desired mode of operation, read/load sequence, and selection of binary or binary-coded decimal counting. Writing of the control byte can be in any sequence of counter selection. That is, counter 0 does not have to be programmed first or counter 2 last. The control register for each counter has a separate address selected by SC1 and SC0 of the control word so that its loading is completely sequence independent. The control register is addressed at the base address of the interval timer plus 0C (hexadecimal FSF9000C). Control byte format and definitions of control are shown in Figure 4-6.

Figure 4-6 **Control Byte Format**

D7 D0

SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Definitions of control:

SC — Select counter

SC1	SC0	
0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Illegal instruction

RL — Read/load

RL1	RL0	
0	0	Counter latching (for latched read operation)
0	1	Read/load least significant byte
1	0	Read/load most significant byte
1	1	Read/load least significant byte first, then most significant byte

M — Mode

M2	M1	M0	
0	0	0	Mode 0 (interrupt on terminal count)
0	0	1	Mode 1 (not required)
X	1	0	Mode 2 (not required)
X	1	1	Mode 3 (square-wave rate generator)
1	0	0	Mode 4 (not required)
1	0	1	Mode 5 (not required)

BCD — Binary or BCD counting

BCD	
0	Binary 16-bit counter
1	BCD (not required)

Programming the Counters The interval timing device consists of four 8-bit registers that are accessible at NuBus addresses beginning with hexadecimal FSF9000 continuing through FSF9000C. Three of the registers are used to load and read the count registers. The fourth register is used to program the control register with the control byte as described in the previous paragraph. Table 4-8 shows the addresses and functions of the registers.

Though the writing of the control byte can be in any sequence of counter selection, loading and reading of the actual count value of an associated counter must be done in the exact sequence programmed into the read/load instructions of the control byte. That is, if the control byte instructions call for a counter to be loaded with the least significant byte followed by the most significant byte, then that sequence must be followed exactly.

Table 4-8 Interval Timer Register Addresses and Functions

Read/Write	*Address (Hexadecimal)*	*Function*
Write	FSF90000	Load counter 0 (short-term interval timer)
Write	FSF90004	Load counter 1 (square-wave rate generator)
Write	FSF90008	Load counter 2 (long-term interval timer)
Write	FSF9000C	Write control byte to control register
Read	FSF90000	Read counter 0
Read	FSF90004	Read counter 1
Read	FSF90008	Read counter 2
Read	FSF9000C	No operation. High-impedance output to 8-bit buffered data bus; no read of control register occurs.

Interrupt outputs of the interval timer are undefined until the control bytes for counters 0 and 2 are initially loaded. Therefore, it is necessary to send control bytes to these registers on power-up before the event generator is enabled. The control bytes order the counters to a stable mode. Otherwise, the SI board could generate spurious events on power-up. Do not load the counters at this time or they will begin counting. The read/load field determines the form that an eventual load or read operation must take. The simplest form is when RL1 and RL0 of the control byte are both one. This specifies that the loading or reading of the specified counter occurs in a least significant byte followed by the most significant byte order.

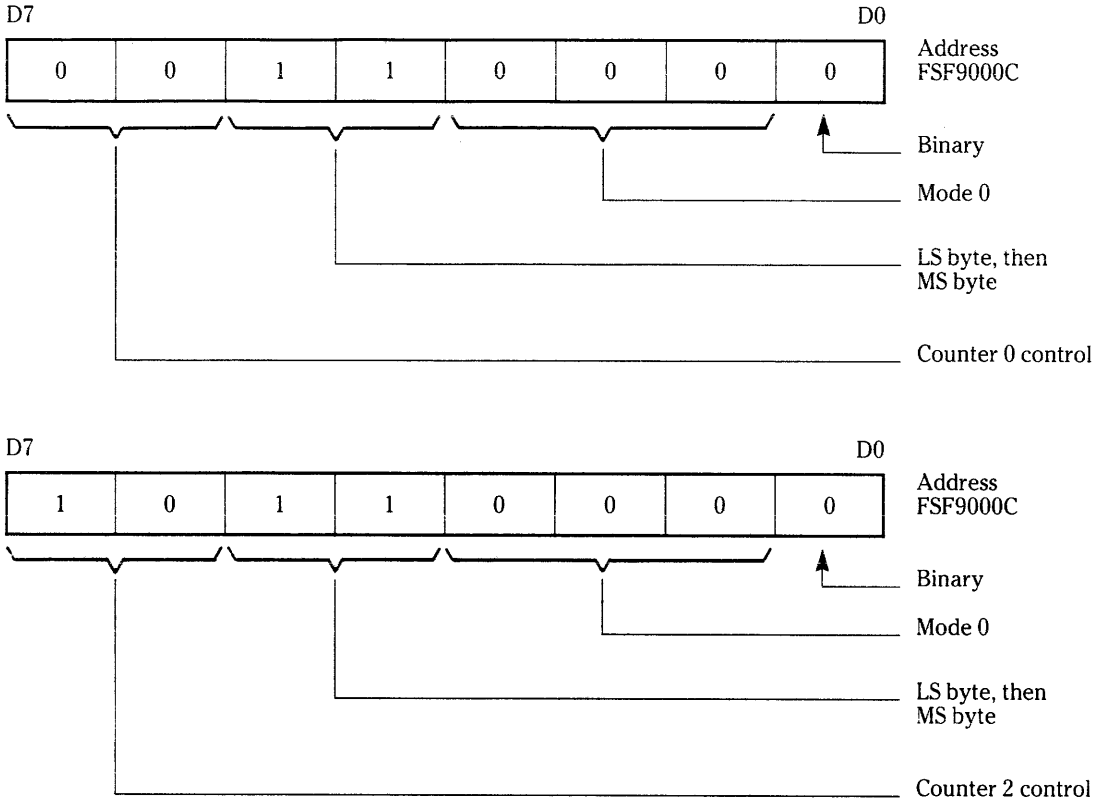
Counter 1 cannot trigger an event and need not be initialized on power-up. However, counter 1 must be loaded and running before the long-term interval timer starts.

In summary, the correct order of operations at power-up is as follows:

1. Power-up
2. Write control bytes to control registers for counters 0 and 2
3. Disable real-time clock interrupts
4. Clear real-time clock interrupt register
5. Reset SI board
6. Load event addresses into event generator file
7. Enable event generator
8. Load interval timer counters

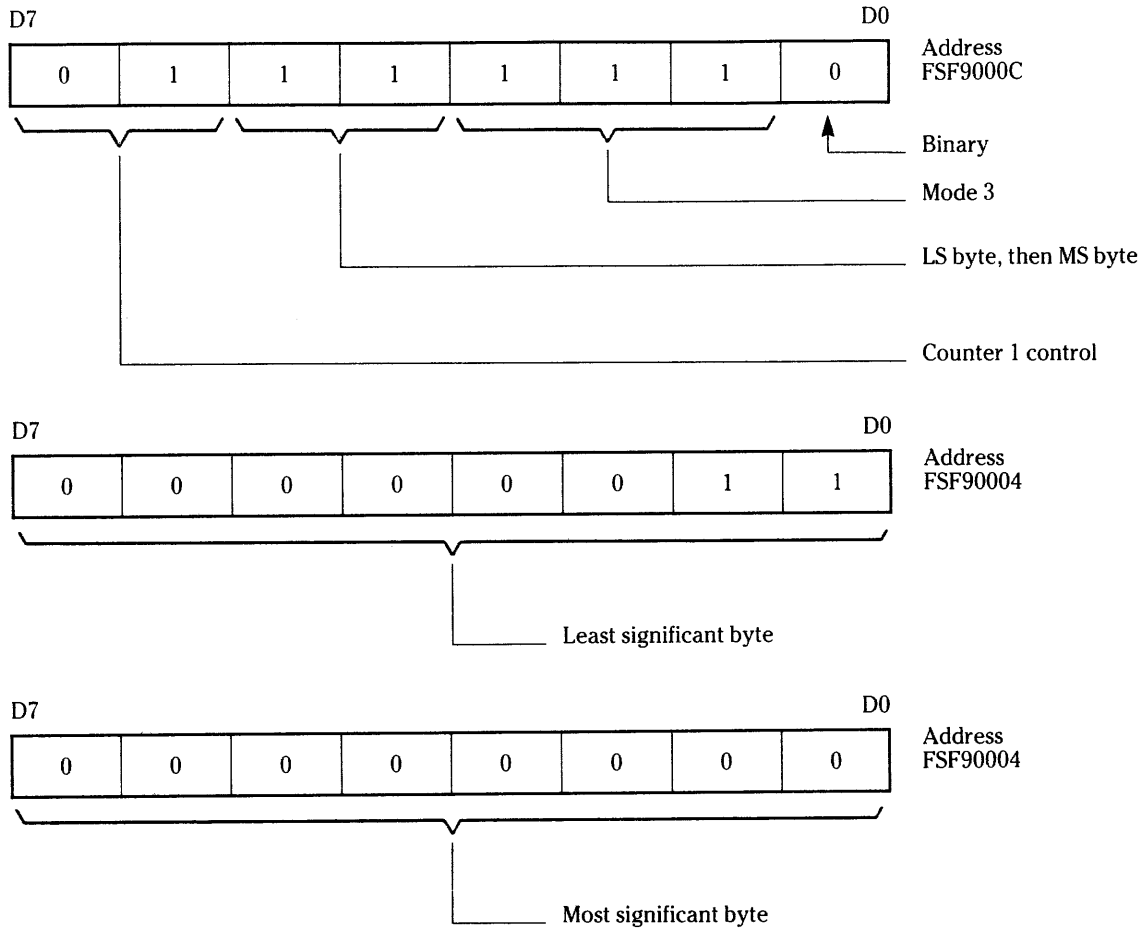
Interval Timer Initialization Figure 4-7 shows a typical format for the control bytes sent to the control registers for counters 0 and 2 at power-up initialization. Note that both control bytes are addressed at hexadecimal FSF9000C.

Figure 4-7 Typical Interval Timer Initialization at Power-Up



Loading and Starting the Interval Timers In this description of typical loading and starting procedures for the interval timers assume that all start-up operations are complete and that the SI board is operating with the event generator enabled. Further assume that the need is to start a 7-microseconds interval timer and a 15-microseconds interval timer. This requires that both the short-term interval timer (counter 0) and the long-term interval timer (counter 2) be used. The square-wave rate generator (counter 1) must also be operating before the long-term interval timer since its output drives the long-term interval timer. The first step in deriving the 15-microseconds interval is to start the square-wave rate generator running at a 3-microseconds period as shown in Figure 4-8. Note that a control byte first addresses the control register for counter 1 and that the least significant and most significant bytes of counter 1 are loaded with the count data of 3.

Figure 4-8 Starting the Square-Wave Rate Generator

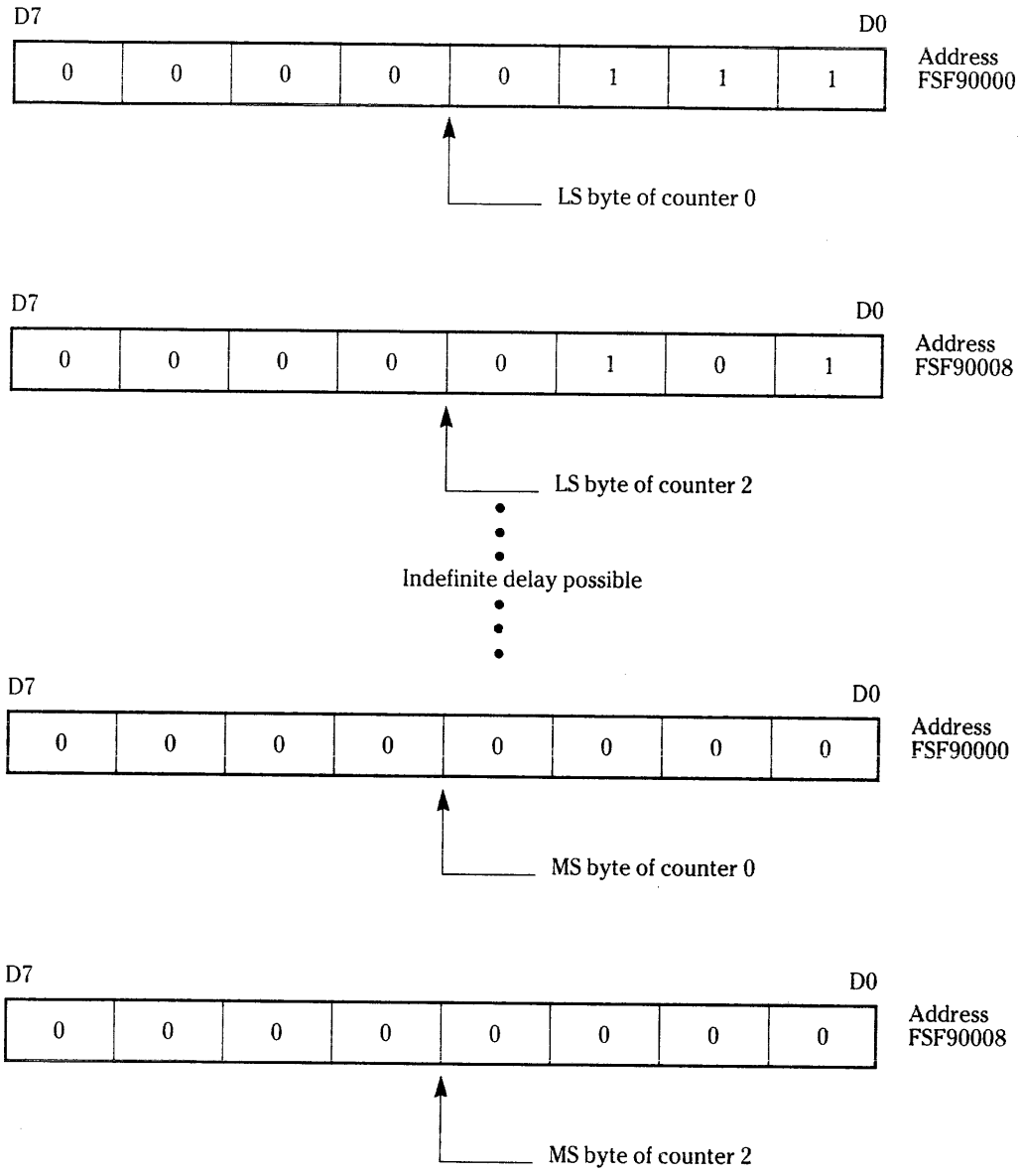


At power-up initialization, control registers for counters 0 and 2 were loaded with mode of operation data as shown in Figure 4-7. The first step in generating the 15-microseconds long-term interval was to start the square-wave generator to provide a 3-microseconds period signal to drive counter 2 (Figure 4-8).

To complete the operation of providing the 7-microseconds short-term interval and the 15-microseconds long-term interval, counters 0 and 2 must be loaded and synchronized. Counter 0 is loaded with 7 to provide the 7-microseconds short-term interval (7 times the period of the 1-megahertz driving signal to counter 0). Counter 2 is loaded with 5 to provide the 15-microseconds long-term interval (5 times the period of the 3-microseconds period of the square-wave generator driving signal to counter 2). To achieve minimum skew in the start times, the load operations for the counters are interleaved as shown in Figure 4-9.

To stop an interval timer, either address a new control byte to its control register or reload the least significant byte of the counter. The counter does not restart until the counter is completely loaded.

Figure 4-9 Starting the Interval Timers



Parallel Printer Port

4.4.9 The parallel printer port is an SI board interface that transmits 8-bit parallel data and handles control and handshake signals between the SI board and a Centronics-compatible external device (usually a printer). The interface logic consists of an output control register, an output data register, a loopback data register, and a status buffer. The port can transmit data at speeds up to 100 kilobytes per second. A software-enabled interrupt provides the option of event-triggered or polled-printer status updates.

Parallel Printer Port Interface Signals

4.4.9.1 The parallel printer port signals are brought to connector P3 on the SI board, where the fiber-optic board serves as a physical adapter and extender between a DIN electrical connector at the rear of the backplane and the parallel printer cable connector. Table 4-9 lists the parallel printer port input and output signals.

Table 4-9 Parallel Printer Port Signals

Signal	*In/Out*	*Description*
DATSTRB–	Out	Data strobe. The falling edge of DATSTRB– indicates that the data is valid. The minimum duration of the strobe low pulse is 2 microseconds.
DAT< 8:1>	In/out	Data out to printer, data in to loopback data register. DAT1 is the least significant data bit, DAT8 is the most significant data bit.
PACK–	In	Acknowledge pulse returned by the printer to indicate that the last character has been received and that the next character can be sent. The pulse is also returned each time the printer goes from the offline to the online state.
BUSY	In	Active level returned by the printer to indicate that it is not able to receive characters. BUSY is generally asserted on the leading edge of the data strobe pulse and remains on until the printer is able to receive characters.
PE	In	Paper out error. PE is asserted by the printer to indicate that the machine is out of paper. FAULT– is also asserted and the SELECT (selected) signal is released.
SELECT	In	Printer selected. SELECT is asserted to indicate that the printer has been selected and can print data if BUSY or FAULT– is not asserted. SELECT means that the printer is online.
AUTOF–	Out	Automatic feed. This command advances the paper feed mechanism on each carriage return code. The command is not required by all printers.
INIT–	Out	Initialize. INIT– is asserted by the host processor to reset the printer logic and initialize the operation.
FAULT–	In	Fault condition. FAULT– is asserted by the printer to indicate that the printer cannot operate due to a fault condition at the printer. Possible faults include paper out, data buffer overflow, or paper jam detected.

Parallel Printer Port Programming

4.4.9.2 For programming purposes, the parallel printer port can be considered to be two 8-bit read/write registers, register 0 and register 1. Register 0 is the data register addressable at address hexadecimal FSF10000. Register 1 is the control and status register at address hexadecimal FSF10004. Bit assignments for the registers are given in Table 4-10 and Table 4-11.

The parallel printer port can be operated in either the polled mode or the interrupt mode. To send a character to the printer in the polled mode, perform the following procedure:

1. Read register 1 (address FSF10004) to ensure that the printer is not busy, the paper is installed, the printer is selected, and no fault has occurred (bit 0 equals 0, bit 1 equals 0, bit 2 equals 1, and bit 3 equals 1).

2. Write the data byte to register 0 (address FSF10000).

3. Assert the data strobe by writing hexadecimal 05 to register 1 (address FSF10004).

4. Deassert the data strobe by writing hexadecimal 07 to register 1 (address FSF10004).

5. Poll register 1 (address FSF10004), bit 0. This bit will remain set (1) until the printer can accept the next character. Note that the other bits in this register should be checked for errors. Once bit 1 becomes reset (0), the previous sequence can be repeated for additional characters.

Table 4-10 Parallel Printer Port Register 0 Bit Functions

Bit	*Function*
0	Data bit 0 (least significant bit)
1	Data bit 1
2	Data bit 2
3	Data bit 3
4	Data bit 4
5	Data bit 5
6	Data bit 6
7	Data bit 7 (most significant bit)

Table 4-11 Parallel Printer Port Register 1 Bit Functions

Bit	*Read Operation*	*Write Operation*
0	Busy	Automatic feed (active low)
1	Paper out error	Data strobe (active low)
2	Select (online)	Initialize (active low)
3	Fault (active low)	Interrupt enable
4	Not used	Not used
5	Not used	Not used
6	Not used	Not used
7	Not used	Not used

To send a character in the interrupt mode, perform the following sequence:

NOTE: It is assumed that the SI board has been properly initialized and the event generator has been enabled.

1. Ensure that the desired event posting address has been written into the event generator register file at address FSF00010.
2. Read register 1 (address FSF10004) to ensure that the printer is not busy, the paper is installed, the printer is selected, and no fault has occurred (bit 0 equals 0, bit 1 equals 0, bit 2 equals 1, and bit 3 equals 1).
3. Write the data byte to register 0 (address FSF10000).
4. Enable the interrupts by writing hexadecimal 0F to register 1 (address FSF10004).
5. Assert the data strobe by writing hexadecimal 0D to register 1 (address FSF10004).
6. Deassert the data strobe by writing hexadecimal 0F to register 1 (address FSF10004).

7. When the character has been received by the printer and is ready for the next character, an interrupt will be posted by the SI board at the event generator address programmed in step 1.

8. Clear the interrupt by writing 07 into register 1 (address FSF10004). Repeat this procedure for additional characters.

Note that the current character that has been written into register 0 can be read at any time at the same address (FSF10000).

Bit-Mapped Graphics Display Controller

4.4.10 The bit-mapped graphics display controller consists of an 8K-word by 64-bit bit-mapped memory, a NuBus interface, a local bus interface, address latches, a logical operation unit, a CRT controller, shift register latch, shift register, and video generator as shown in Figure 4-10. The basic operation of each of these display controller functions is described in the following paragraphs. Additionally, programming of the bit-mapped graphics display controller is described.

Bit-Mapped Memory Functional Description

4.4.10.1 The bit-mapped memory stores an image of the information displayed on the video display. The memory uses 16 high-speed 8K by 8-bit static RAMS to provide a single-plane video map of 1024 by 1024 picture elements (pixels). Of these, 1024 by 808 pixels are displayed on the video display. The display controller must access the bit-mapped memory in order to refresh the video display. These accesses are shared with the processor memory accesses in such a way as to maximize processor access without sacrificing video display quality.

NOTE: The bit-mapped memory is not intended to be used as a general purpose memory resource and should not be used for any purpose other than as a video bit-mapped memory. Memory cycles to the bit-mapped memory cannot be locked from either the NuBus or the local bus and should never be used for any data structure requiring a locked memory cycle.

The processor accesses the bit-mapped memory using either the local bus or the NuBus. If a processor has a local bus interface and the addressed SI board is in a slot that is serviced by that local bus, then all transfers between the processor and the bit-mapped memory occur over the local bus. If either the SI board or the processor is not connected to the local bus, then transfers between the processor and the bit-mapped memory occur over the NuBus. In the Explorer system, all memory transfers from the host processor originate on the local bus. If the addressed memory is not accessible on the local bus, one of the memory boards in the system transfers the request to the NuBus to complete the memory cycle. Connecting the SI board directly to the local bus speeds up all references to the bit-mapped memory by eliminating NuBus overhead. Note that the bit-mapped memory is the only resource on the SI board that is accessible on both the NuBus and the local bus; all other SI board transactions occur only over the NuBus.

The processor can perform three basic bit-mapped memory operations on either a word, halfword, or byte basis as follows:

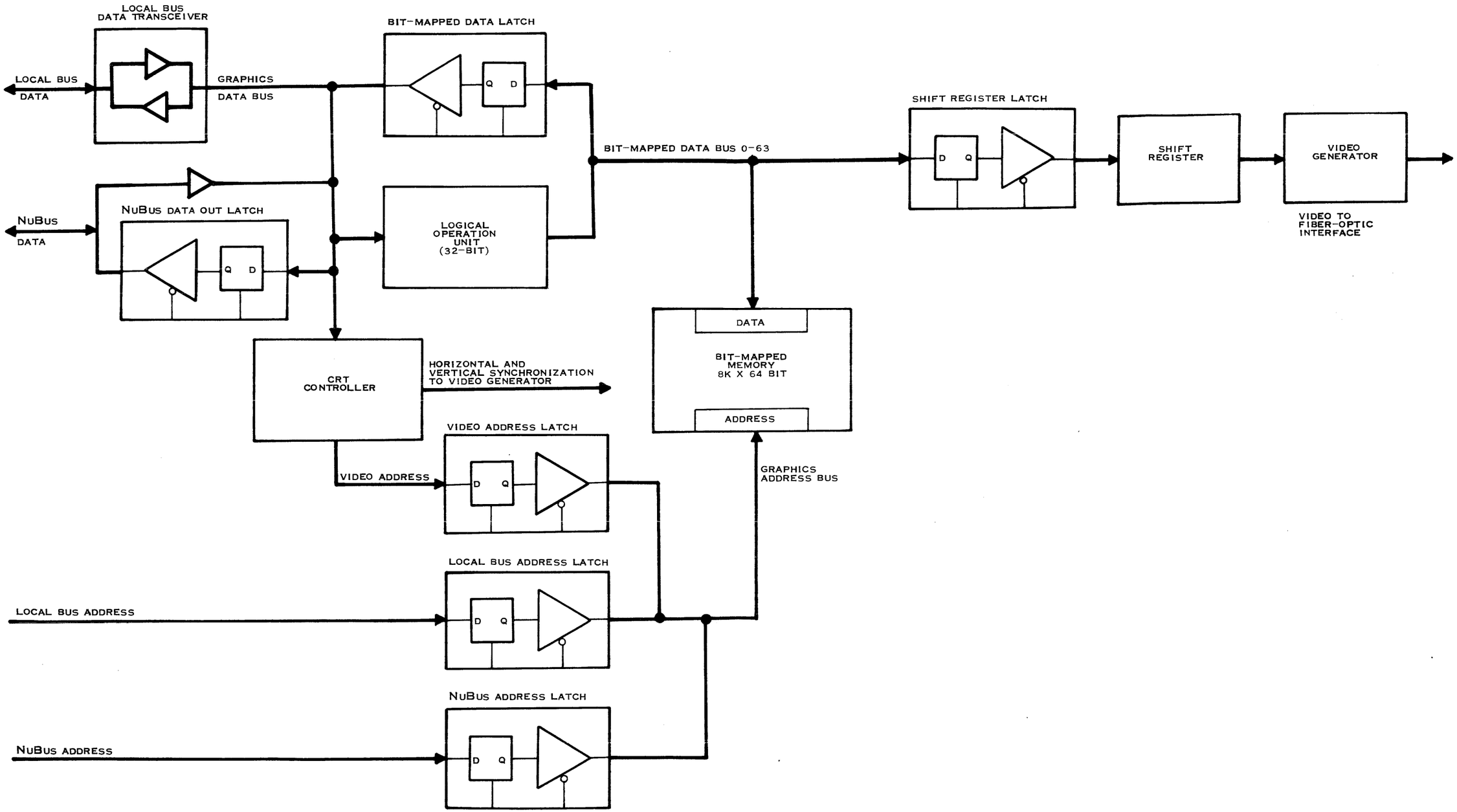
- Read — A processor can read 32-bit words, 16-bit halfwords, or 8-bit bytes.
- Write — A processor can write 32-bit words, 16-bit halfwords, or 8-bit bytes.
- Read-Modify-Write (RMW) — A processor can read-modify-write 32-bit words, 16-bit halfwords, or 8-bit bytes.

To perform an RMW operation, the processor executes a write operation to the bit-mapped memory. The SI board performs three operations during an RMW operation as follows:

1. The current value of the addressed memory data is read from the bit-mapped memory and temporarily stored in an on-board register.
2. A previously-specified logical operation, such as an AND or EXCLUSIVE-OR, is performed between the data read from memory and the data provided by the processor write operation.
3. The result of this logical operation is then written to the addressed location in the bit-mapped memory.

A 32-bit mask register is invoked during an RMW operation. This provides a bit-by-bit enable for the operation. If any of the 32 mask bits are set, the corresponding data bits in the addressed memory are not altered by the cycle. This mask function is especially useful when dealing with display windows.

Figure 4-10 Bit-Mapped Graphics Display Controller Block Diagram



Logical Operation Unit

4.4.10.2 The logical operation unit (LOU) performs a selected logical operation during RMW cycles using memory data and data written by the processor. Any one of 16 logical operations, including AND and EXCLUSIVE-OR operations can be performed. The masking operation is also performed within the LOU.

Shift Register Latch, Shift Registers, and Video Generator

4.4.10.3 The shift register latch latches 64 bits of bit-mapped memory data once every 942 nanoseconds. At the beginning of each video refresh memory cycle, the CRT controller generates a video refresh memory address. Once the addressed data is valid at the output of the memory, the shift register latch clocks in the 64 bits of data. This latch holds the data until the shift registers become empty and are ready to accept another 64 bits of data. When this happens, the shift register latch data is transferred to the shift registers. The shift registers are clocked by the 67.8989-megahertz pixel clock and convert the 64-bit parallel data into a serial data stream beginning with the least significant bit. One bit is shifted out of the shift registers each clock period (one bit every 14.72 nanoseconds). This serial bit stream is fed to the video generator, that in turn, feeds video to the fiber/optics interface for transmission to the monitor.

CRT Controller

4.4.10.4 The CRT controller generates horizontal and vertical synchronization pulses and blanking signals that ultimately control the video display and the fiber-optic link. Additionally, the controller generates memory addresses that fetch the proper memory data to be converted into the video that is sent to the monitor on the fiber-optic link. Also, the controller can generate an interrupt at the end of each full frame of video, when the vertical retrace begins.

NuBus Interface

4.4.10.5 The graphics display controller interface to the NuBus is through the IDATA< 31–00> data bus lines and the IADDR< 31–00> address bus lines. The SI board NuBus interface handles all the control, data, and address lines required to move data back and forth across the NuBus. The graphics interface consists of a set of data buffers and latches and the address latch that transfers data between these two subsystems.

Local Bus Interface

4.4.10.6 The local bus interface consists of an address selector, data selector, and memory arbitration and control logic. The local bus is a high-speed data bus with 32 data lines and 32 address lines. The local bus is dedicated to memory transfers between the system processor, the memory boards, and the SI board graphics logic. With separate address and data lines and without the need for arbitration between multiple masters to gain control of the bus, local bus overhead requirements are less than that for the NuBus.

Byte, halfword, and full 32-bit word transfers are possible on the local bus. The local bus has a bus transfer mode code that selects the type of data transfer that is virtually identical to the NuBus scheme for transfer mode control. One 96-pin male DIN connector is used to connect the SI board to the local bus on the backplane. Functions of the local bus signals are shown in Table 4-12.

Table 4-12 Local Bus Signal Functions

Signal Signature	*Function*
BCLK–	BCLK– is the local bus clock and is generated by the processor. All local bus control and data transfers are synchronized to BCLK–.
MEMREQ–	MEMREQ– is a memory request signal generated by the processor. MEMREQ– is asserted by the processor when a memory cycle is required.
MEMACK–	MEMACK– is a memory acknowledge signal asserted by the bus slave device to the processor to indicate that the current memory cycle will be completed on the next falling edge of the local bus clock.
BERR–	BERR– is a bus error signal asserted by a bus slave device to the processor to indicate that an error has been detected on a data transfer.
AD–< 31:00>	AD–< 31:00> are the 32-bit local bus address signals generated by the processor.
DAT–< 31:00>	DAT< 31:00> are the 32-bit data signals used for read and write data transfers on the local bus.
BS0–	BS0– is a local bus select signal generated by the processor. In addition to the slot occupied by the processor, BS0– is wired to only one other local bus board slot into which a memory board is installed (slot 4 of the 7-slot enclosure). The memory board occupying this specific hard-wired slot becomes the NuBus master device when BS0– is asserted by the processor.
FAST–	FAST–, when asserted by the processor, indicates to devices on the local bus that the processor on the local bus is an Explorer processor.
LOCK–	LOCK– is generated by the processor. When LOCK– is asserted, the resources of the memory board currently being accessed by the processor are locked and cannot be accessed by the NuBus. When locked, the memory board stays locked until LOCK– returns high inactive (is no longer asserted).
DECODE–	DECODE– is an open-collector output from memory boards. Any memory board accessed on the local bus pulls DECODE– low. If the bit-mapped memory is addressed, the board pulls DECODE– low. If DECODE– is high, no memory on any board is selected for a local bus transfer, and a NuBus master (on a memory board) can acquire the NuBus and complete a NuBus cycle.

Table 4-12 Local Bus Signal Functions (Continued)

Signal Signature	*Function*
LTM0–, LTM1–	LTM0– and LTM1– are the local bus transfer mode signals that are generated by the processor. LTM0– and LTM1– are used with AD00– and AD01– to form a 4-bit binary code that informs the slave device of the type of transfer mode the processor is requesting. Command encoding of transfer modes of reads and writes of words, halfwords, or bytes is as follows:

LTM0–	*LTM1–*	*AD1–*	*AD0–*	*Transfer Mode*
L	L	L	L	Write byte 3
L	L	L	H	Write byte 2
L	L	H	L	Write byte 1
L	L	H	H	Write byte 0
L	H	L	L	Write halfword 1
L	H	L	H	Reserved
L	H	H	L	Write halfword 0
L	H	H	H	Write word
H	L	L	L	Read byte 3
H	L	L	H	Read byte 2
H	L	H	L	Read byte 1
H	L	H	H	Read byte 0
H	H	L	L	Read halfword 1
H	H	L	H	Reserved
H	H	H	L	Read halfword 0
H	H	H	H	Read word

Bit-Mapped Graphics Display Controller Programming

4.4.10.7 The following paragraphs describe the programming of the graphics display controller. There are four basic areas discussed:

- Bit-mapped memory manipulations including read, write, and read-modify-write operations
- CRT controller initialization and interrupts
- Video test register reads
- Video attribute register programming

CRT Controller Initialization and Interrupts The CRT controller must be initialized before operation of the data link and monitor is possible since the fiber-optics data link requires horizontal and vertical synchronizing signals before it can function. To initialize the controller, first reset the controller by writing a byte of zeros to address hexadecimal FSE00058. Then write the data listed in Table 4-13 for each of the registers, from register R00 through register R15. The order must be followed and register R16 must not be rewritten, or the controller may be reset and require reinitialization. These registers are write-only registers. Data returned during read operations are not valid. Note also that other CRT controller functions suggested by the register names in Table 4-13 are not functional due to hardware constraints. Varying the data written into registers R00 through R15 can cause unpredictable results.

Also note that a board software reset can reset the CRT controller, which would require it to be reinitialized.

A board software reset condition requires that the system initialization sequence be followed precisely to ensure that the controller is not inadvertently reset. When the board is first powered up, it is necessary to reset the timers and real-time clock (to clear any spurious interrupts), then perform a board reset by writing to bit 0 of the configuration register. This operation clears the interrupt registers and readies the board for interrupt operation. After this has been completed, the CRT controller can be initialized. If the CRT controller is initialized before the interrupts have been cleared, clearing the interrupts will also clear the CRT initialization.

The CRT controller contains interrupt logic that enables an interrupt to be generated at the start of each vertical retrace. This occurs once every 16.67 milliseconds immediately after the CRT has completed a full video display refresh. This interrupt is enabled by setting bit 6 (hexadecimal 40) of a byte written to address hexadecimal FSE00068. A series of read and write operations to the CRT controller causes video glitches on the monitor. For example, polling for an interrupt during active video generates visible video glitches.

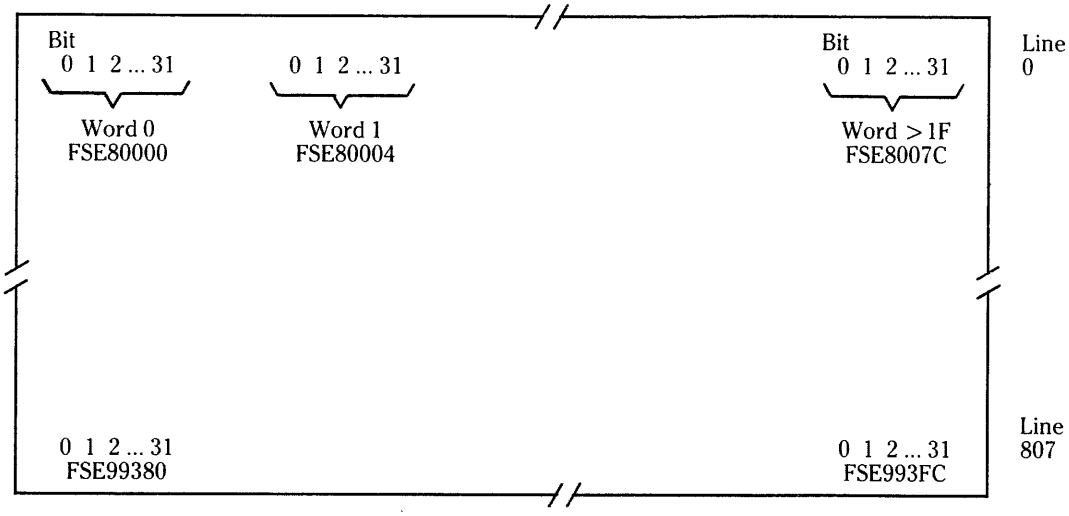
Reading address hexadecimal FSE00068 returns interrupt status and simultaneously clears the interrupt. Bit 7 is set if an interrupt is pending (hexadecimal C0); all bits are clear (hexadecimal 00) if no interrupt is pending. Once the interrupt occurs, an event is generated at the address loaded into the event generator register file. The interrupt must be cleared before another interrupt is generated.

Bit-Mapped Memory Operations The address sequence of the bit-mapped memory (as it is displayed on the video display) is from left to right and from top to bottom, in that order. The least significant bit of each word, halfword, or byte is the left-most displayed pixel of that word, halfword, or byte. Figure 4-11 diagrams the video addressing scheme. Read and write operations into the bit-mapped memory are handled by the processor in the same manner as any normal memory reference. RMW cycles are handled as normal processor write operations.

Table 4-13 CRT Controller Register Functions

Register Number	*Read/ Write*	*Control Space Address*	*Register Function*	*Data (Hex)*
Screen Format Registers:				
R00	Write	FSE00000	Characters per horizontal period	2A
R01	Write	FSE00004	Characters per data row	1F
R02	Write	FSE00008	Horizontal delay	07
R03	Write	FSE0000C	Horizontal sync. width	0A
R04	Write	FSE00010	Vertical sync. width	24
R05	Write	FSE00014	Vertical delay	25
R06	Write	FSE00018	Skew, pin configuration	80
R07	Write	FSE0001C	Visible data rows per frame	64
R08	Write	FSE00020	Scan lines (frame, data row)	67
R09	Write	FSE00024	Scan lines per frame LS byte	4A
Control and Memory Address Registers:				
R0A	Write	FSE00028	Control and memory	00
R0B	Write	FSE0002C	address registers	00
R0C	Write	FSE00030	R0A through R14 are	00
R0D	Write	FSE00034	auxiliary registers.	40
R0E	Write	FSE00038	These registers	00
R0F	Write	FSE0003C	require initialization.	00
R10	Write	FSE00040		80
R11	Write	FSE00044		80
R12	Write	FSE00048		80
R13	Write	FSE0004C		00
R14	Write	FSE00050		00
Cursor, Status, and Miscellaneous Registers:				
R15	Write	FSE00054	Start command	00
R16	Write	FSE00058	Reset command	XX
R17	Write	FSE0005C	Reserved (do not use)	XX
R18	Read/write	FSE00060	Reserved (do not use)	XX
R19	Read/write	FSE00064	Reserved (do not use)	XX
R1A	Write	FSE00068	Interrupt enable	XX
R3A	Read	FSE00068	Status	XX
R3B	Read	FSE0006C	Reserved (do not use)	XX
R3C	Read	FSE00070	Reserved (do not use)	XX

Figure 4-11 Bit-Mapped Memory Addressing Scheme



To perform an RMW cycle, the processor performs three separate write operations to the SI board: two that set up the operation and one that initiates the RMW cycle. The RMW image corresponds, address for address, to the basic bit-mapped memory. The difference is that address bit 18 is set to address the RMW image and to initiate the SIB read-modify-write operation. Thus, the RMW image occupies memory locations from FSEC0000 upward. These locations correspond, address for address, with basic memory map locations FSE80000 upward through FSE9FFFC.

The processor first writes the desired mask value into the mask register. The processor then writes a 4-bit code into the logical operation register to select the desired operation to be performed. Note that the values programmed into each of these registers are not altered during the RMW cycle; values are effective until changed by the processor. Once the logical operation code and mask value have been written, the processor executes a normal write operation with address bit 18 set high, which invokes an RMW cycle. If address bit 18 is low, a normal write, rather than an RMW cycle occurs. A read operation occurs normally, whether or not address bit 18 is set.

The mask register is a read/write register that determines which bits in an addressed memory word can be changed during an RMW operation. A one at a bit position in the mask register keeps the corresponding bit in memory from being modified while a zero in the mask register allows the corresponding bit to be modified. The mask register resides at address hexadecimal FSE00084.

The logical operation unit specifies which 1 of 16 logical operations is to be performed by an RMW cycle. Codes for each of the logic functions performed on the source data written by the processor (S) and the destination data read from memory (D) are listed in Table 4-14. The logical operation register is addressed at hexadecimal FSE00088.

Table 4-14 **Logical Functions**

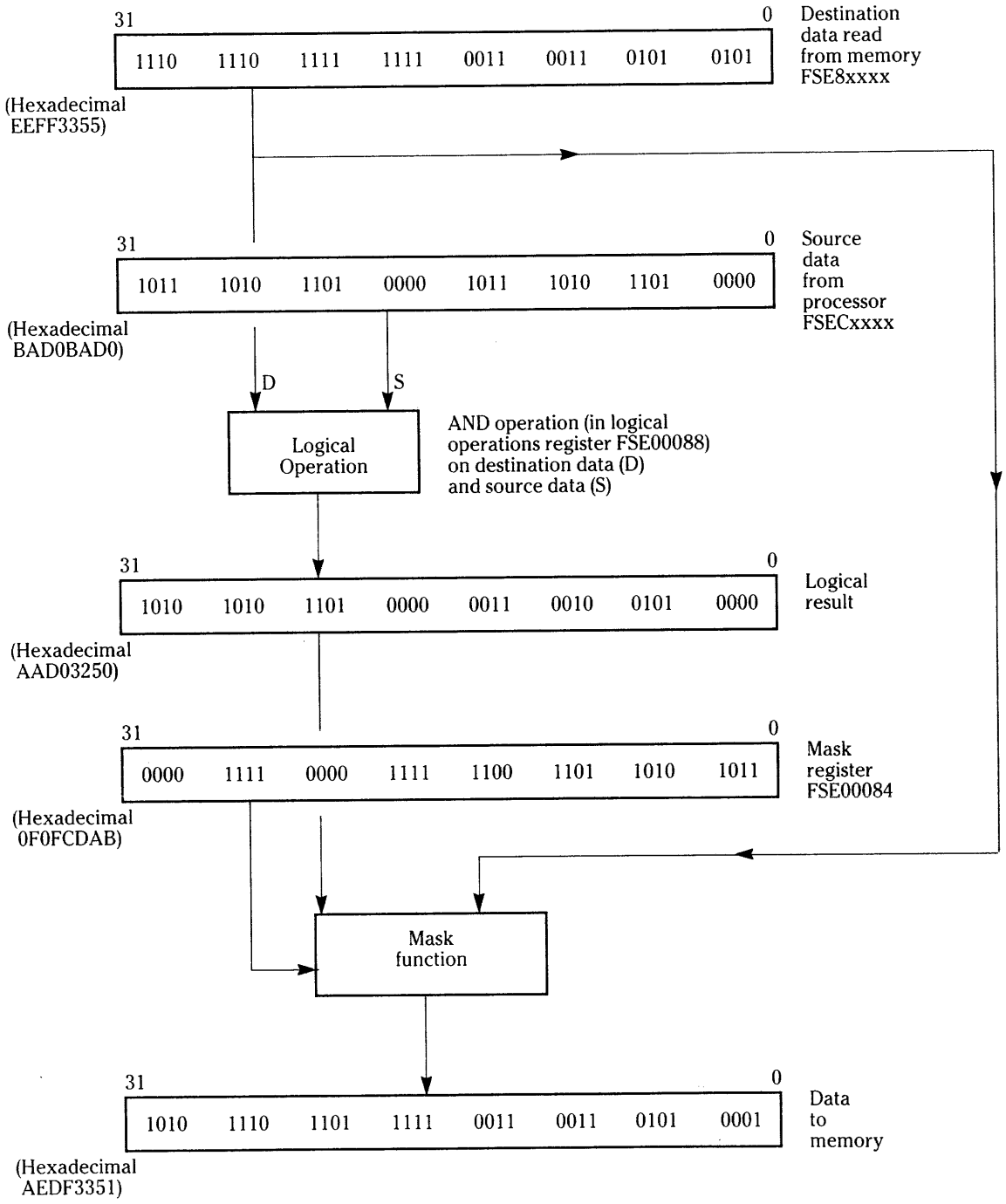
Data at Address Hexadecimal FSE00088				
Bit 3	***Bit 2***	***Bit 1***	***Bit 0***	***Function Performed***
0	0	0	0	CLEAR
0	0	0	1	D NOR S
0	0	1	0	S AND D–
0	0	1	1	D–
0	1	0	0	S– AND D
0	1	0	1	S–
0	1	1	0	D XOR S
0	1	1	1	D NAND S
1	0	0	0	D AND S
1	0	0	1	D XNOR S
1	0	1	0	NOP (S)
1	0	1	1	S OR D–
1	1	0	0	D
1	1	0	1	D OR S–
1	1	1	0	D OR S
1	1	1	1	SET

Example 4-1 shows how the RMW cycle operates. In this example, the hexadecimal value of EEFF3355 resides in the bit-mapped memory at the addressed location. The processor first writes hexadecimal 08 into the logical operation register at address hexadecimal FSE00088 to specify the logical operation AND. The processor then writes hexadecimal 0F0FCDAB, the desired mask register value, into address FSE00084. Note that the mask can be totally disabled by writing zeros to this address. The processor then writes the data to the SI board, which in this example is hexadecimal BAD0BAD0.

As shown in the example, the logical ANDing of hexadecimal EEFF3355 with hexadecimal BAD0BAD0 yields hexadecimal AAD03250. The mask register then selects the bits from this result that correspond to the zeros in the mask register. Bits from the original memory data (destination) that correspond to the ones in the mask register are selected to form the complete cord. This word is written into the bit-mapped memory. In this example, the result written to memory is hexadecimal AEDF3351.

In practice, the XOR function is used almost exclusively since that operation allows data written to the bit-mapped memory to be visible regardless of the contents of the bit-mapped memory. The mask register is most useful when windows are displayed on the video display; data outside a window boundary can be protected without previously reading and saving the data.

Example 4-1 Read-Modify-Write AND Operation



Video Attributes Register The video attributes register is a 4-bit read/write register that controls video polarity and video blanking (see Figure 4-12). Bit 0 (the least significant bit) controls blanking. When set (1), the video display is fully blanked (no video is visible). When reset (0), the video display displays the contents of the bit-mapped memory. Bit 1 controls video polarity. When it is reset (0), a 1 in the bit-mapped memory causes the corresponding pixel on the video display to light; a 0 in memory darkens the pixel (normal video). When bit 1 is set (1), a 0 in the bit-mapped memory causes the corresponding pixel on the video display to light; a 1 in memory darkens the pixel (reverse video). The video attributes register is located at address hexadecimal FSE00080.

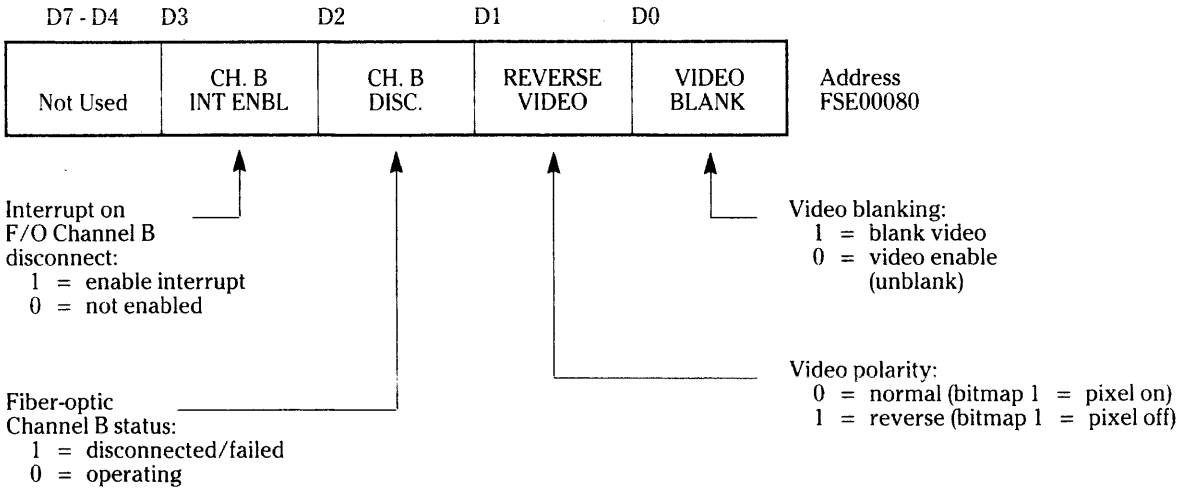
A series of video-blanking write operations during active video causes video glitches on the CRT monitor. For example, if the video is blanked several times during a vertical retrace period, video glitches will be visible.

NOTE: When the system is powered up or the board is reset, the attributes register will be set to reverse video and a blanked video display. The video display must be unblanked to view video.

Bit 2 of the video attributes register can be read to determine the status of the fiber-optic data link. If bit 2 is 0, channel B of the data link is operational; if bit 2 is 1, channel B of the data link is not operational.

Bit 3 of the video attributes register enables the data link interrupt. When enabled, an interrupt is generated if the operation of the channel B portion of the data link becomes inoperational. This could occur, for example, if monitor power is turned off. Writing a 1 to bit 3 enables the interrupt; writing a 0 to bit 3 disables the interrupt. This bit can be read.

Figure 4-12 Video Attributes Register



Video Test Register A 4-bit video test register records the video transitions in each scan line using 2 bits for the negative transitions and 2 bits for the positive transitions (see Figure 4-13). The video test register is at hexadecimal address FSE0098. This register updates after each horizontal scan and remains valid until the end of the next line scanned (about 20 microseconds). Bits 0 and 1 are used for the *negative* (high-to-low) transitions. Bit 0 of the register is set if there were an odd number of negative (high-to-low) transitions in the previous horizontal scan line. Similarly, bit 1 is set if there were an even number of negative transitions in the line. Bits 2 and 3 are used for the *positive* (low-to-high) transitions. Bit 2 is set if there were an odd number of positive transitions in the line; bit 3 is set if there were an even number of positive transitions.

The video test register always reflects an extra pair of transitions in the register contents. This is because the TSTOUT bit falls at the end of the Manchester encoded data on channel A.

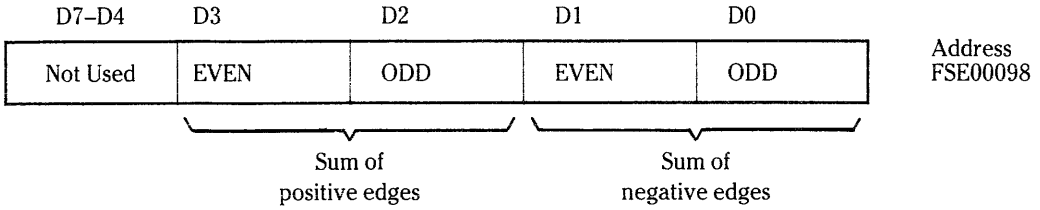
NOTE: Before a read operation is performed on the video test register, the TSTOUT bit must be set to a logic 0 at hexadecimal address FSF20010. The TSTOUT bit assignment is described in paragraph 4.4.11.9, Diagnostic Register.

For test purposes, a pattern with a known number of video transitions per line can be entered into the bit map. Reading the video test register verifies proper functioning of the shift register and video logic.

For example, if the bit map contains all zeros and the video polarity is normal (where normal is a zero), the contents of the video test register will be 5 (0101), indicating an odd number of positive and negative transitions, that is, an extra pair of transitions. If you reverse the video and wait for the screen to reverse polarity, the register will contain a hexadecimal A (1010), which indicates an even number of positive and negative transitions. A working SI board generates the values 0 (0000), 5 (0101), A (1010), and F (1111) in the video test register.

NOTE: The 20-microsecond timing between updates makes it difficult to assure that the data read is that of the intended line. For best results, do the following: (1) Enter a test pattern in the entire bit map; (2) start a wait loop, beginning on a vertical cycle and ending 8.8 milliseconds later; and (3) read the video test register. This procedure allows the bit map to become uniform and the read operation to occur during active video.

Figure 4-13 Video Test Register



Mouse/Keyboard/ Audio Interface

4.4.11 The mouse/keyboard/audio interface provides the data paths and processing logic for all data received from the monitor and all non-graphic data transmitted to the monitor.

Data coming from the monitor includes mouse motion data, mouse keyswitch closures from three switches, keyboard data in 2400-baud serial format, and digitized voice input. Data to the monitor includes keyboard functions in 2400-baud serial format, sound generator control, and speech synthesis data.

Mouse/Keyboard/ Audio Interface Address Decoders

4.4.11.1 The mouse/keyboard/audio interface has two similar address control decoders. Each decoder decodes a base address and a small range of addresses above the base address. Mouse position, mouse keyswitch, sound control, voice, speech synthesis, and diagnostic functions share the same address decoder and the same base address, which is hexadecimal FSF20000. The other address control decoder, used for keyboard functions, decodes a base address of hexadecimal FSFC0000.

Table 4-15 shows the address map for the mouse, audio, and diagnostic registers. All addresses fall on word boundaries, although a 16-bit halfword is the largest unit of data in any of the registers. Keyboard addressing is described in a later paragraph.

Mouse Motion and Position Logic

4.4.11.2 Mouse motion and position logic resides on the SI board to process mouse motion data received from the video display via the fiber-optic data link. The logic consists of a motion/keyswitch data register, a motion detector, and mouse position counters and registers.

Motion/Keyswitch Data Register Inputs to the motion/keyswitch data register include the status of the mouse keyswitches, the raw quadrature mouse data, and the keyboard serial data line. The motion/keyswitch data register functions to hold mouse keyswitch data and other data for transfer over the I bus to the host processor and to provide stable inputs to the keyswitch detector and motion detector circuits.

The host processor is normally interested only in the keyswitch status so the mouse raw quadrature data is masked off. The capability exists, however, for processor software to bypass the mouse position counters and registers to collect raw mouse data.

Table 4-15 **Address Map for Mouse, Audio, and Diagnostic Registers**

Address	*Register Selection*	*IDATA Bits*	*Access*
FSF20000	Mouse Y-position register	15-00	Read/write
FSF20004	Mouse X-position register	15-00	Read/write
FSF20008	Motion/keyswitch register	07-00	Read
FSF2000C	Interrupt enable register Diagnostic control register	07-04 03-00	Read/write
FSF2000D	Monitor control register	11-08	Read/write
FSF20010	Diagnostic data register	07-00*	Read/write
FSF20011	Diagnostic data register (continued)	08*	Read/write
FSF20014	Sound control register	07-00	Read/write
FSF20015	Sound control register (continued)	08**	Read
FSF20018	Speech register	07-00	Read/write
FSF20019	Speech register (continued)	08**	Read
FSF2001C	Voice register	07-00*	Read
FSF2001D	Voice register (continued)	08*	Read

NOTES:

* The diagnostic data register and voice register each require bits 00 through 08, with bits 09 through 15 unused.

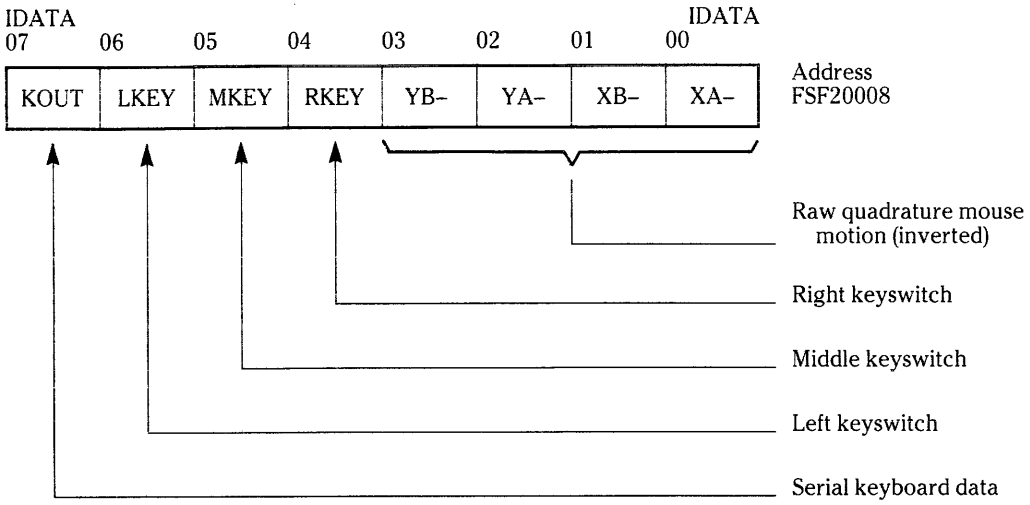
** SI board hardware generates odd parity in bit 08 of the speech and sound control registers. Bit 08 is readable but not directly writable via the NuBus/I bus.

The serial keyboard data can also be sampled by the host processor for possible testing, after being processed through a deglitching circuit that ensures data validity. Like the raw motion data, software normally masks off the keyboard serial data line input to the processor.

The motion/keyswitch data register is loaded on every clock pulse unless an I-bus operation (with mouse block selected) is in progress. This nonloading on an I-bus operation prevents the outputs from changing during a read operation.

Figure 4-14 shows bit assignments for the motion/keyswitch data register.

Figure 4-14 Motion/Keyswitch Data Register Bit Assignments



Motion Detector As the mouse is moved, it sends directional movement data, encoded in quadrature format, to the mouse interface. The motion detector decodes the direction of change and either increments or decrements a corresponding X and/or Y mouse position counter. Mouse motion in the X (horizontal) direction is positive to the right; motion in the Y (vertical) direction is positive downward. The convention of positive X to the right and positive Y downward corresponds to a mouse cursor with the origin at the top left corner of the video display.

The motion detector is a programmable logic device that consists of four states. Depending on the present state of the XA– and XB– signal lines and their last state, edge detection is accomplished and the decision to increment or decrement the X mouse position counter is determined. A similar finite-state machine decodes the YA– and YB– signal lines to determine movement in the Y direction.

The XA–, XB–, YA–, and YB– quadrature mouse motion signal lines are sampled once every 100 nanoseconds. This sampling rate, which is approximately 1000 times the maximum expected data rate, ensures fast and accurate response to motion changes.

Mouse Position Counters and Registers The mouse position counters consist of a 16-bit X-position up/down counter and a 16-bit Y-position up/down counter. Each counter holds one component of the absolute mouse position (16-bit X and 16-bit Y coordinates). The host processor can preload these counters, making the mouse position relative to some fixed point.

The incrementing and decrementing of the position counters is controlled by the motion detector. As the mouse is moved, movement information in quadrature format is sent to the motion detector. The motion detector detects the direction of movement and increments or decrements the corresponding X-position or Y-position counter.

To prevent interference between updating and reading the mouse position, an X-position register and a Y-position register buffer the appropriate counter data onto the I bus. Mouse position counter contents are copied into the mouse position register on every clock pulse. An update inhibit circuit prevents the transfer if the mouse logic is selected for a data bus cycle. This prevents the register contents from changing in the middle of an I bus data transfer. However, accurate motion changes are maintained during the read operation. The position registers have three-state driver outputs to the I bus.

Figure 4-15 shows bit assignments for the mouse position registers.

Keyswitch Detector

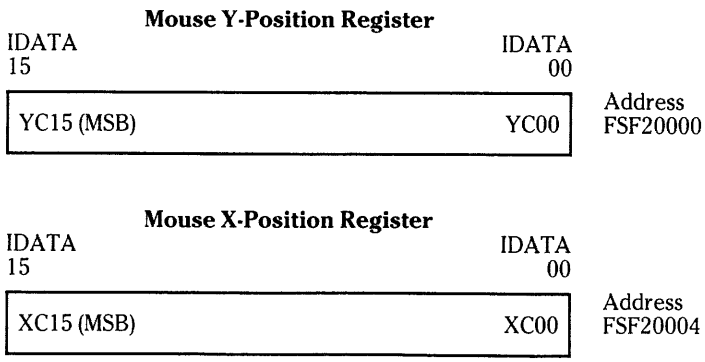
4.4.11.3 The keyswitch detector detects changes in the status of the three keyswitches located on the mouse. The keyswitch detector monitors the keyswitch outputs of the motion/keyswitch register. When the operator presses or releases any of the keyswitches, the keyswitch detector recognizes the change in status and notifies the interrupt handshake controller. If enabled by the keyswitch interrupt enable signal, the interrupt handshake controller sends the keyswitch interrupt to the event generator, and a keyswitch event goes out on the NuBus. The event notifies the host processor to read the motion/keyswitch register.

The processor software does not have to use this event-based scheme to check for keyswitch state changes. It has the option of disabling the keyswitch event by setting the keyswitch interrupt enable signal low and then polling the motion/keyswitch detector. A polling rate of once per video horizontal scan line is the maximum practical rate. Using this option, the software must include the functional equivalent of the keyswitch detector logic.

Interrupt Enable and Monitor Control Registers

4.4.11.4 The interrupt enable register allows the processor to selectively mask sound error interrupts, mouse motion interrupts, keyswitch interrupts, and voice data present interrupts. Although the interrupt enable bits share a byte of address space with the diagnostic control bits, they are not limited to diagnostic functions. The interrupts are disabled on power-up or SI board reset, so the programmer must set the appropriate interrupt enables as part of the board initialization procedures.

Figure 4-15 **Mouse Position Register Bit Assignments**



The monitor control register includes a monitor sound enable bit, a microphone gain-control bit, and a pair of monitor test functions. Monitor sound enable (MOSOENB) is disabled on power-up or SI board reset. The monitor sound enable bit must be set by a subsequent control word to enable the monitor speaker amplifier. This disabling and enabling procedure prevents possible annoying sound bursts between the time a board reset is released and the time the sound generator and speech synthesizer setups are completed.

The microphone gain control bit (HIGAIN) allows a threefold increase in microphone gain under software control. There is no manual gain control for the microphone inputs to the monitor.

The analog loopback bit (ALOPBAK) sets up a special audio path in the monitor for operator verification of coder/decoder (codec) circuits. The special audio path uses the codec to encode the microphone output and to immediately decode the voice data to drive the speaker amplifiers. The effect is to create an audio system in which the codec is a central component. Absence of distortion in the audio output verifies codec operation. No data transfers on the fiber-optic link are involved except that MOSOENB and ALOPBAK must both be set for this test.

The parity check bit (PARCHK) forces bad (even) parity on the sound control output to the monitor. This is a diagnostic test that verifies the ability of the monitor circuits to recognize bad parity. Sound parity is described in the description of the sound control register in a later paragraph.

Figure 4-16 shows the bit assignments for the interrupt enable register and the monitor control register. The interrupt enable and monitor control registers are addressed at addresses hexadecimal FSF2000C and FSF2000D, respectively.

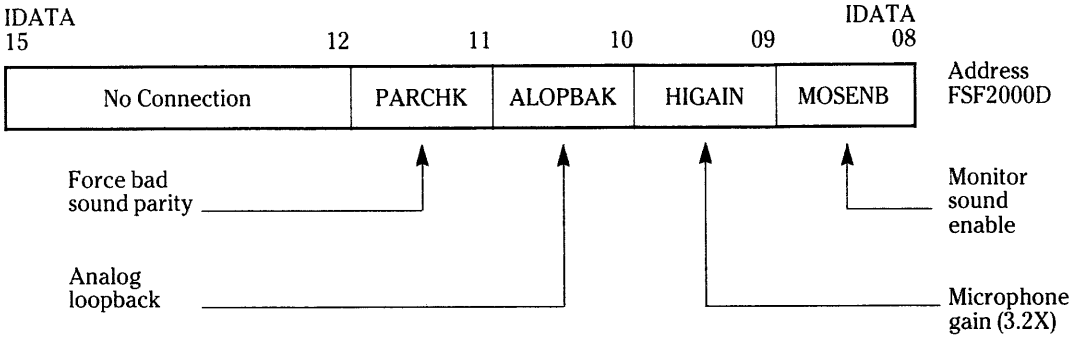
Interrupt Handshake Controller

4.4.11.5 The interrupt handshake controller handles the mouse, keyboard, and sound end of the interrupt and interrupt acknowledge exchanges with the event generator and the host processor. Inputs to the interrupt handshake controller include:

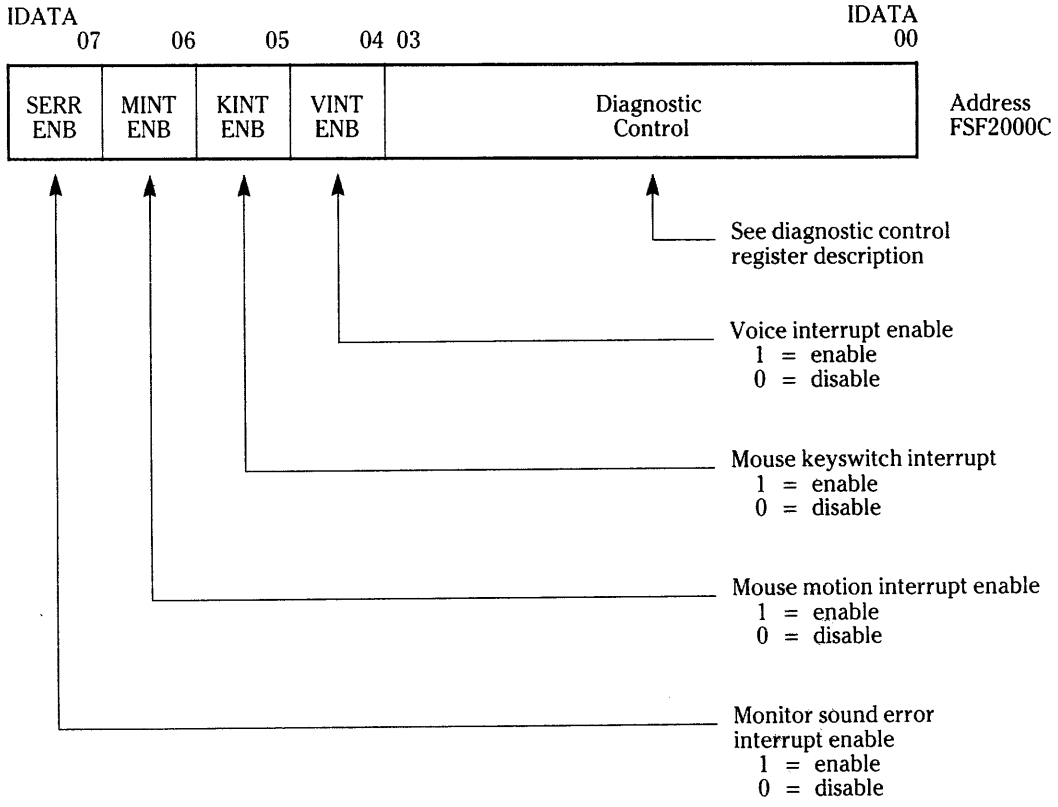
- Interrupt enable signals from the interrupt enable register
- Interrupt acknowledge (handshaking) signals from the event generator
- Interrupt initiating signals from the fiber-optic data link, mouse motion detector, and mouse keyswitch detector circuits
- SI board clock

Figure 4-16 Interrupt Enable and Monitor Control Registers

Monitor Control Register



Interrupt Enable and Diagnostic Control Register



Logic on the SI board calculates an odd parity bit each time the I bus loads a new 8-bit byte into the sound control register. The sound control parity bit (SC08) and data byte are transmitted to the monitor. Parity is recalculated at the monitor and compared with the transmitted sound control parity bit. If they differ, the monitor returns a sound error bit (SONDERR) to the SI board. If the sound error interrupt is enabled, a sound error interrupt triggers an event at the event generator. The event generator clears the sound error interrupt with the sound interrupt acknowlege signal.

Separate mouse and keyswitch interrupt outputs trigger events at the event generator. Each interrupt is cleared by its respective acknowledge input.

The keyboard interrupt shares the interrupt handshake controller with the mouse and sound error interrupts. The keyboard interrupt does not require an explicit enable signal. Either a keyboard ready to receive or a keyboard ready to transmit signal initiates a keyboard interrupt, assuming the keyboard is programmed to generate these outputs. A separate interrupt synchronizer in the event logic forces an interrupt event. The keyboard acknowledge signal clears the interrupt synchronizer. The ready to receive or ready to transmit signals remain active until the processor reads or writes keyboard data.

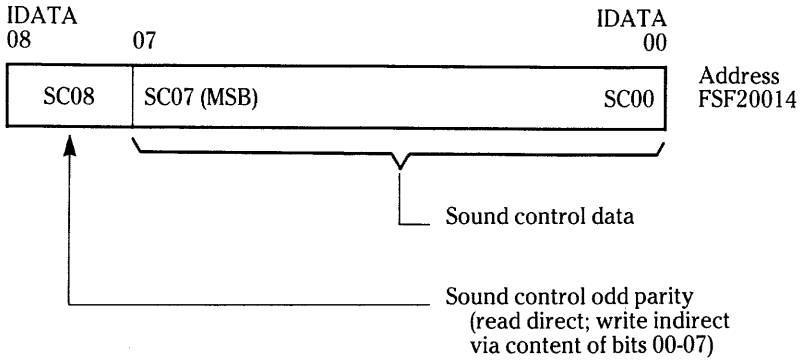
Sound Control Register

4.4.11.6 Audio circuits in the system monitor include a sound generator, an audio amplifier, and a speaker. The monitor audio is controlled by control bytes loaded into a sound control register on the SI board and transmitted over the fiber-optic link to the monitor. Since an undetected single-bit error in a sound control byte could cause blaring or other undesirable sounds, a parity bit is generated and is transmitted along with the sound control byte.

The sound control register consists of a 9-bit wide register and a parity generator. An 8-bit value from the I bus (IDATA< 07:00>) is applied to the input of the register and the parity generator monitors the same eight bits to generate an odd parity bit. At the output of the register, sound control bits SC< 07:00> are combined with odd parity bit SC08.

The sound control bits and the parity bit feed a parallel-to-serial converter in the fiber-optic interface logic. A set of three-state drivers provides diagnostic read-back capability. There is no path to read data back from the sound generator at the monitor. However, if sound control parity calculated at the monitor disagrees with the transmitted parity bit, a sound error signal is returned to the SI board. A parity check bit input to the sound control register allows a diagnostic test to force the transmission of bad parity. A properly operating monitor will detect the bad parity and return a sound error signal. Sound control register bit assignments are shown in Figure 4-17. The sound control register is addressed at address hexadecimal FSF20014.

Figure 4-17 **Sound Control Register Bit Assignments**



The frequencies, relative amplitudes, and noise characteristics are programmable at intervals comparable to the video display scan rate. The fiber-optic interface transmits all nongraphic data to the monitor during horizontal retrace periods.

Refer to the *Explorer Display Unit General Description* manual for a detailed description of the sound generator programming formats, sequences, and examples.

Speech Register **4.4.11.7** A speech register stores speech synthesis data for transmission to the monitor via the fiber-optic link. The monitor includes speech synthesis capability. The speech synthesizer circuits are driven by bytes of data loaded into the speech register at the SI board and transmitted to the monitor over the fiber-optic link.

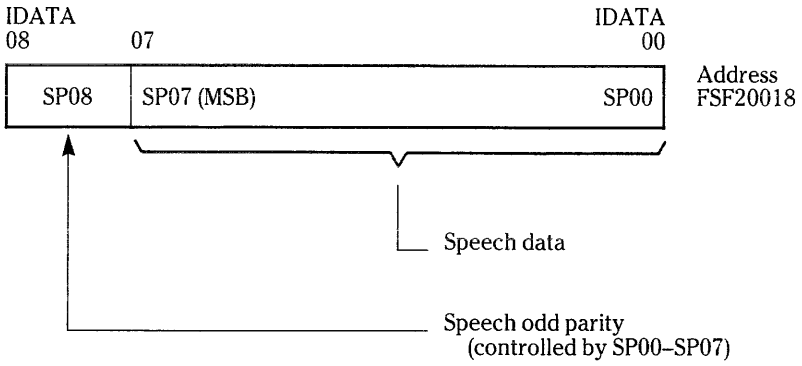
The speech register consists of a 9-bit wide register (of which 8 bits form a byte of speech) and a parity generator. An 8-bit value from the I bus (IDATA< 07:00>) that consists of a byte of speech synthesis data is applied to the input of the register; the parity generator monitors the same eight bits to generate an odd parity bit. At the output of the register, speech data bits SP< 07:00> are combined with odd parity bit SP08.

The speech data bits and the parity bit feed a parallel-to-serial converter in the fiber-optic interface logic. A set of three-state drivers provides diagnostic read-back capability. There is no path to read data back from the speech synthesizer in the monitor. Speech register bit assignments are shown in Figure 4-18. The speech register is addressed at address hexadecimal FSF20018.

Speech components must be programmed at a fixed, 8-kilohertz rate that corresponds to and is synchronized with the 8-kilohertz operating frequency of the codec on the monitor interface board. Speech data, like all other nongraphic data, is transmitted over the fiber-optic interface during horizontal retrace periods.

Refer to the *Explorer Display Unit General Description* manual for additional information on speech output.

Figure 4-18 **Speech Register Bit Assignments**



Voice Register and Voice Interrupt Generator

4.4.11.8 A 9-bit voice register provides an input path for digitized voice data from the monitor to the SI board. The most significant bit of the received digitized voice data, (VO08), is a data-present bit that is set to identify a valid byte of voice data and is used as a synchronizing bit. The data-present bit is applied to the input of the voice interrupt controller. If the data-present bit is set, there is no voice register read operation in progress, and the voice interrupt is enabled; then the voice interrupt controller triggers an event and loads the voice register with voice data bits VO<07:00>. The voice interrupt is cleared when the contents of the voice register is read onto the I bus. There is no explicit interrupt acknowledge signal.

The fiber-optic data link can read samples from the monitor at the 50.52-kilohertz horizontal scan rate. However, the codec on the monitor interface board provides new voice data samples at a fixed 8-kilohertz rate asynchronous to the scan timing. The voice-data-present bit serves as a flag to identify each new voice data sample as it arrives at the SI board.

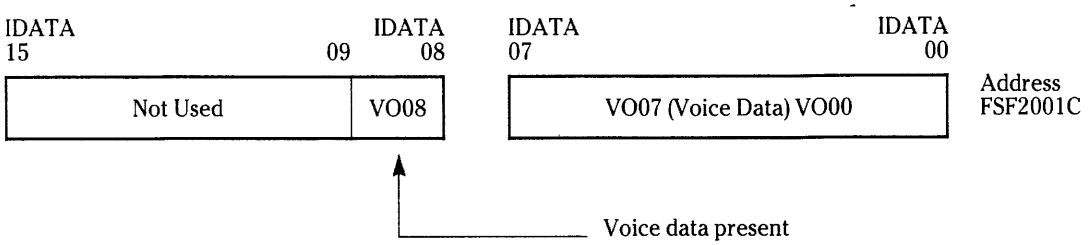
Figure 4-19 shows the voice register bit assignments. The voice register is addressed at address hexadecimal FSF2001C.

Diagnostic Register

4.4.11.9 The diagnostic register enables the host processor to loop data back through serial data paths and to supply simulated data to parallel data paths. By forcing known data patterns into circuit inputs and then reading the outputs, the processor can detect and isolate hardware failures.

The diagnostic register is divided into a diagnostic control field and a diagnostic data field. The diagnostic control field shares a byte of address space with four interrupt enable bits. Bit assignments for the diagnostic control field and the diagnostic data field are shown in Figure 4-20 and Figure 4-21. The diagnostic control field and the diagnostic data field are addressed at addresses hexadecimal FSF2000C and FSF20010, respectively.

Figure 4-19 Voice Register Bit Assignments



The serial data external loopback select bit (ELOPBACK) is applied to the fiber-optic interface where it is sent to the monitor via the high-frequency (channel A) fiber-optic channel. A high ELOPBAK signal decoded at the monitor selects the external loopback mode. When in the external loopback mode, the fiber-optic transmit output to the keyboard is connected directly to the fiber-optic receive input from the keyboard. In this manner, data transmitted from the SI board on channel A is returned directly to the SI board on the low-frequency (channel B) fiber-optic channel. If a read operation verifies that the returned data is unchanged, the external loopback has verified the proper operation of both fiber-optic data channels and the interface logic at both ends of the data path. If the keyboard is not responding to commands, but the external loopback performs properly, the problem is in the keyboard.

Figure 4-20 Diagnostic Control Field Bit Assignments

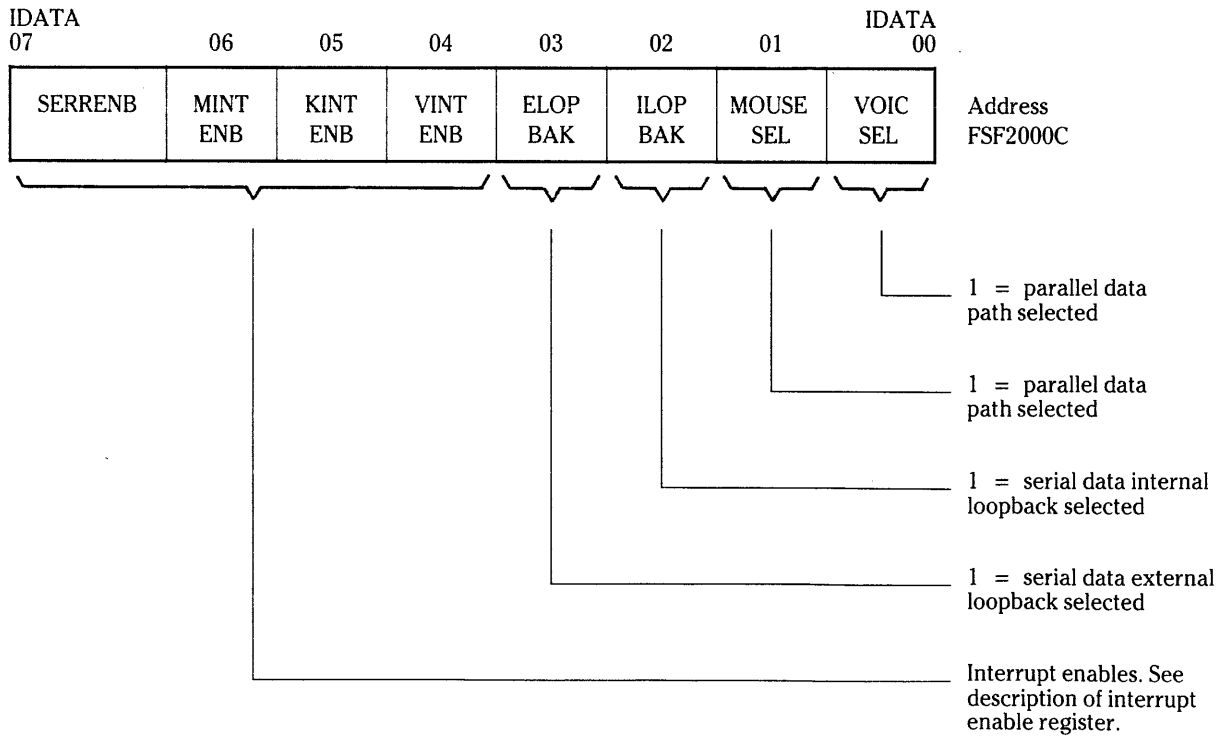
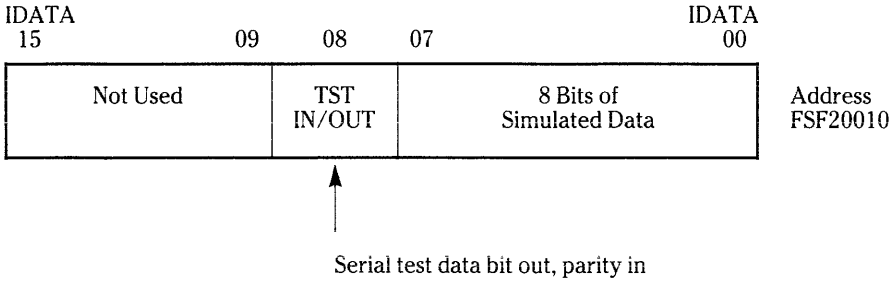


Figure 4-21 **Diagnostic Data Field Bit Assignments**



The serial data internal loopback select bit enables an internal check of the SI board fiber-optic interface. A data simulator picks off the value of the serial test data bit, TSTOUT, from the channel A transmitted data and replaces the normal channel B received data with a simulated channel B data stream. The simulated channel B data stream has 19 data bits set equal to TSTOUT. Sampling any of the bits in the motion/keyswitch data register (at hexadecimal location FSF20008) should produce the complement of the TSTOUT bit. This test verifies proper multiplexing and Manchester encoding of channel A and the ability to decode received data on channel B, but does not verify whether bit slippage in channel B demultiplexing has occurred.

If the test in external loopback mode fails and the test in internal loopback mode also fails, the problem is isolated to the SI board. If the internal loopback test passes after the external loopback test fails, the problem is in the fiber-optic link or in the monitor.

The mouse select bit (MOUSSEL) gates the simulated parallel data byte to replace the mouse motion, keyswitch, and keyboard data normally received from the fiber-optic interface. The host processor can then read the mouse motion/keyswitch registers to verify the path. The host processor can also check the performance of the motion and keyswitch detectors by writing motion bit sequences to the diagnostic register and then reading the mouse position registers for proper increment/decrement results.

The voice select bit (VOICSEL) gates the simulated parallel data byte to the voice register. The TSTOUT bit serves as the voice-data-present bit. TSTOUT must be set to trigger a voice interrupt and to load the voice register. After the video has been established, setting VOICSEL to a logic 1 causes the video to blank.

Keyboard Asynchronous Communication Interface

4.4.12 The keyboard asynchronous communication interface uses a universal synchronous/asynchronous receiver/transmitter (USART) device to accept data characters from the SI board in parallel format and convert them into a continuous serial data stream for transmission via the fiber-optic data link to the monitor. Simultaneously, the USART can receive serial data streams from the keyboard at the monitor via the fiber-optic data link and convert them into parallel-formatted data characters for application to the SI board. The fiber-optic data link consists of the fiber-optic board and the fiber-optic cable. Assuming proper operation, the fiber-optic data link can be considered to be a straight piece of wire between the keyboard and the USART on the SI board.

The USART serial data communication format consists of 8-bit data characters, even parity, and a single stop bit. The serial data is transmitted in the asynchronous mode at a 2400-baud signaling rate. The least significant bit of the data character is transmitted first. Start, stop, and parity bits are handled entirely within the USART. Parity error, receiver overrun error, framing error, and break detect are available as USART status bits and do not generate interrupts.

A transmit interrupt is generated each time the transmitter buffer of the USART is empty and ready to accept a new character. A receive interrupt is generated each time a new character is fully shifted into the receive buffer of the USART. The interrupts are ORed outside the USART to produce an overall keyboard event.

USART Operation

4.4.12.1 The USART connects to the eight least significant bits of the I bus of the SI board via the M data bus (MDATA< 07:00>). The USART is completely programmable and requires software setup before receiving or transmitting any data. The software setup consists of control bytes that are loaded in sequence. The control bytes can be formatted into either a mode instruction or a command instruction. After reset, the first control byte is treated as a mode instruction. This byte selects fundamental device parameters such as clock division ratio, character length, parity, and the number of stop bits transmitted.

The second and subsequent control bytes are command instructions that deal with transmit and receive operations. A command instruction includes a transmit enable bit, a receive enable bit, a request to send, a status error reset, and a USART internal reset. All control bytes received after the first one are treated as command instructions until either an internal reset or an SI board reset occurs. After either reset, the first control byte to the USART device is treated as a mode instruction. There is no readback capability for mode or command instructions.

A status-byte read operation to the USART can be made at any time. Status update is inhibited during the status read but this does not interfere with ongoing operations. A status byte includes parity error, framing error, and overrun error conditions, as well as other status information.

Asynchronous Receive Operations Assume that the mode of operation for the USART has been established and that a command instruction byte has turned on the receive enable bit. In the absence of data, the receive data line is normally high (marking). A falling edge on the receive data line triggers the start bit detection sequence. An internal strobe at the nominal bit center samples the line to detect a genuine start bit. If the start bit is good, a bit counter is started that locates the nominal center of each data bit, the parity bit, and the first stop bit. Data and parity bits are sampled on the rising edge of the receive clock.

A parity error sets the parity flag in the status register without disrupting the receive cycle. The stop bit marks the end of the data character. If the stop bit is low when sampled, the character is not correctly framed between the start and stop bits. This framing error sets a framing error flag in the status register.

After the stop bit is received, the received data character is loaded into the receive data buffer, and a receive ready output signal from the USART causes a NuBus event. The new character overwrites the previously received data character. If this occurs before the old character is read, a data overrun error has occurred. A data overrun error sets a bit in the status register. It is the responsibility of the programmer to check the status on each received byte and to read each data byte before an overrun occurs. All error flags can be reset by a bit in the command instruction byte.

Asynchronous Transmit Operations The USART attaches a start bit, parity bit, and stop bit to each transmit data byte received from the I bus of the SI board. This frame of data is shifted out serially by the USART on its transmit data line. Data shifts on the falling edge of the USART transmitter clock. In the absence of data, the transmit data line remains high (marking) unless it is programmed to send continuous break (low) characters.

USART Programming

4.4.12.2 For programming purposes, the USART appears as two addressable memory locations as shown in Table 4-16. The address for the USART is hexadecimal FSFC0000. There are four 8-bit registers. Two of the registers, the status register and read data register are read-only registers. The other two, the control register and transmit data register, are write-only registers. The following programming descriptions are based on asynchronous operation. Features of synchronous operation are either abbreviated or omitted entirely.

USART Mode Selection The USART requires an initial mode instruction to define its operating parameters and subsequent command instructions to control ongoing data transfers. Programming of the USART requires care because the control bytes containing mode and command instructions do not have separate addresses or bit assignments. Sequential logic in the USART steers a control byte into either a mode register or a command register.

Table 4-16 Addresses and Functions of USART Registers

Address (Hexadecimal)	*Read/Write*	*Register Selection and Function*
FSFC0000	Read	Status register. Contains the status byte.
FSFC0000	Write	Control register. Contains the control byte that is either a mode or command instruction, depending on the reset history and the write sequence.
FSFC0004	Read	Read data register. Contains the receive data.
FSFC0004	Write	Transmit data register. Contains the transmit data.

On power-up, the SI board reset forces the USART into a mode set operation. The first control byte written to the USART is loaded into the mode register; subsequent control bytes are loaded into the command register. Figure 4-22 shows bit assignments for the mode instruction. In the SI board application, the mode instruction is set to provide for one stop bit, even parity, parity enabled, 8-bit characters, and the transmit/receive clock divided by 64.

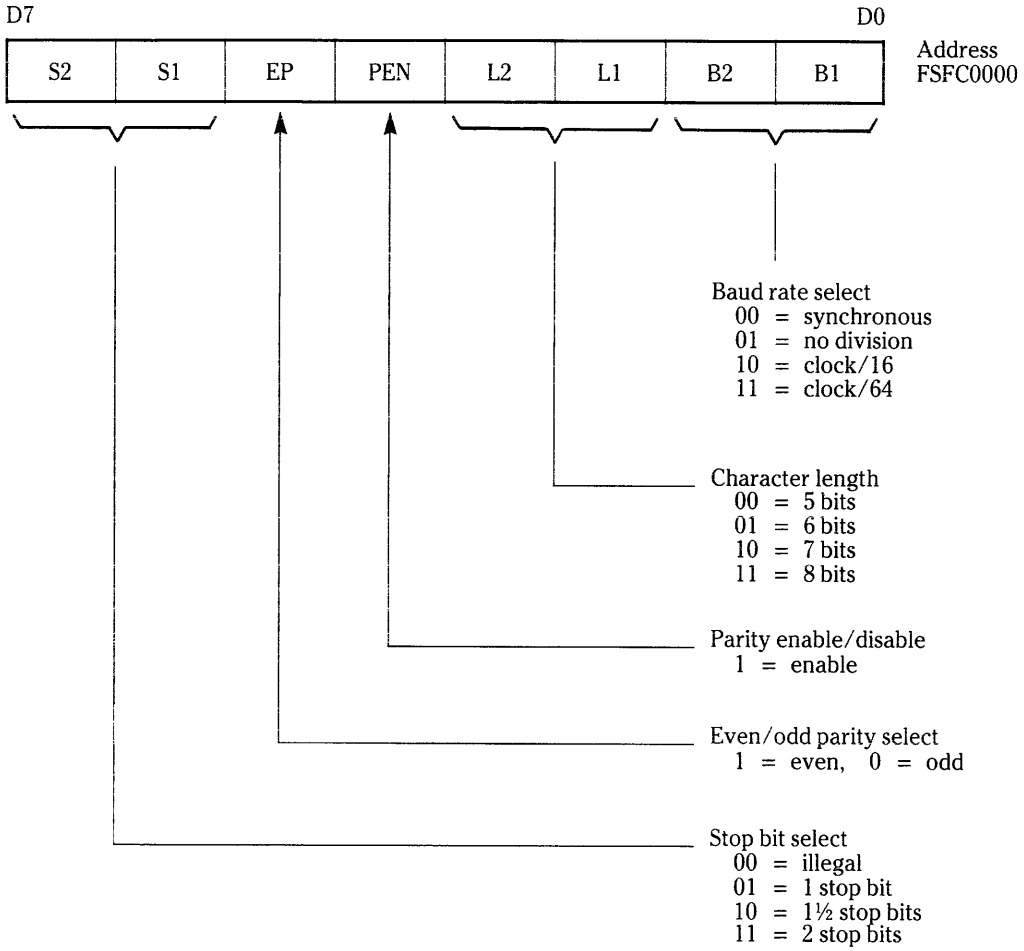
NOTE: The USART has a longer cycle time than its NuBus access time, which requires the processor to wait at least 3.2 microseconds between successive write operations to the USART.

After the reception of the initial mode instruction control byte, subsequent control bytes to the same address go into the command register. The USART mode of operation can only be changed after a general SI board reset or after a USART internal reset forced by a bit set in the command instruction. The initial USART mode and operating parameters must be set after each power-up or reset. The keyboard is not usable until the USART is up and running.

The mode instruction control byte cannot be read. If errors are suspected in the mode instruction control byte, write a command instruction control byte to generate an internal reset and then follow this with a new mode instruction control byte.

USART Operating Parameters Selection The command instruction control byte controls the character-by-character operation of the USART. Figure 4-23 shows the bit assignments in the command instruction control byte.

Figure 4-22 USART Mode Instruction Control Byte Format



USART Status Reporting Transmit ready and receive ready output signals from the USART are ORed to generate a summary keyboard event. Upon recognition of the event, the host processor should check the operation of the USART by reading the status byte in the status register. Several of the bits in the status byte duplicate USART output signals. Also, the status byte includes error flags for receive operations. These error flags should be checked on each received character since the USART gives no other indication that the received data character may be bad. Figure 4-24 shows bit assignments for the USART status byte.

Figure 4-23 USART Command Instruction Control Byte Format

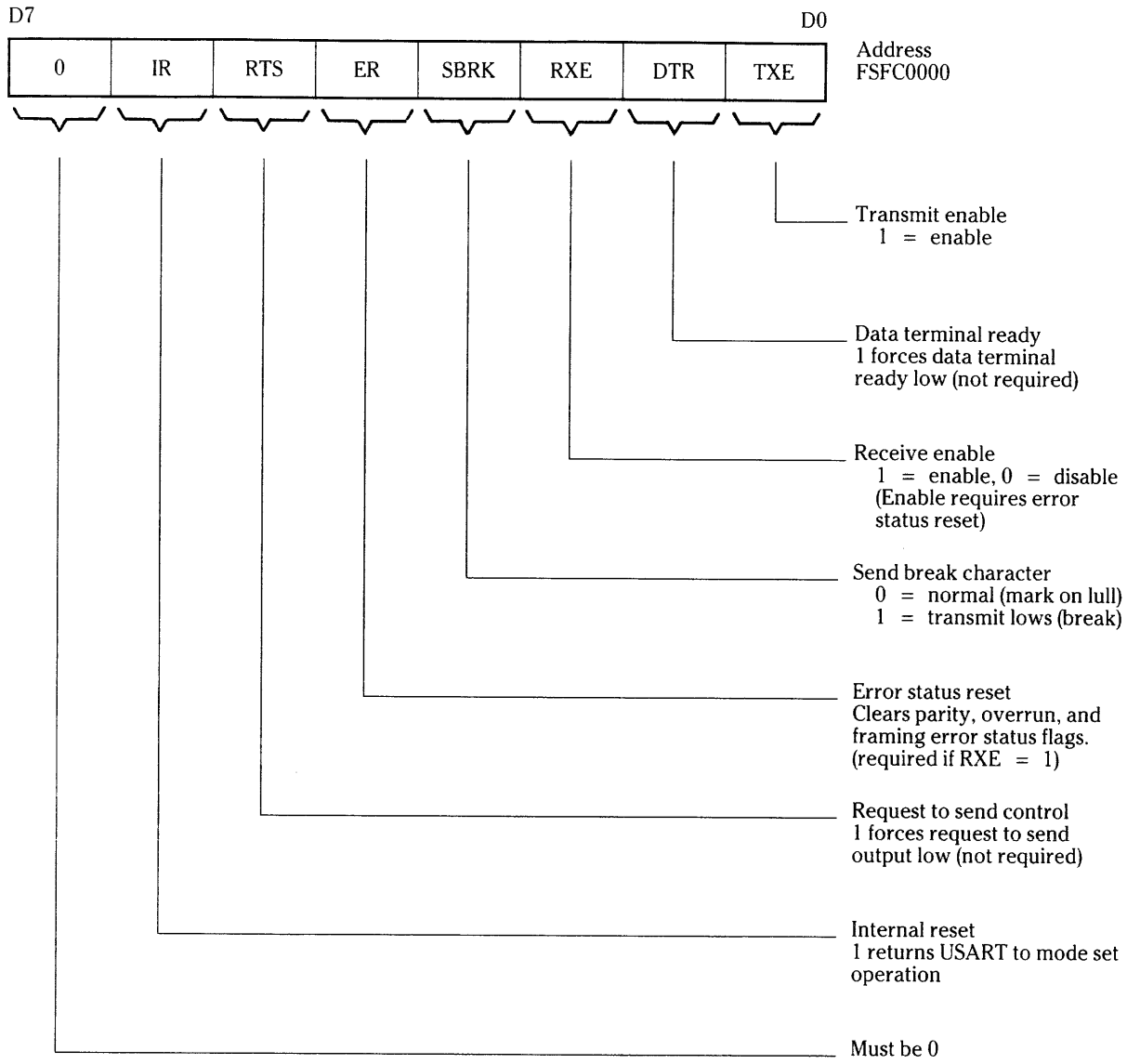
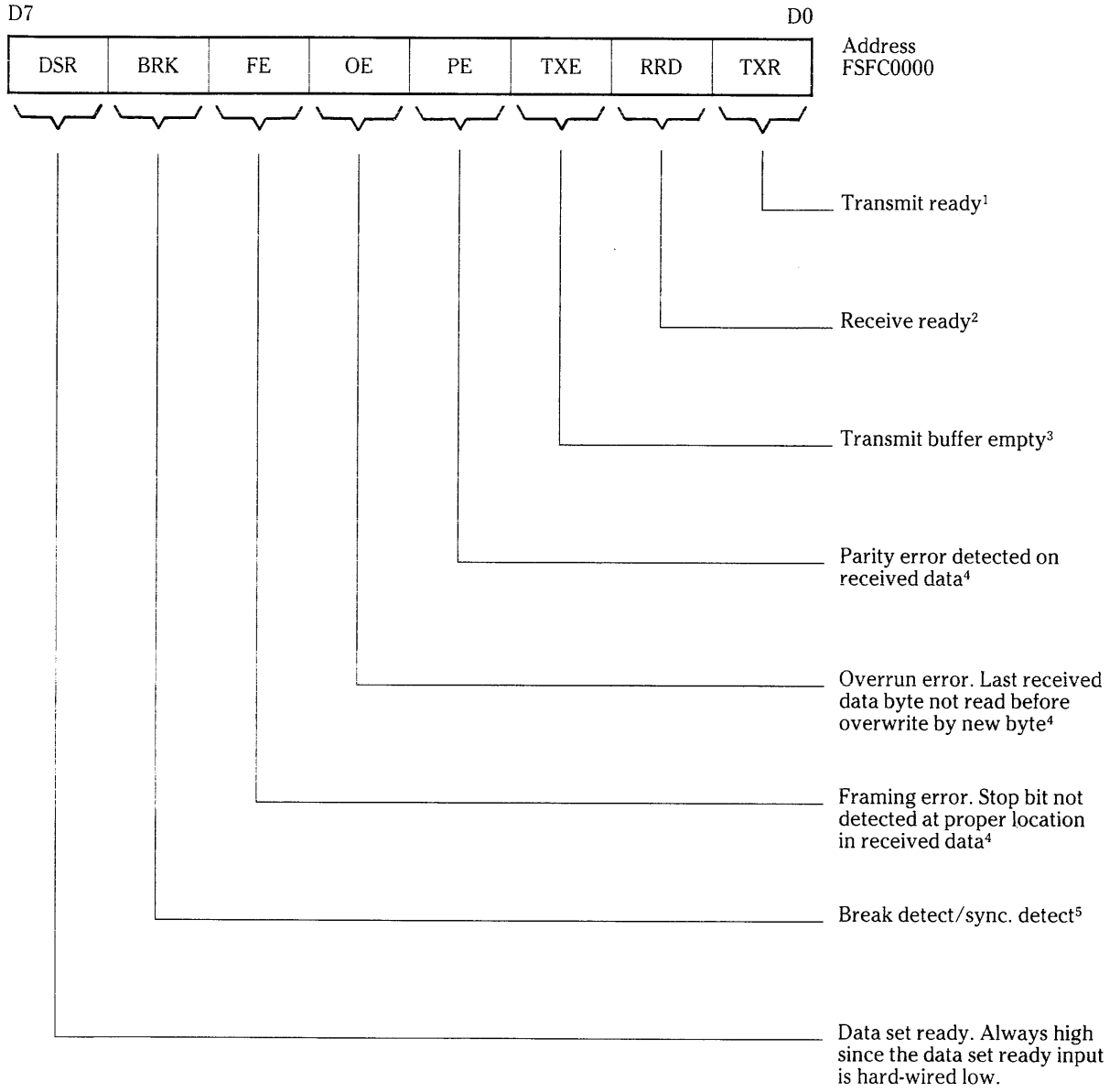


Figure 4-24 USART Status Byte Format



NOTES:

[1] TXR status bit is not conditional on the CTS– (clear to send) signal input to USART and the transmit enable bit of the command byte.

[2] Indicates that the USART receive buffer has completed the serial-to-parallel shift and has a character ready for transmission on the I bus.

[3] Indicates no more data in the USART transmit buffer or that the transmitter is disabled (not connected).

[4] Status errors FE, OE, and PE can be reset by the error status reset bit in the command instruction byte.

[5] Break detect is for asynchronous reception, and synchronous detect is for synchronous reception. This is a programmable function (not connected).

USART Transmit and Receive Data Format Figure 4-25 shows the format of the transmit and receive data bytes. All processing of start, parity, and stop bits is internal to the USART. These bits do not appear on the I bus of the SI board.

Special Chord Detector

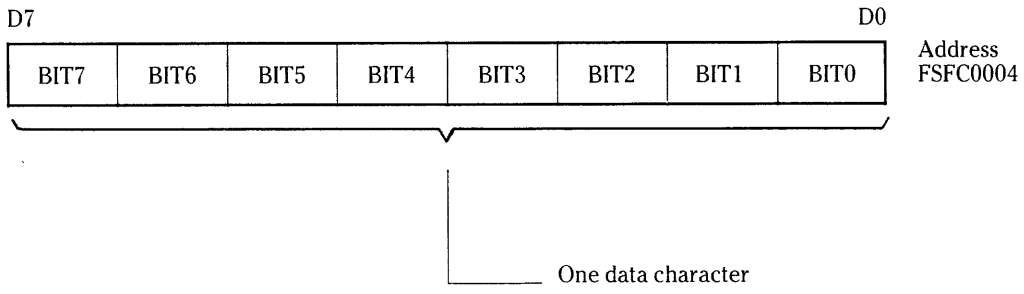
4.4.13 The special chord detector on the SI board provides the capability for an operator at the keyboard to force either a complete system reset or to initiate one of several types of reboot events. Keyboard control of the system reset (reboot) is a convenience to the operator and permits additional freedom to locate the system enclosure away from the keyboard and monitor. The special chord detector is a state machine on the SI board that monitors the incoming keyboard data stream and recognizes a system reset or reboot code. Depending on the code, the special chord detector either generates a system reset signal to the system backplane or an interrupt that is applied to the event generator.

To force a system reset or a reboot event, the keyboard operator presses a specific set of keys simultaneously (a chord) to generate the code that specifies either a system reset or a type of reboot event. The key chord META-CTRL-META-CTRL-ABORT resets the entire system as if the system were just powered up. (The type of event generated is not limited to a reboot event but can be other types of events as dictated by the system software.) The purpose of using a special chord to generate the reset or event code is twofold:

1. It prevents an accidental reset or reboot occurrence that might result from pressing a dedicated key or from entering a reserved keystroke sequence.

2. It forces a local scan/communication microcomputer located at the keyboard into a special state in preparation for sending a special escape sequence.

Figure 4-25 USART Transmit and Receive Data Byte Format



After recognizing the chord, the keyboard microcomputer sends the first two codes of a three-code special escape sequence and then scans for a single differentiating keystroke that selects either a type of event or a system reset. When the keyboard operator enters the differentiating keystroke, the keyboard microcomputer completes the special escape sequence by sending the third code.

After transmitting this special escape sequence, the keyboard microcomputer ignores keyboard entries except for special chords. Recognition of special chords is allowed to prevent a lockup if the operator attempts a reboot from a device that is not available for booting. If the system fails to reboot, the operator can initiate a reboot from a different device or select a full system reset. The keyboard microcomputer resumes normal acceptance of keyboard entries after it receives a reset command as part of self-test procedures or a system reset.

A special chord sequence is irreversible after the operator presses the set of keys that form the chord. System software can include a time-out provision that takes default action if the operator fails to press a differentiating key. The lack of an easy way to back out makes it imperative that the chord keys be selected carefully.

Refer to the *Explorer Operations Guide* for a list of keys that form the special chord. These keys are not relevant to the SI board since logic at the keyboard recognizes the chord and sends special escape sequence codes to the SI board. In sampling the incoming keyboard data stream, the special chord detector recognizes only two categories of differentiating keystrokes or codes. The code is either the code that specifies the system reset command or it is any other code. The special chord detector responds to any code other than the reset command code by generating an event. It is the responsibility of the system processor to determine the exact type of event by polling the keyboard data stream.

The special chord detector does not receive system reset and can be cleared only by turning the power off and then turning the power back on.

Fiber-Optic Interface

4.4.14 The fiber-optic data link consists of a transmit serial data transfer path and receive serial data transfer path between the system enclosure and the monitor. The transmit transfer path, designated channel A, is a wide-band channel that originates at the fiber-optic interface on the SI board and transfers data via the fiber-optic interface board and the fiber-optic cable to the video display and related devices in the monitor. Channel A data to the monitor includes:

- Video display signals (including horizontal and vertical sweep frequency synchronization signals)
- Sound control data and parity to the sound generator
- Speech data and parity to the speech synthesizer

- Keyboard control data
- Timing for the receive serial data transfer path
- Other control and test bits

The receive serial data transfer path, designated channel B, originates at the monitor and transfers data via the fiber-optic cable and the fiber-optic interface board to the fiber-optic interface on the SI board. Channel B, a slower channel than channel A, carries data that includes:

- Mouse motion and keyswitch data
- Keyboard output data
- Digitized voice input data
- Sound parity-error-detected bit
- Loopback test bit

The channel A transmit end of the SI board fiber-optic interface includes circuits that convert multiple parallel data inputs into a time-division multiplexed serial format. The channel B receive end of the interface includes circuits that regenerate the timing and convert the received serial input into multiple parallel outputs. Refer to the *Monitor Interface Board Specification* for a description of the overall data link and of the data link interface to the monitor, sound generator, keyboard, and speech synthesizer.

The following paragraphs describe the serial data formats for both channels and the fiber-optic interface logic (channel A transmit, channel B receive) on the SI board.

Channel A (SI to Monitor) Description

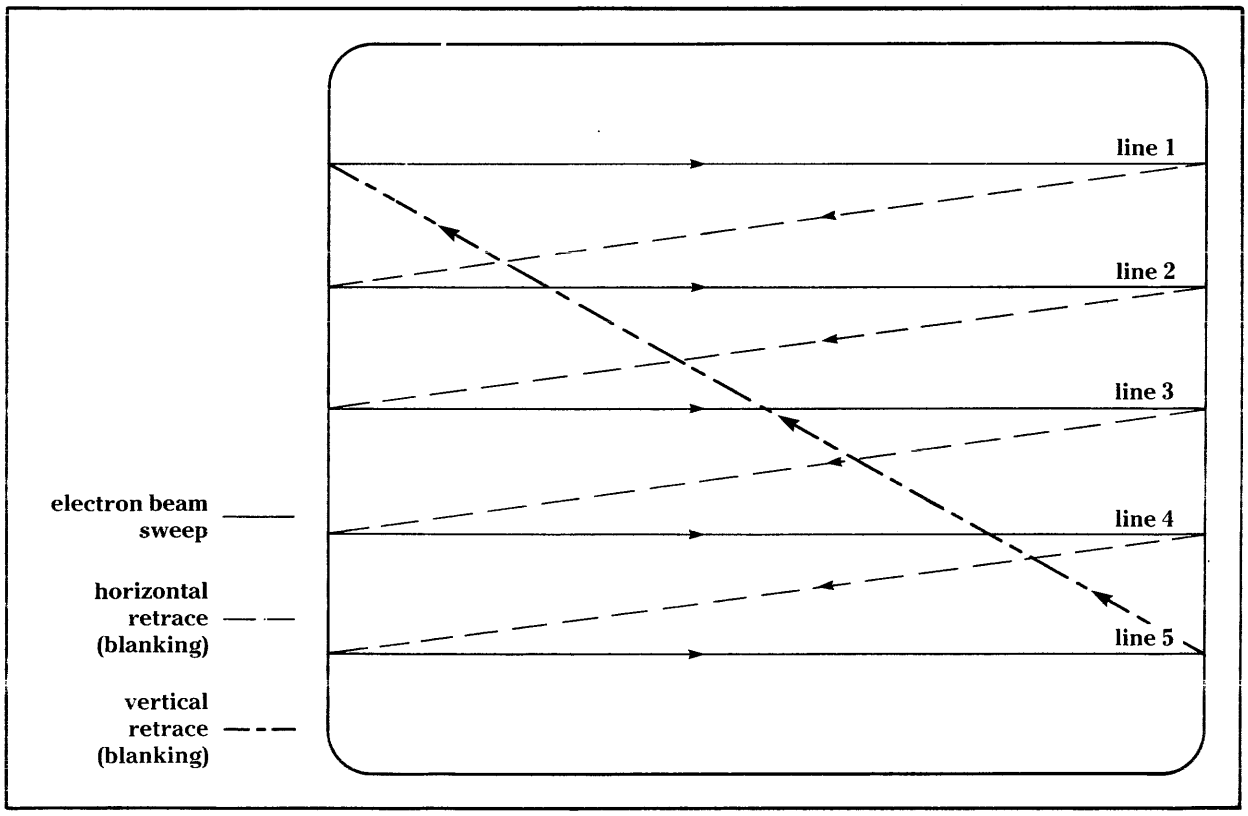
4.4.14.1 Video bandwith and video timing requirements determine the basic timing and format of the wideband channel A signal. Other data is worked into the periods that are not explicitly required for sending video. Table 4-17 summarizes the current monitor timing specifications.

Video Requirements Figure 4-26 shows a simplified version of the non-interlaced raster-scan display used for the high-resolution monitor. The electron beam sweeps across the face of the display in a regular pattern, left to right and top to bottom, at a sweep rate of 50.52 kilohertz. At the end of each line, the beam turns off as it returns to the start of the next line. The return sweep is called the *horizontal retrace.* Horizontal blanking turns off the beam for the horizontal retrace. When the beam reaches the right end of the last line in the frame, the beam turns off (using vertical blanking) and returns to the upper left corner of the frame. This return motion is called the *vertical retrace.* When the beam completes the vertical retrace, it has painted one complete frame.

Table 4-17 Monitor Timing

Item	*Frequency*	*Period*
Sweep rate	50.52 kHz	19.6 μs
Frame rate	60.00 Hz	16.666 ms
Pixel clock	67.8990 MHz	14.7277 ns
Pixel clock/32	2.1218437 MHz	471.2864 ns
Pixel clock/8	8.487375 MHz	117.8216 ns
Pixel clock/4	16.97475 MHz	58.9108 ns
	Active	***Total***
Lines per frame	808	842
Pixels per line	1024	1344
	Time	***Line Equivalent***
Horizontal retrace	4.7129 μs	—
Vertical retrace	—	34 lines
Vertical sync delay	—	17 lines
Vertical sync width	—	3 lines minimum

Figure 4-26 Simplified Raster-Scan Display



The display controller repaints (refreshes) the screen continually to keep the image bright. The persistence of the phosphor and the rate at which the image is refreshed work together to present a flicker-free image that appears stationary to the human eye.

The display controller provides pulses that identify the start times of the horizontal and vertical blanking periods. This synchronizing information goes over the data link to the monitor interface board. The monitor interface board includes logic that derives standard vertical and horizontal synchronization pulses that synchronize the monitor sweep circuits to the video signal.

During the active period of a sweep, the video data occupies fiber-optic channel A. During the horizontal retrace periods (while the screen is blanked), the data link transmits serial data to the other logic on the monitor interface board. This data is encoded in a biphase Manchester format.

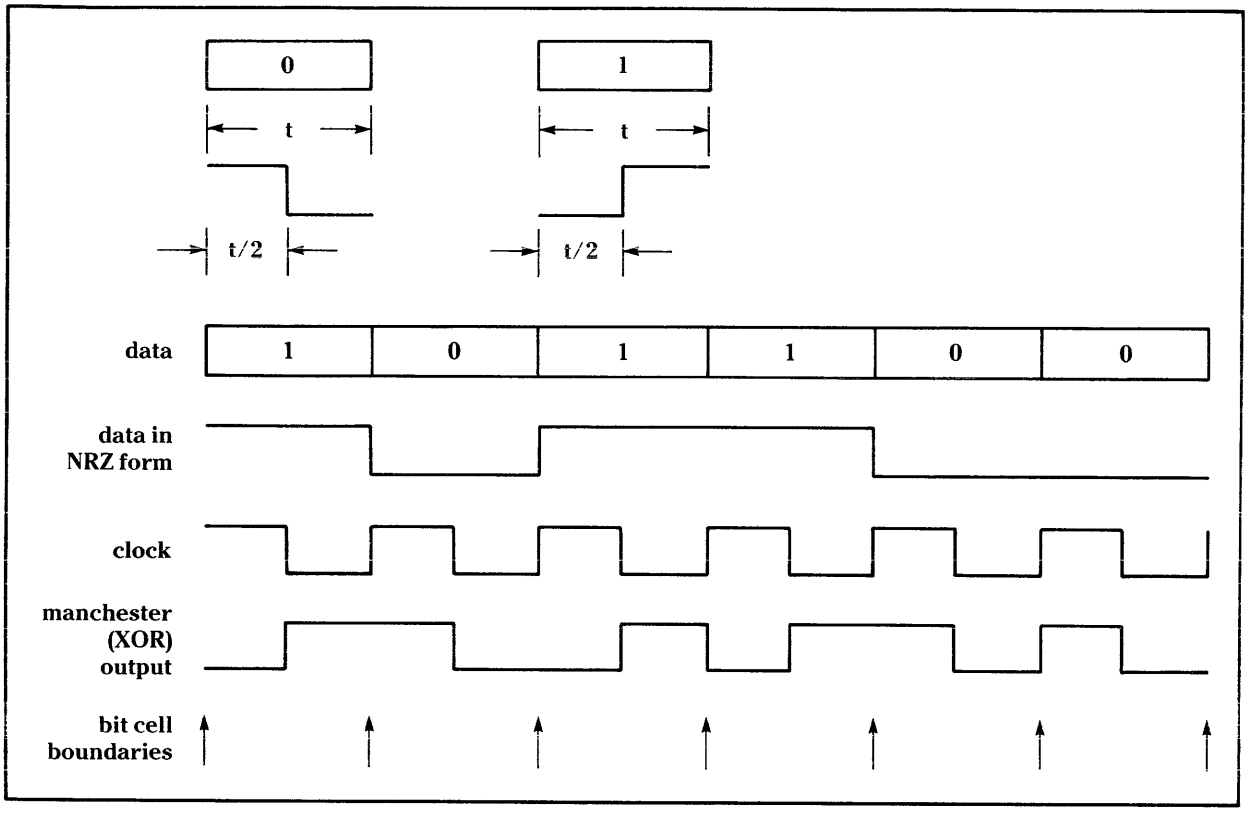
Manchester Encoding Figure 4-27 shows the Manchester coding rules. A logical 1 is represented by a signal that is low for the first half of the bit cell and high the the second half. A logical 0 is represented by a signal that is high for the first half of a bit cell and low for the second half. In terms of transitions, a logical 1 has a positive-going transition in the middle of the bit cell and a logical 0 has a negative-going transition in the middle of the bit cell. Consecutive 1s or consecutive 0s have an additional transition at the bit cell boundary.

The Manchester code is effectively a logical XOR of the data stream and a clock signal at the data frequency. The standard Manchester encoder is an XOR gate with data and clock inputs. The XOR gate is followed by a flip-flop to smooth (deglitch) the output waveform.

Manchester code is self-clocking because of the signal transition in every bit cell. The self-clocking feature of the code simplifies clock/data recovery and reduces the bit error rate (BER). Also, the duty cycle of a Manchester waveform is closer to fifty percent than the NRZ data that it represents. A fifty-percent duty cycle is desirable to prevent dc offset and consequent pulse distortion in the data link.

Single-Frame Data Format (Channel A) Figure 4-28 shows the simplified format of the channel A transmissions for a single video frame. A frame consists of 808 active video scan lines with horizontal retrace and blanking between successive lines. Vertical retrace and blanking occur at the end of the frame. The time required for a vertical retrace is equivalent to another 34 scan lines, for a total of 842 lines. A horizontal blanking pulse occurs at the end of each of these 34 lines, just as though they were active video.

Figure 4-27 Manchester Encoding

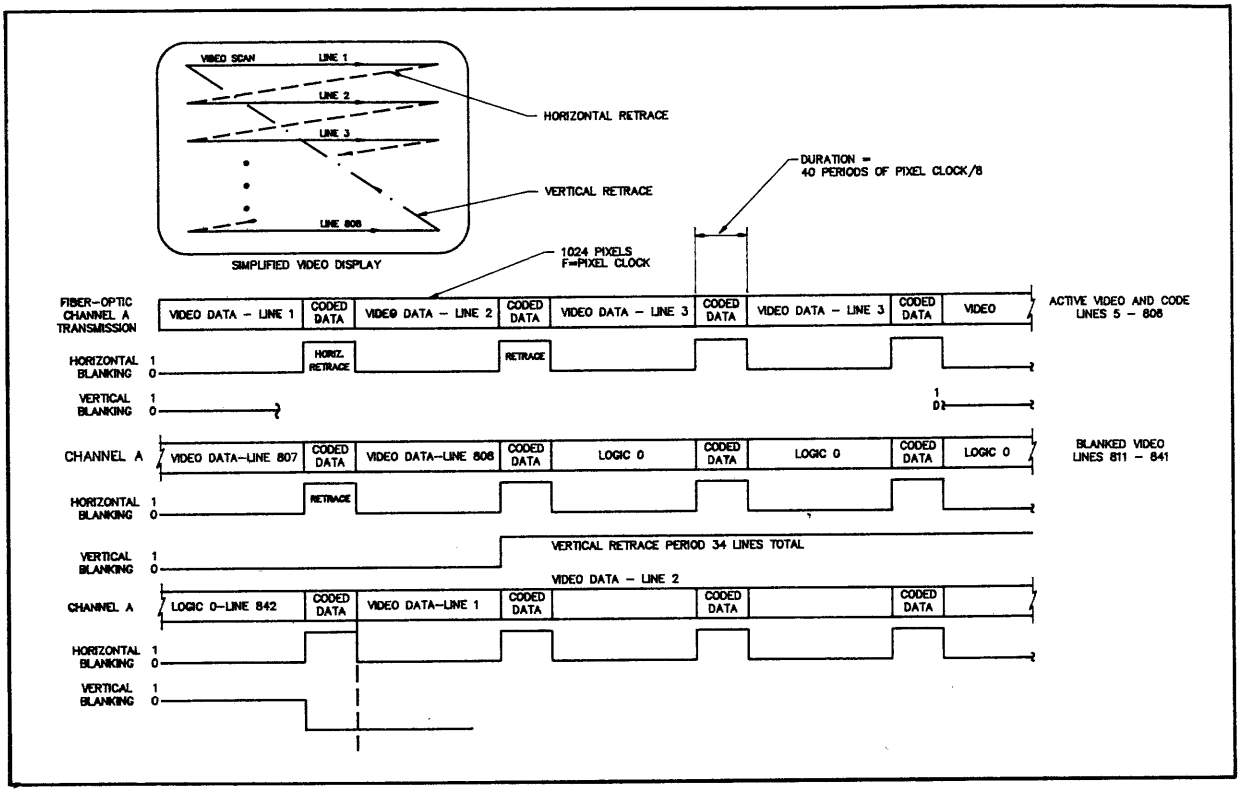


Each line of video consists of 1024 pixels. Both the instantaneous frequency and duty cycle are pattern-sensitive. The peak frequency is equal to the pixel clock frequency from the display controller. At the end of each video line, horizontal blanking turns on and the data block is multiplexed onto the link. The Manchester-encoded waveform data frequency is pixel clock/8.

The data block transmitted following the first blanked video line (line 809) includes a vertical blanking bit that provides a reference for regeneration of the vertical synchronization. Vertical retrace is equivalent to 34 lines with no video but with data blocks transmitted during horizontal blanking periods. A logic 0 replaces the video during the vertical retrace.

Figure 4-29 shows the bit assignments for the data block sent during each horizontal blanking period. The first two bit periods are dead time with no encoded data transmitted. The dead time allows some tolerance in the receive circuits and in the monitor sweep timing. The first two encoded bits are logic 0 dummy bits. The next 7 bits (1110010) form a data link synchronization pattern. This pattern identifies the start of a valid data stream. Also, this pattern occurs once per horizontal blanking period and is used to develop a horizontal synchronous output to the monitor sweep circuits.

Figure 4-28 Simplified Channel A Format and Timing



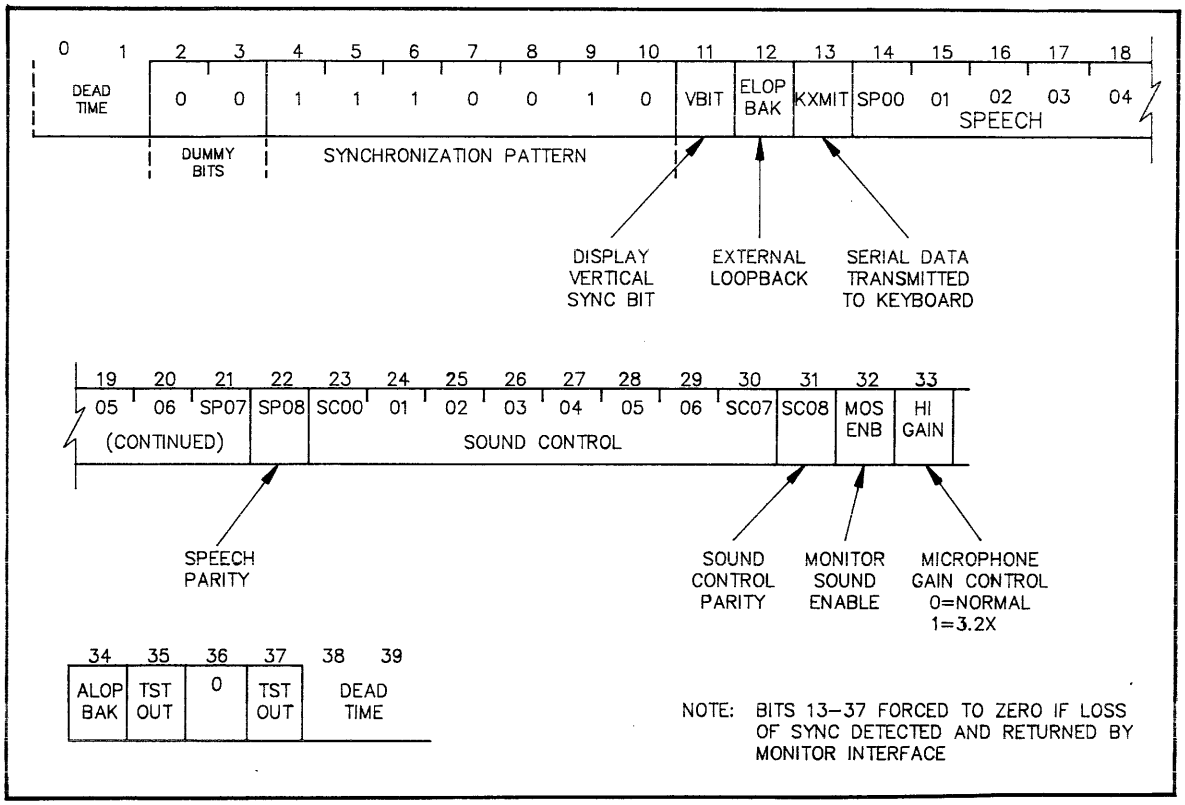
The first active data bit is the vertical synchronization bit. This bit is set following line 809 to provide a reference for the vertical retrace. After a set of fixed delays, this bit enables a vertical synchronization pulse to the monitor.

ELOPBAK is a keyboard loopback control bit. When set, it causes the keyboard UART to loop back the KXMIT data on the keyboard output line. Keyboard output is multiplexed on low-frequency channel B.

KXMIT is the 2400-baud serial data stream transmitted to the keyboard UART in the monitor. Keyboard data includes parity in the UART data format. KXMIT has a much lower data rate than the optical link and is oversampled.

The next 9 bits consist of 8 bits of speech data (SP00 through SP07, with the LSB first) and an odd parity bit (SP08) for speech. Speech parity assures the integrity of the speech transmission against single-bit errors. Eight bits of sound control data (SC00 through SC07 with the LSB first) and a sound control parity bit (SC08) follow the speech data. MOSENB, the monitor sound enable, is a monitor audio on/off control.

Figure 4-29 Bit Assignments for Channel A Multiplexed Data



The HIGAIN bit sets the gain of the voice input microphone amplifiers at either normal gain (HIGAIN = 0) or high (3.2X) gain. These two gain levels allow the operator either to work close to the microphone or to lean away from the microphone.

ALOPBAK (the analog loopback) enables a special local test of the microphone input, codec, and audio output circuits in the monitor. ALOPBAK, when set, connects the PCM-encoded voice output of the codec back to the PCM speech decoder input. The effect is to create a miniature PA system at the monitor. This causes undistorted speech to come out of the speaker when the operator speaks into the microphone. No data transfer to or from the SIB is involved. If the speech is distorted, the error is probably in the codec circuit.

TSTOUT (test out) appears at two bit positions with a spare bit (zero) between them. Although TSTOUT is not implemented in the current generation of the monitor interface board, these test-out bits are reserved to serve as loopback bits to verify the entire fiber-optic link.

The last two encoded bits are logic 0 dummy bits. A dead time of two bit periods acts as a gap to compensate for the timing tolerances allowed between monitor sweep and data transmissions.

If the MIB loses synchronization to the channel A data stream, the channel A loss of synchronization (ALOSYNC) is reported on channel B. ALOSYNC forces all data bits from KXMIT through TSTOUT (bits 13 through 37) to zeros so that the synchronization pattern and the vertical synchronization bit (VBIT) keep transmitting until the monitor interface board can synchronize again. Forcing the other bits to zero assures that any misdecoded data has minimum effect in the MIB.

Channel A Transmit Section Figure 4-30 is a simplified block diagram of the channel A transmit logic. All inputs to the channel A transmit logic come from two sources: the bit-mapped display controller and the keyboard/mouse/audio interface logic. Clock, horizontal synchronization, vertical synchronization, and video come from the bit-mapped display controller. Speech data and sound control data come from data registers and parity generators in the keyboard/mouse/audio interface logic. Keyboard data comes from the keyboard USART. The keyboard loopback control, ELOPBAK, comes from the diagnostic control register, and the test bit comes from the diagnostic data register.

The shift register acts as a data sampler and a parallel-to-serial converter. It accomplishes those functions by parallel loading at the beginning of retrace for each horizontal sweep and then shifting the data out at a much higher clock rate derived from the pixel clock. The serial data output goes through a Manchester encoding circuit to one input of a 3:1 multiplexer. The other inputs are video data and the logic 0 that substitutes for video during a vertical retrace.

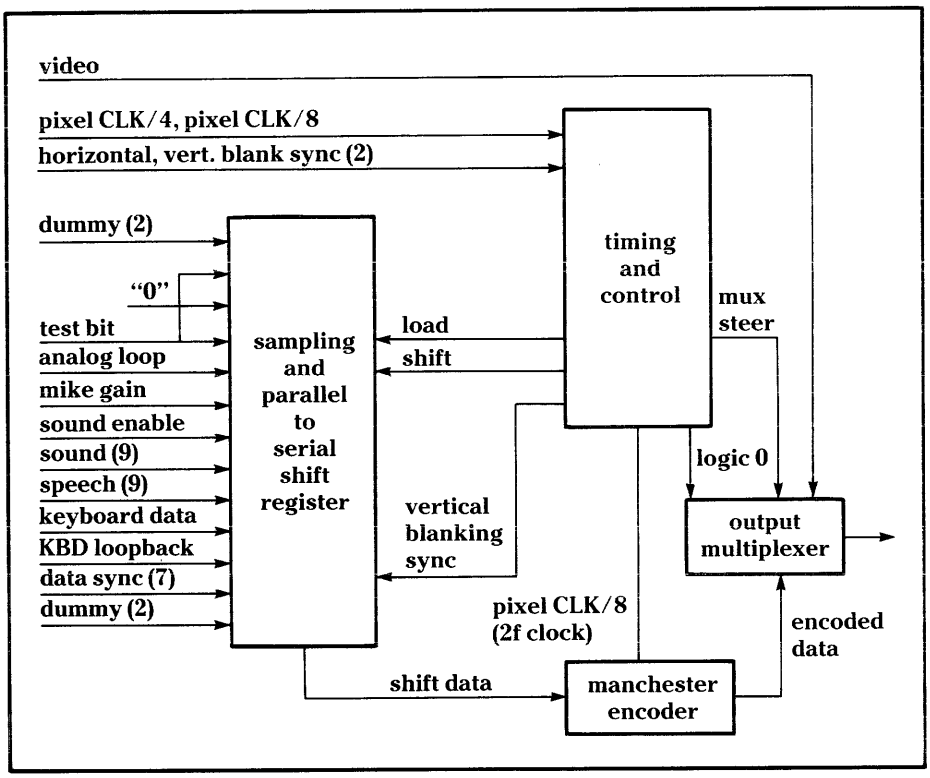
Timing and control logic generates the square wave and controls load timing, shift timing, and the steering of the output multiplexer logic.

NOTE: Channel B operates constantly, regardless of whether there are any commands to be sent to the monitor enclosure. The data channel shuts down on reset and cannot restart until the bit-mapped display controller resumes operation. Monitor initialization takes place through the data and programming commands transmitted on channel A.

Channel B (Monitor to SI) Description

4.4.14.2 Channel B, the downlink channel from the monitor enclosure to the SI board, has a much lower data rate than channel A. Channel B data includes the 2400-baud output of the keyboard, four parallel mouse motion bits, mouse keyswitch outputs, digitized voice data, an error bit, and a data link synchronization pattern. All the data on channel B (except the synchronization pattern) is encoded in Manchester format. (Refer to the channel A description in a previous paragraph for the Manchester coding rules.)

Figure 4-30 **Channel A Transmit Block Diagram**

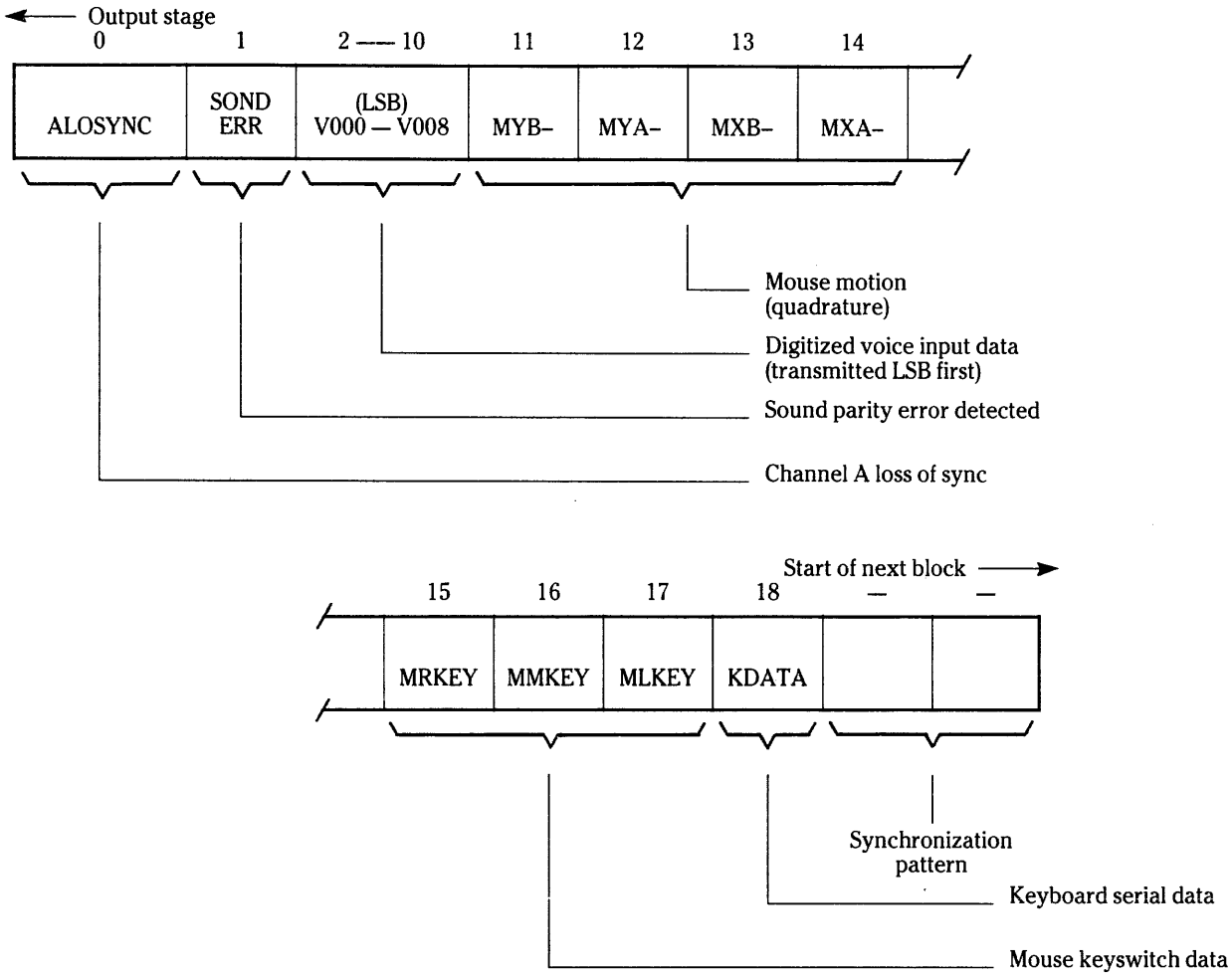


Channel B transmit timing comes from a phase-locked loop (PLL) circuit that recovers the clock from channel A. Channel B data runs at a fraction of the channel A encoded data rate. A failure in channel A causes the PLL to free-run, and channel B continues to operate.

Channel B Data Format Figure 4-31 shows the format of the channel B data. Bits appear in the order of transmission with the first bit at the left. Each block of channel B data occupies the entire time between successive horizontal sweeps of the monitor. Transmission is continuous, with new data samples loaded at each synchronization time. With a horizontal sweep rate of 50.52 kilohertz and 21 bits (including synchronization), the data rate is 1.06092 megahertz. New samples are loaded every 19.7941 microseconds. Each bit period is 0.9425781 microseconds.

ALOSYNC is the monitor interface board loss-of-synchronization bit for the high-speed fiber-optic data channel A. If the channel A receive circuits in the MIB are out of synchronization, ALOSYNC goes active (high). ALOSYNC forces the data fields of the channel A data transmission to zeros. Any data decoded wrong in the MIB due to a synchronization fault is decoded as a zero.

Figure 4-31 Channel B (Monitor-to-SI) Data Format



SONDERR is generated by the parity generator and comparator circuits associated with the sound generator. SONDERR is set if the local parity calculated at the sound generator differs from the transmitted sound control parity.

VO00 through VO08 is a 9-bit block of digitized voice input data from the codec (voice coder/decoder) on the monitor interface board. VO08 serves as a voice data-present bit. A 50.52-kilohertz sample rate oversamples the 8-kilohertz codec data rate by more than 6 to 1. The voice data-present bit identifies each new nonredundant data sample.

MYB– and MYA– are the two quadrature waveforms (in active-low form) that represent mouse motion in the Y direction. MXB– and MXA– represent mouse motion in the X direction. A 50.52-kilohertz sampling rate oversamples this data by more than 8 to 1.

MLKY, MMKY, and MRKY are the states of the LEFT, MIDDLE, and RIGHT keyswitches (buttons) on the mouse body.

KDATA is serial 2400-baud data from the keyboard. At a 50.52-kilohertz sample rate, keyboard data is oversampled by 21 to 1.

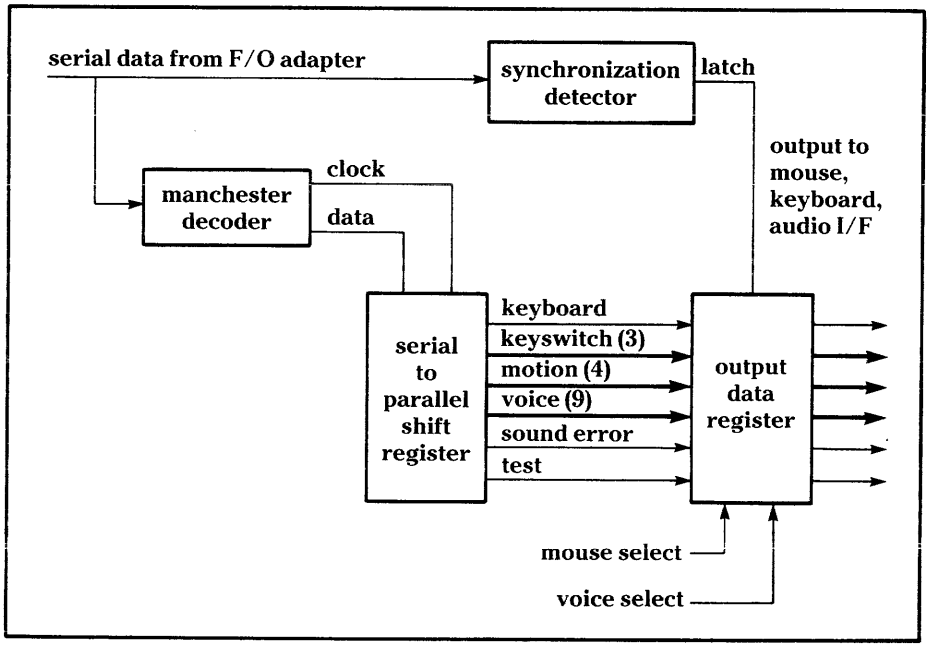
Manchester encoding rules call for a minimum of one transition per bit cell. The data synchronization pattern consists of two bit periods with no transmission, which is an illegal pattern. Recognition of the illegal pattern indicates that the data block is complete. A new data block starts at the end of the synchronization period.

Channel B Receive Section The channel B signal (from the optic adapter board) goes to a synchronization detector and a Manchester decoder as shown in Figure 4-32. The clock and data lines from the decoder feed a demultiplexing shift register. Shift register outputs are loaded into data registers by the detected synchronization signal.

RS-232C Serial Data Port

4.4.15 The RS-232C serial data port uses a fully programmable Zilog Z8530 serial communications controller along with line drivers and line receivers to provide the capability for serial, half-duplex or full-duplex, asynchronous or synchronous communication at all standard baud rates from 50 to 19 200. The line drivers and line receivers at the output and input of the Z8530 provide signals at levels that are compatible with the Electronic Industry Association (EIA) RS-232C standards. All standard EIA modem and terminal control signals are provided.

Figure 4-32 **Channel B Receive Block Diagram**



The Z8530 has two independent channels that can work together or independently, however only one channel is operational on the SI board. In the configuration used for the RS-232C serial port, channel A of the Z8530 is dedicated to communication; channel B is wired to provide some less frequently used support functions. Figure 4-33 shows the input/output connections to the Z8530.

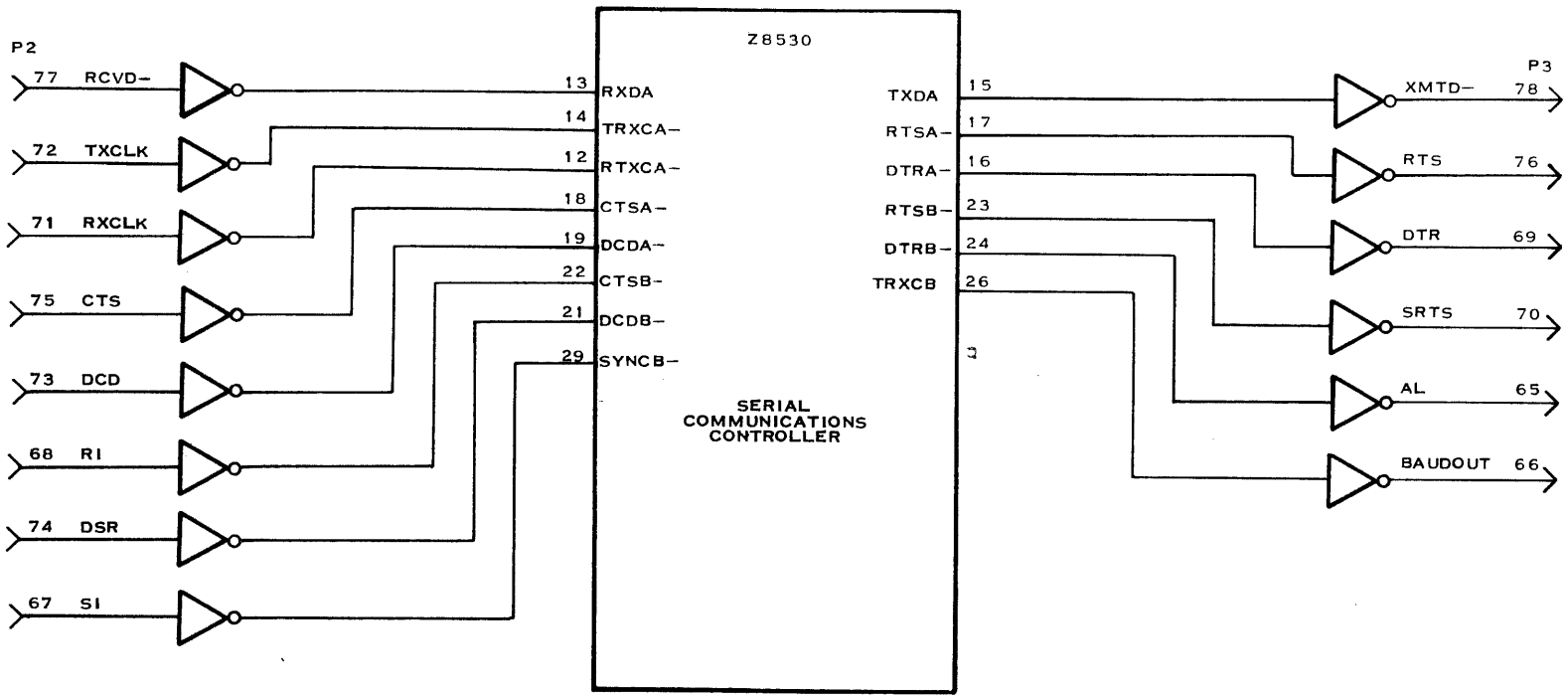
Z8530 available capabilities include:

- General features
  - Four-deep first-in, first-out buffering of receiver data registers
  - Double buffering of transmitter data registers
  - Non-return-to-zero (NRZ), non-return-to-zero-inverted (NRZI), or frequency-modulation (FM) encoding/decoding
  - Baud rate generation
  - Digital phase-locked loop (PLL) for FM and NRZI data recovery
  - Auto echo
  - Internal loopback
  - Polled- or interrupt-based operation
- Asynchronous features:
  - 5, 6, 7, or 8 bits-per-character
  - 1, 1.5, or 2 stop bits-per-character
  - Odd or even parity, and parity error detection
  - X1, X16, X32, or X64 clock division ratio
  - Break generation and detection
  - Overrun and framing error detection

- Synchronous byte-oriented features:
  - Internal character synchronization (external synchronous line omitted in SI board)
  - 1 or 2 synchronous characters, 6-bit or 8-bit (monosynchronous), or single 12-bit synchronous pattern (bisynchronous)
  - Automatic synchronous character generation/deletion
  - CRC-16 or CRC-CCITT error checking with ones or zeros preset
- Synchronous bit-oriented features (SDLC, HDLC protocols)
  - Abort sequence generation and checking
  - Automatic zero insertion and deletion
  - Automatic flag insertion between messages
  - Address field recognition
  - CRC-16 or CRC-CCITT error checking with ones or zeros preset
  - SDLC loop mode with end-of-poll (EOP) recognition/loop entry and exit

Parallel 8-bit byte data, control, and status transmissions to and from the Z8530 occur over the buffered M data bus. Internal address bit IADDR03 determines whether the transmission is a data transfer or whether it is a control or status transfer. Internal address bit IADDR02 selects either channel A or channel B registers. A read or write operation to the Z8530 with address bit IADDR04 high acknowledges an interrupt. This is ordinarily a read operation directed to read register 2 of channel B, which includes interrupt status bits and an unused interrupt vector.

Figure 4-33 Z8530 Input/Output Connections



Port Input and Output Signals

4.4.15.1 The following are the buffered EIA input/output signals at the RS-232C port interface brought to connector P3 of the SI board.

- Receive data (RCVD–)
- Clear to send (CTS)
- Ring indicator (RI)
- Data carrier detect (DCD)
- Data set ready (DSR)
- Transmit data (XMTD–)
- Request to send (RTS)
- Data terminal ready (DTR)
- Secondary request to send (SRTS)
- Analog loop-back (AL)
- Speed indicator (SI)
- Transmitter frequency out (BAUDOUT))
- Transmit clock (TXCLK)
- Receive clock (RXCLK)

Table 4-18 lists the functions of each of the EIA input and output signals that are generated or received by the RS-232C port.

Table 4-18 RS-232C Port Input and Output Signals

Signal Name	*Description*
XMTD– (output)	Transmit data. The port generates the serial transmit data stream from the data byte written into the transmit data register for transmission to the external device. This output is held in the marking condition during intervals between characters and when no data is being transmitted. The following four signals must be in the on condition before data can be transmitted: Request to send (RTS) Clear to send (CTS) Data set ready (DSR) Data terminal ready (DTR)

Table 4-18 **RS-232C Port Input and Output Signals (Continued)**

Signal Name	*Description*
RCVD– (input)	Receive data. Serial data is received on this line consisting of start bits, stop bits, parity bit (if enabled), and a data byte. The port extracts the data byte from the serial data stream, converts it to a parallel format, and places it in the receive data register.
RTS (output)	Request to send. Can be asserted by central processing unit (CPU) to indicate that the port is ready to transmit data.
CTS (input)	Clear to send. Can be sampled by the CPU. It is generally asserted by the external device (modem) to indicate that the modem is ready to transmit data that has been transmitted to the modem from the port.
DSR (input)	Data set ready. Can be sampled by the CPU. It is generally asserted by an external device to indicate that the device is connected to a data channel and that all control circuits are valid.
DCD (input)	Data carrier detect. Can be sampled by the CPU. Generally asserted by an external device to indicate that it is receiving a signal that meets its suitability criteria and that receive data is valid.
SRTS (output)	Secondary request to send. Can be asserted by the CPU. Used to instruct the external device to transmit a tone on the secondary channel (also called the reverse, supervisory channel, or backward channel).
SI (input)	Speed indicator. Can be sampled by the CPU. Generally asserted by the external device to indicate which of two speeds is active. The on condition indicates that the device is in the high-speed mode.
DTR (output)	Data terminal ready. Can be asserted by the CPU. Asserted to indicate that the port is ready to communicate with an external device.
RI (input)	Ring Indicator. Can be sampled by the CPU. Generally asserted by an external device to indicate the reception of a ringing signal on the communication channel.

Table 4-18 **RS-232C Port Input and Output Signals (Continued)**

Signal Name	*Description*
AL (output)	Analog loopback. Can be asserted by the CPU. Generally asserted to instruct compatible modems to disconnect from the telephone network and connect the transmitter output to the receiver input.
BAUDOUT (output)	Transmitter signal element timing. Transmit clock to drive the transmitter section of a synchronous modem. High-to-low transition identifies the center of each signal element.
RXCLK (input)	Receiver signal timing. Receive clock from the synchronous modem. High-to-low transition identifies the center of each signal element.
TXCLK (input)	Transmitter clock. Transmitter clock from synchronous modem returned to RS-232C port. Low-to-high transition occurs at the transition between signal elements.

RS-232C Port Programming

4.4.15.2 The following paragraphs contain a summary of programming information for the Z8530 device. Refer to the *Z8030/Z8530 SCC Serial Communications Controller Technical Manual* for complete programming data for the Z8530.

The Z8530 includes two independent channels that are designated channels A and B. The SI board is wired to allow all channel A functions, and all communications occur through this channel. Channel B is wired only to provide some auxiliary control line input/output and is never used as a communications channel. As implemented, the port includes 16 write and 8 read registers. Of the write registers, 9 are used for control, 2 for synchronous character generation, 2 for baud rate generation, 2 for interrupt handling, and 1 for the transmit data buffer. Of the 8 read registers, 4 indicate status, 2 are used by the baud rate generator, 1 is the receive data buffer, and 1 is the interrupt pending register.

An A/B– channel-select input to the Z8530 selects the channel in which a read or write operation occurs. The D/C– data/control input determines whether the access is for data or for control/status information. The receive data buffers and transmit data buffers are directly accessible when the D/C– data/control input is high. An active low signal applied to either the WR– or RD– input selects the Z8530 for write or read operations, respectively.

Access to the control and status registers is more complicated. Read register 0 (RR0) and write register 0 (WR0) are the only registers directly accessible with a one-step read or write operation. Access to the other registers is a two-step operation:

1. A write to WR0 loads a pointer code that selects a register for the next data bus access.

2. A read or write operation transfers the data to/from the selected register. The pointer bits in write register 0 automatically clear after the second access.

Table 4-19 and Table 4-20 summarize the functions of the Z8530 read and write registers.

Table 4-19 **Summary of Z8530 Read Register Functions**

Register Number	*Function*
RR0	Transmit/receive buffer status and external status
RR1	Special receive condition status
RR2	This register is shared by both channels: Modified interrupt vector (includes status), channel B Unmodified interrupt vector, channel A
RR3	Interrupt pending bits (channel A register only)
RR8	Receive data buffer
RR10	Miscellaneous status
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External/status interrupt information

Table 4-20 **Summary of Z8530 Write Register Functions**

Register Number	*Function*
WR0	Command register. Register pointers, CRC initialize, intialization commands for various modes
WR1	Transmit/receive interrupt and data transfer mode definition
WR2	Interrupt vector (accessed through either channel)
WR3	Receive parameters and control
WR4	Transmit/receive miscellaneous parameters and modes
WR5	Transmit parameters and control
WR6	Synchronous characters or SDLC address field
WR7	Synchronous characters or SDLC flag
WR8	Transmit data buffer
WR9	Master interrupt control and reset (accessed through either channel)
WR10	Miscellaneous transmitter/receiver control bits
WR11	Clock mode control
WR12	Lower byte of baud rate generator time constant
WR13	Upper byte of baud rate generator time constant
WR14	Miscellaneous control bits
WR15	External/status interrupt control

Table 4-21 lists the addresses for each of the directly accessible RS-232C port registers.

Table 4-21 **RS-232C Register Addresses**

Register	*Address (Hexadecimal)*
Channel A receive data buffer (RR8)	FSFB000C
Channel A transmit data buffer (WR8)	FSFB000C
Channel A buffer status and external status (RR0)	FSFB0004
Channel A pointer register, control data (WR0)*	FSFB0004
Channel B receive and transmit data buffers (RR8 and WR8) are unused	
Channel B buffer status and external status (RR0)	FSFB0000
Channel B pointer register, control data (WR0)*	FSFB0000
Interrupt acknowledge address (write only). Interrupt status available by multistep read operations to internal registers.	FSFB0010

NOTE:

* Any numbered register is accessible on the first read or write after loading the register number code in the pointer register (WR0) of that channel. The NuBus/I-bus address is the same as the pointer register address.

Programmable Baud Rate Generator The programmable baud rate generator derives any desired baud rate from the master clock input frequency to the Z8530 by counting down a preloaded time constant at the clock rate. The master clock frequency is 2.4576 megahertz. Each time the time constant counter reaches zero, it puts out a pulse, reloads automatically, and repeats the processs. A time constant occupies two 8-bit registers. The upper (most significant) byte is loaded into WR13 and the lower (least significant) byte is loaded into WR12. The time constant can be verified by reading back from RR13 and RR12.

A time constant is an integer number. To calculate the time constant, use the formula:

Time constant = (clock frequency/(2 * baud rate)) – 2

= (0.5 * (clock frequency)/baud rate)) – 2

For a clock frequency of 2.4756 megahertz, this formula simplifies to:

Time constant = (1228880/baud rate) – 2

If the resulting number is not an integer, round off to the nearest integer. The actual baud rate (assuming a perfectly accurate clock) is given by:

Baud rate (actual) = 1228800/(time constant integer + 2)

Table 4-22 shows the decimal divisor values and their hexadecimal equivalents that are loaded into the time constant registers to generate standard baud rates.

NOTE: Bit assignments for the Z8530 read and write registers are provided for reference only. Refer to the Zilog *Z8030/Z8530 SCC Serial Communications Controller Technical Manual* for programming data.

Read Register 0 Read register 0 contains the status of the receive and transmit data buffers and also the status bits for the six sources of the external/status interrupts. Figure 4-34 shows the bit assignments for read register 0.

Read Register 1 Read register 1 contains the special receive condition status bits and residue codes. Figure 4-35 shows the bit assignments for read register 1.

Table 4-22 **Baud Rate Generator Divisors**

Baud Rate	*Divisor*	*Percent Error*
50	24 574 (5FFE)	0
75	16 382 (3FFE)	0
110	11 169 (2BA1)	0.001
134.5	9 134 (23AE)	0.001
150	8 190 (1FFE)	0
200	6 142 (17FE)	0
300	4 094 (FFE)	0
600	2 046 (7FE)	0
1 200	1 022 (3FE)	0
1 800	681 (2A9)	0.049
2 400	510 (1FE)	0
3 600	339 (153)	0.098
4 800	254 (FE)	0
7 200	169 (A9)	0.196
9 600	126 (7E)	0
19 200	62 (3E)	0

Figure 4-34 Read Register 0 Bit Assignments

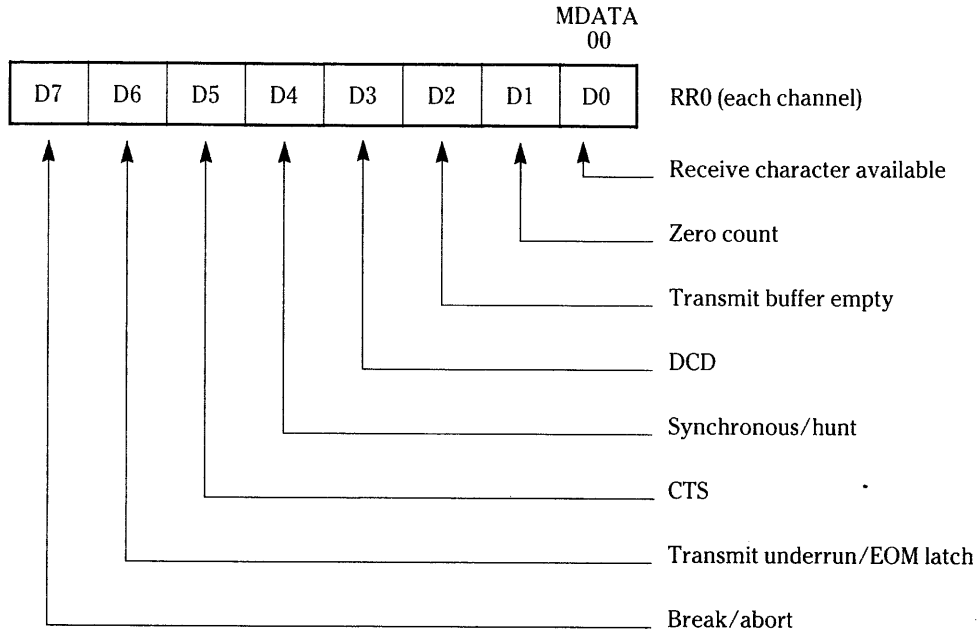
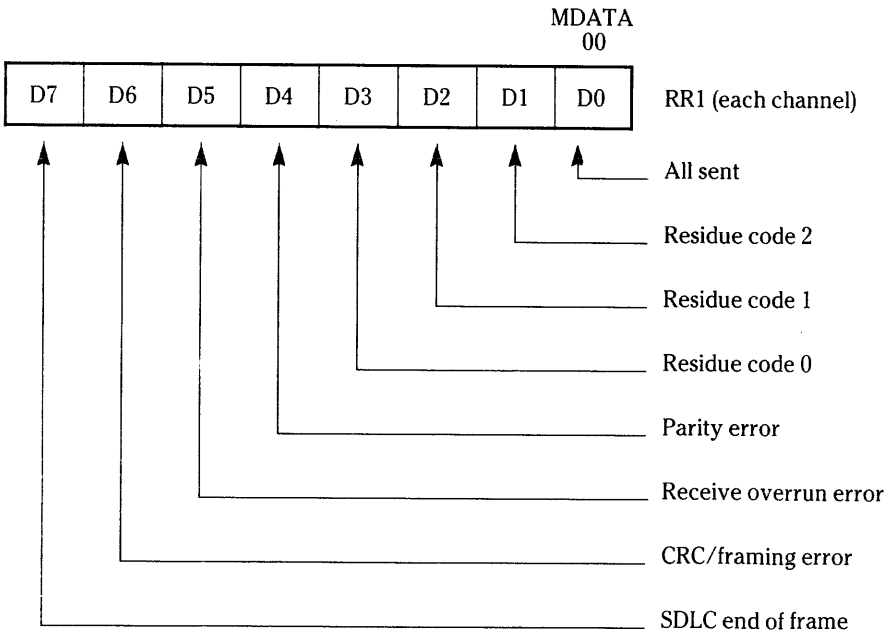


Figure 4-35 Read Register 1 Bit Assignments



Read Register 2 Read register 2 contains the interrupt vector written in write register 2. When the register is accessed in channel A, the vector returned is the vector actually stored in write register 2. This feature is not used in the SI board configuration. When read register 2 in channel B is accessed, the vector returned includes status information in bits 1, 2, and 3 or in bits 6, 5, and 4, depending on the state of the status high/status low bit in write register 9. Write register 9 is shared by both channels. Figure 4-36 and Figure 4-37 show channel A and channel B bit assignments for read register 2.

Read Register 3 Read register 3 is the interrupt pending register. The status of each of the interrupt pending bits in the RS-232C serial data port is reported. The register exists only in channel A; an access to this register in channel B will return all zeros. Bit assignments for read register 3 are shown in Figure 4-38.

Read Register 10 Read register 10 contains miscellaneous status bits. Unused bits are always zero. Bit assignments for read register 10 are shown in Figure 4-39.

Read Register 12 Read register 12 returns the value stored in write register 12, which is the lower byte of the time constant for the baud rate generator. Figure 4-40 shows the bit assignments for read register 12.

Figure 4-36 **Read Register 2 Bit Assignments — Channel A Only**

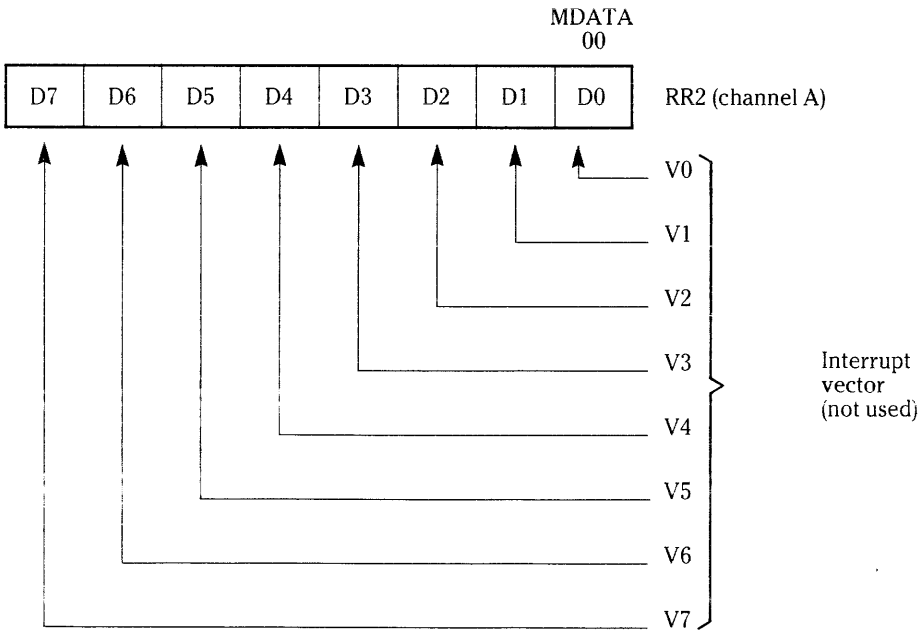
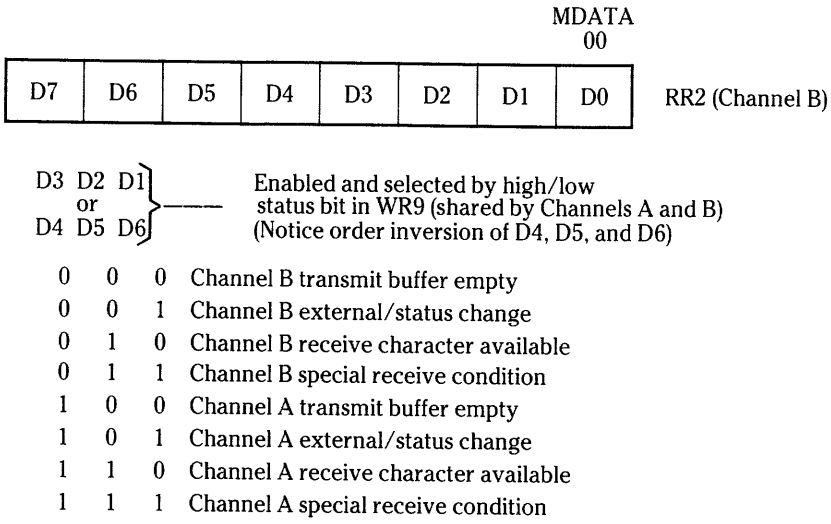


Figure 4-37 **Read Register 2 Bit Assignments — Channel B Only**



Read Register 13 Read register 13 returns the value stored in write register 13, which is the upper byte of the time constant for the baud rate generator. Figure 4-41 shows the bit assignments for read register 13.

Read Register 15 Read register 15 reflects the value stored in write register 15, the external/status interrupt register. Figure 4-42 shows the bit assignments for read register 15. The two unused bits are always returned as zeros.

Figure 4-38 **Read Register 3 Bit Assignments**

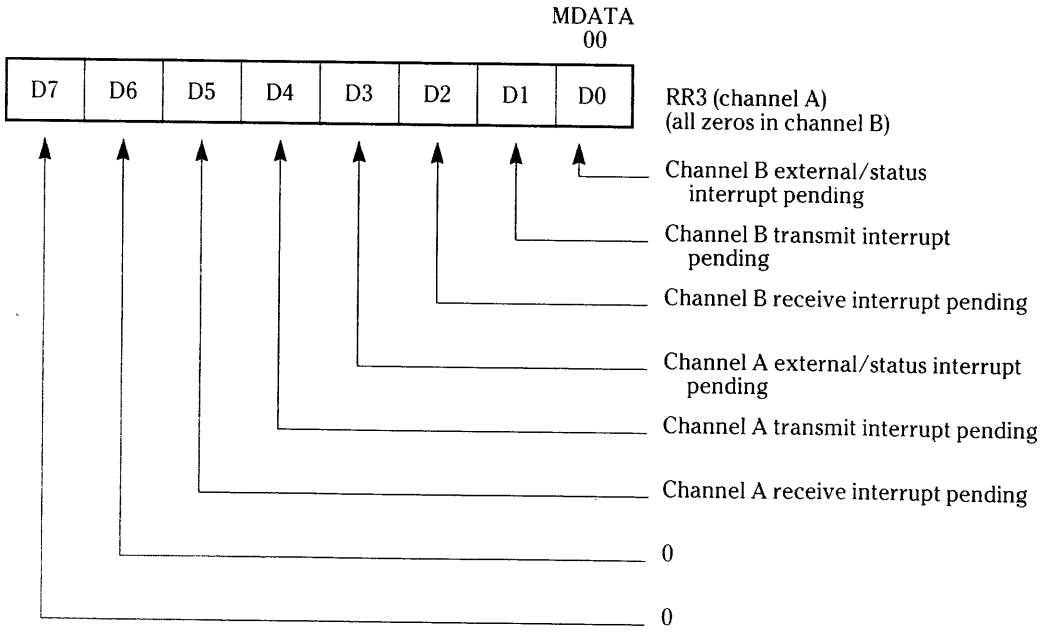


Figure 4-39 Read Register 10 Bit Assignments

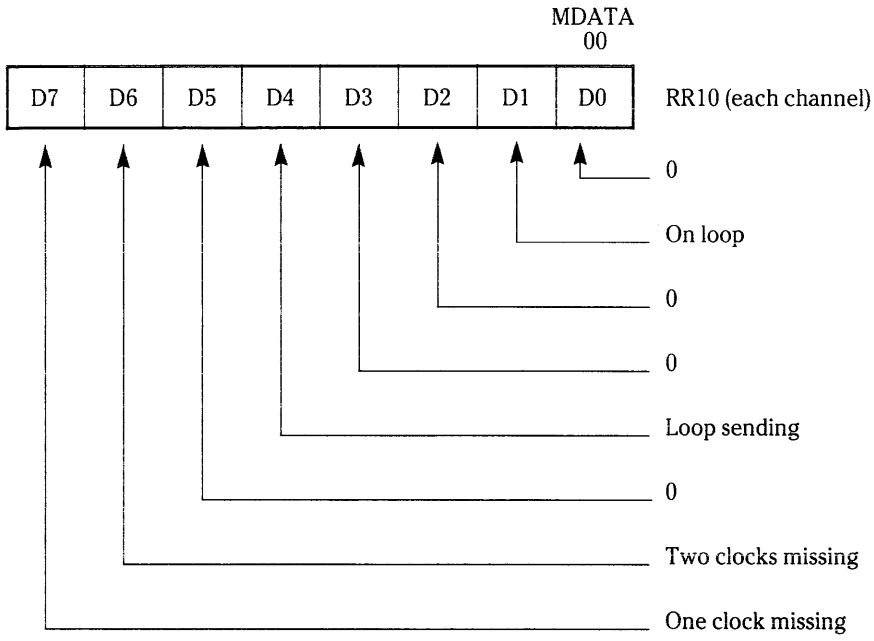


Figure 4-40 Read Register 12 Bit Assignments

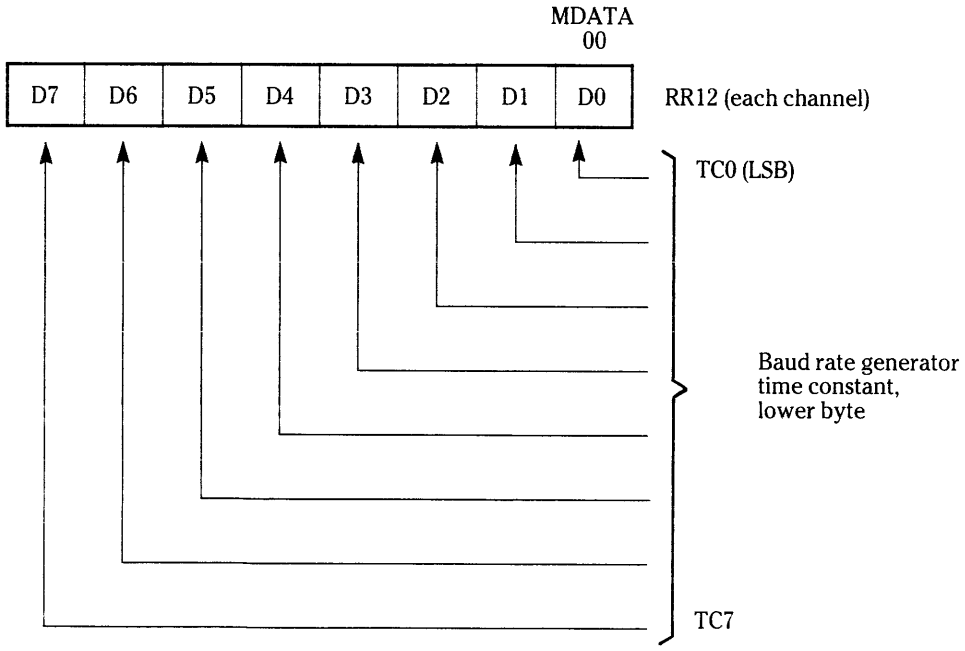


Figure 4-41 Read Register 13 Bit Assignments

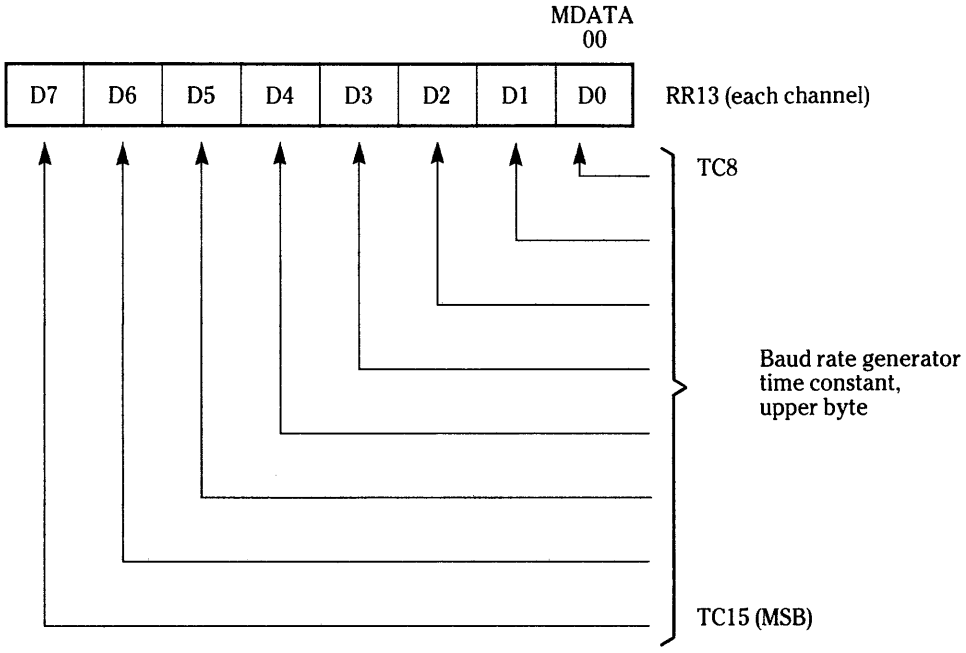
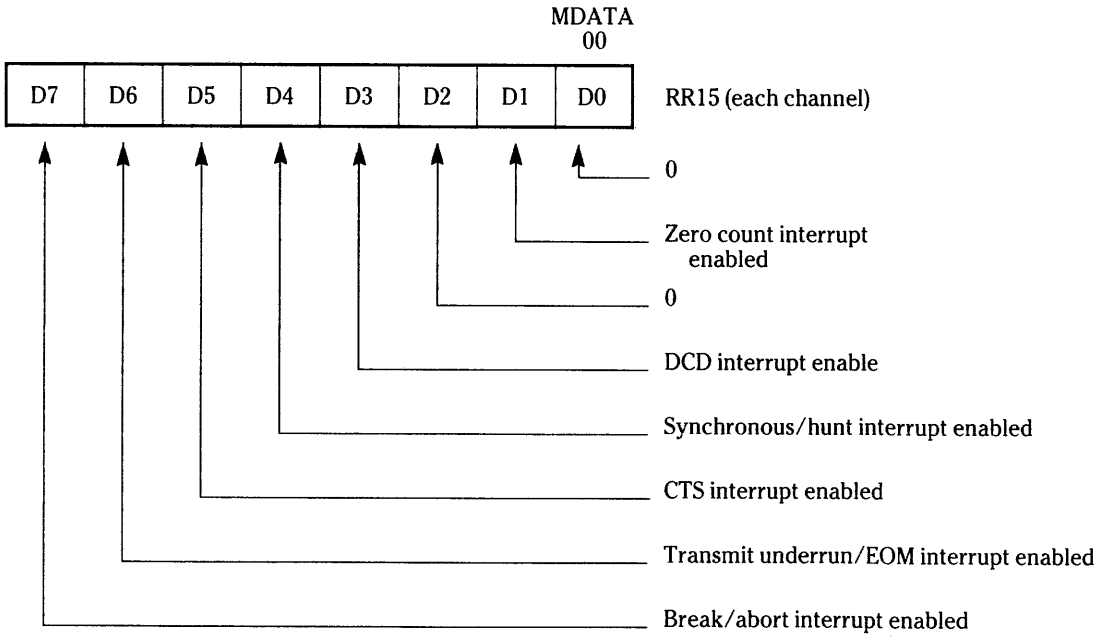


Figure 4-42 Read Register 15 Bit Assignments



Write Register 0 Write register 0 is the command register and the CRC reset code register. Figure 4-43 shows bit assignments for write register 0 and includes register select bits in addition to the command and reset codes.

Write Register 1 Write register 1 is the control register for the various interrupt and wait/request modes. Figure 4-44 shows the bit assignments for write register 1.

Write Register 2 Write register 2 is the interrupt vector register. Only one vector register exists in the Z8530, but it can be accessed from either channel. The interrupt vector can be modified by status information. This is controlled by the vector includes status (VIS) and the status high/status low bits of write register 9. This register is not used in the SI board RS-232C serial data port configuration. Bit assignments for write register 2 are shown in Figure 4-45.

Write Register 3 Write register 3 contains the control bits and parameters for the receive logic as shown in Figure 4-46.

Write Register 4 Write register 4 contains the control bits for both the receiver and the transmitter. These bits should be set in the transmit and receive initialization stage before issuing the contents of write registers 1, 3, 6, and 7. Bit assignments for write register 4 are shown in Figure 4-47.

Figure 4-43 **Write Register 0 Bit Assignments**

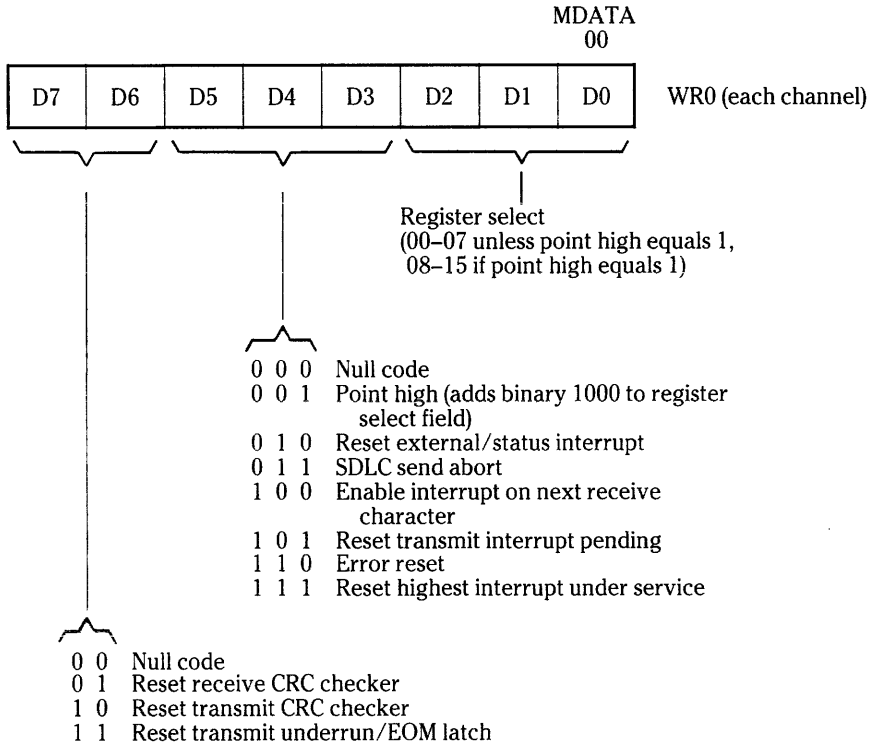


Figure 4-44 Write Register 1 Bit Assignments

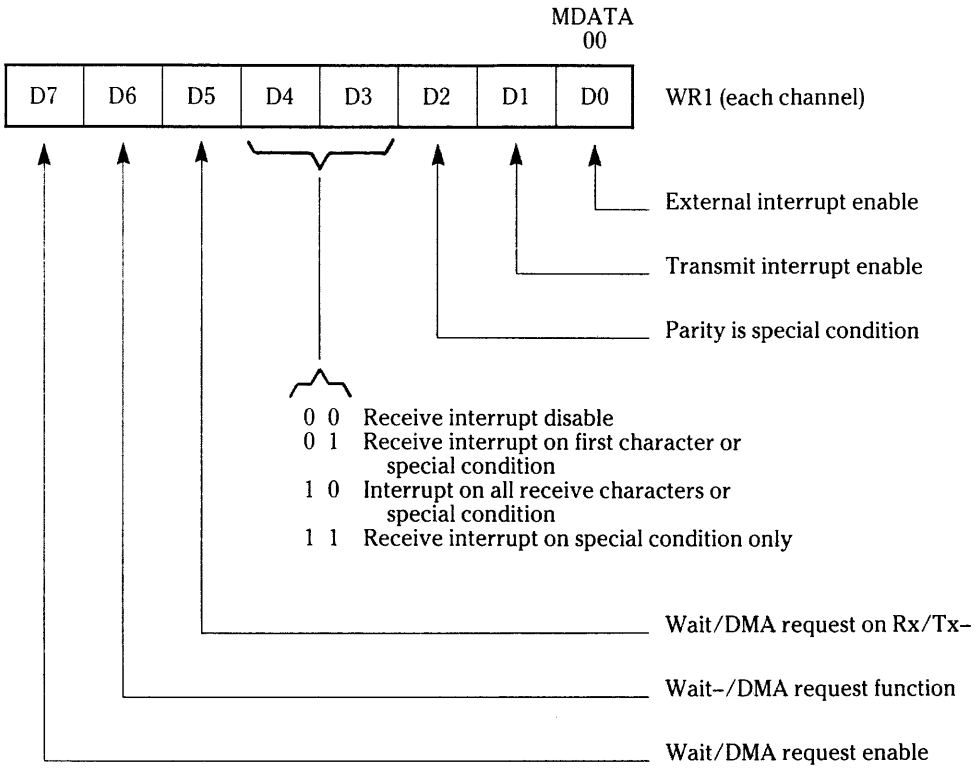


Figure 4-45 Write Register 2 Bit Assignments

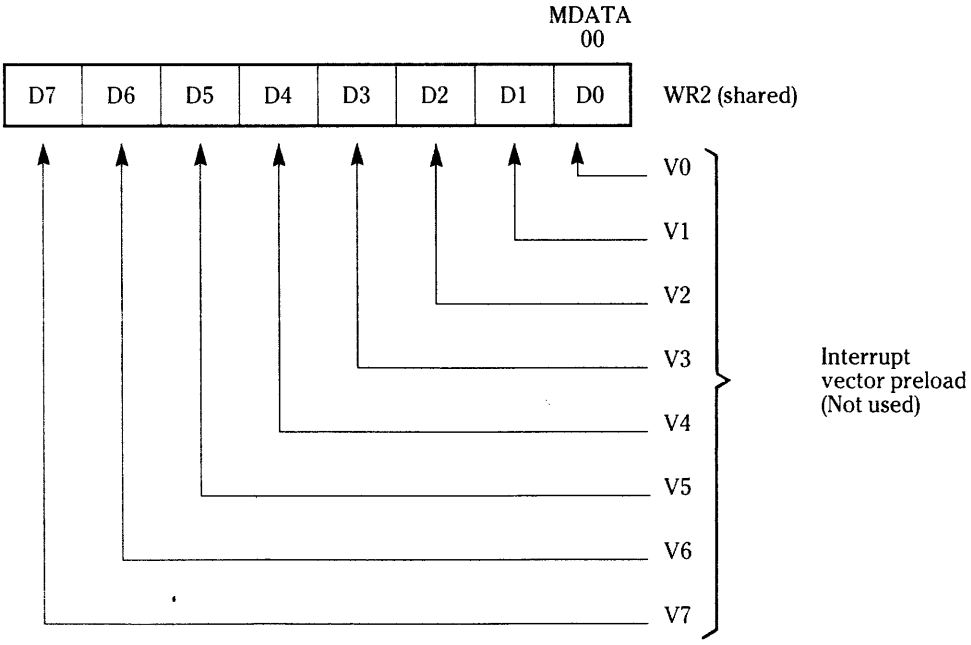


Figure 4-46 Write Register 3 Bit Assignments

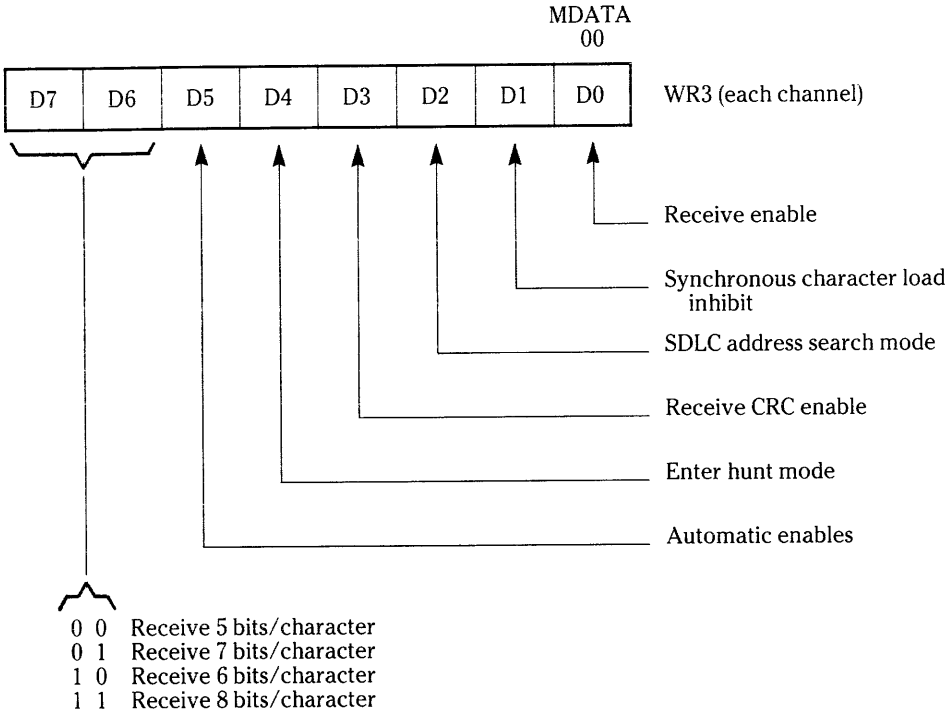
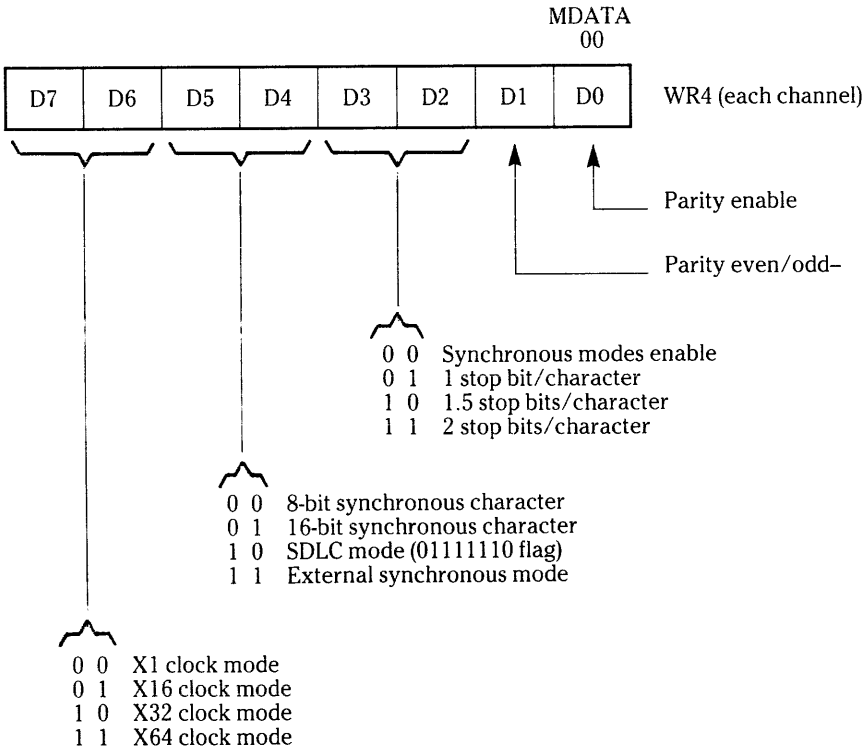


Figure 4-47 Write Register 4 Bit Assignments



Write Register 5 Write register 5 contains control bits that affect the operation of the transmitter. Bit 2 selects the CRC polynomial used by both the transmitter and the receiver. Figure 4-48 shows bit assignments for write register 5.

Write Register 6 Write register 6 is programmed to contain the transmit synchronous character when in the monosynchronous mode, and either the first byte of a 16-bit synchronous character or the low order 4 bits of a 12-bit synchronous character when in the bisynchronous mode. When in the SDLC modes, the register is programmed to contain the secondary address field to compare against the address field of the SDLC frame. Write register 6 is not used in asynchronous modes. Bit assignments for write register 6 are shown in Figure 4-49.

Write Register 7 Write register 7 is programmed to contain the receive synchronous character in the monosynchronous mode; the upper order 8 bits of either a 12-bit or 16-bit synchronous character in the bisynchronous mode; or a flag character in the SDLC modes. Bit assignments for write register 7 are shown in Figure 4-50.

Write Register 9 Write register 9 is the master interrupt control register and contains the reset command bits. Only one register exists in the Z8530 and it can be accessed from either channel. Bit assignments for write register 9 are shown in Figure 4-51.

Write Register 10 Write register 10 contains miscellaneous control bits for both the transmitter and the receiver. Bit assignments for write register 10 are shown in Figure 4-52.

Figure 4-48 **Write Register 5 Bit Assignments**

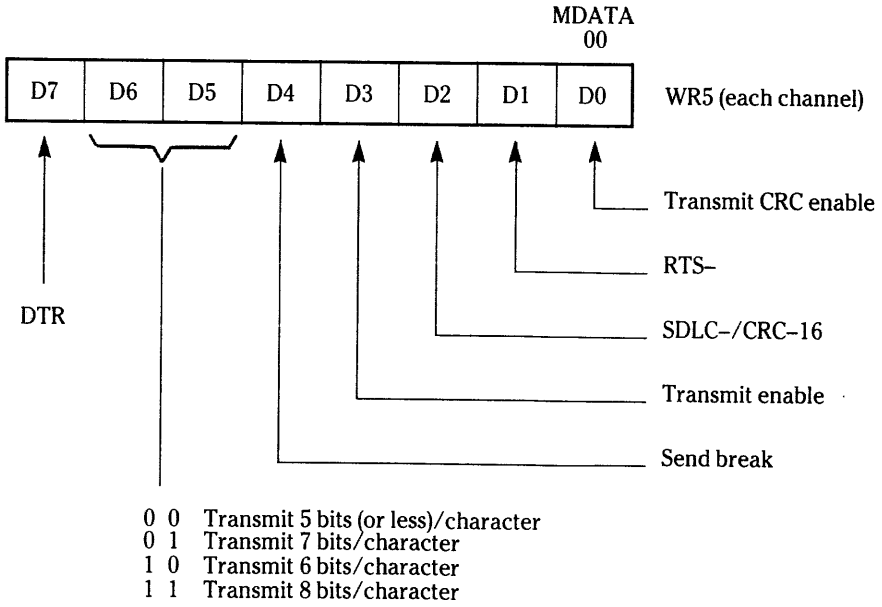


Figure 4-49 **Write Register 6 Bit Assignments**

MDATA 00

D7	D6	D5	D4	D3	D2	D1	D0	WR6 (each channel)
S7	S6	S5	S4	S3	S2	S1	S0	Monosynchronous, 8 bits
S1	S0	S5	S4	S3	S2	S1	S0	Monosynchronous, 6 bits
S7	S6	S5	S4	S3	S2	S1	S0	Bisynchronous, 16 bits
S3	S2	S1	S0	1	1	1	1	Bisynchronous, 12 bits
A7	A6	A5	A4	A3	A2	A1	A0	SDLC
A7	A6	A5	A4	x	x	x	x	SDLC address range

Figure 4-50 **Write Register 7 Bit Assignments**

MDATA 00

D7	D6	D5	D4	D3	D2	D1	D0	WR7 (each channel)
S7	S6	S5	S4	S3	S2	S1	S0	Monosynchronous, 8 bits
S5	S4	S3	S2	S1	S0	x	x	Monosynchronous, 6 bits
S15	S14	S13	S12	S11	S10	S9	S8	Bisynchronous, 16 bits
S11	S10	S9	S8	S7	S6	S5	S4	Bisynchronous, 12 bits
0	1	1	1	1	1	1	0	SDLC

Figure 4-51 Write Register 9 Bit Assignments

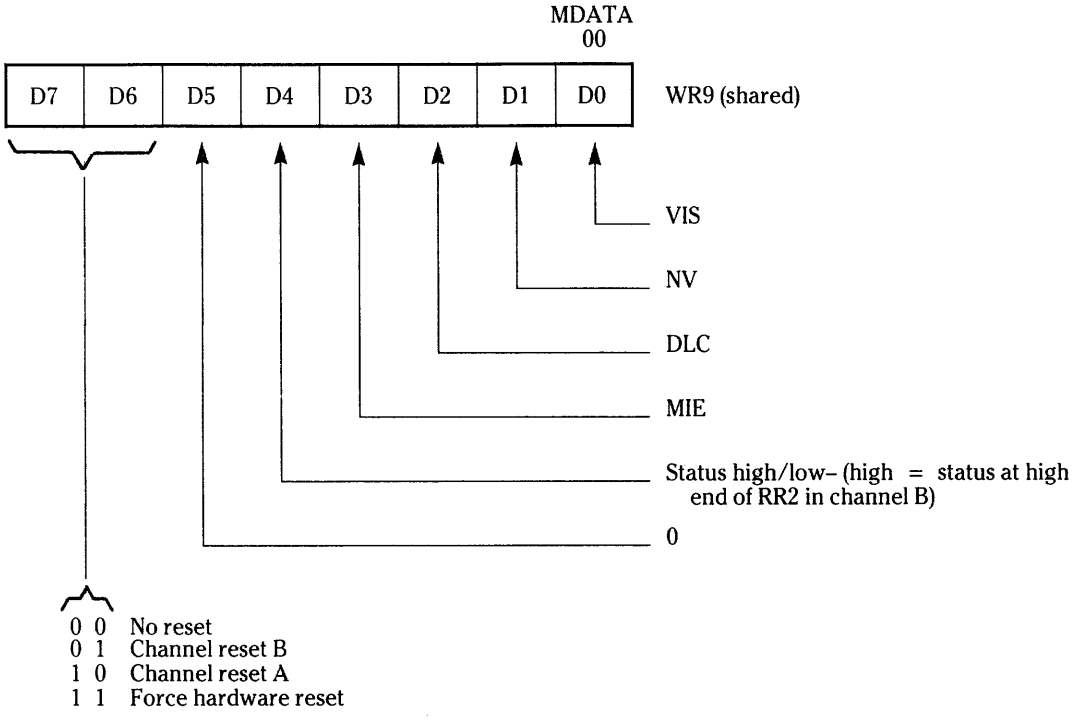
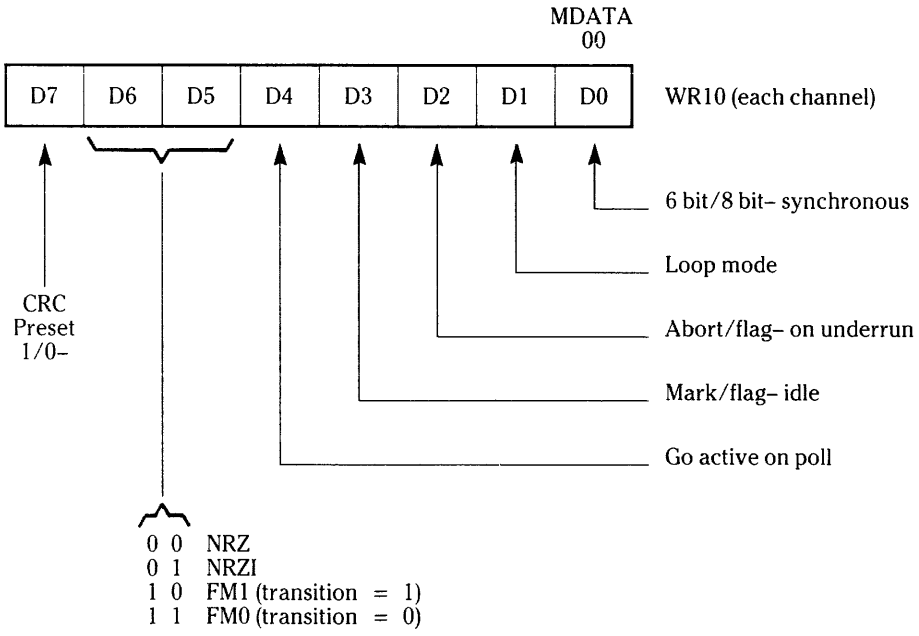


Figure 4-52 Write Register 10 Bit Assignments



Write Register 11 Write register 11 is the clock mode control register. The bits in this register control the sources for both the transmit and receive clocks. Bit assignments for write register 11 are shown in Figure 4-53.

Write Register 12 Write register 12 contains the lower byte of the time constant for the baud rate generator. Figure 4-54 shows bit assignments for write register 12.

Write Register 13 Write register 13 contains the upper byte of the time constant for the baud rate generator. Figure 4-55 shows the bit assignments for write register 13.

Write Register 14 Write control register 14 contains miscellaneous control bits. Bit assignments for write register 14 are shown in Figure 4-56.

Write Register 15 Write register 15 is the external/status source control register. If the external/status interrupts are enabled as a group by way of write register 1, bits in this register control which external/status conditions can cause an interrupt. Figure 4-57 shows bits assignments for write register 15.

Figure 4-53 Write Register 11 Bit Assignments

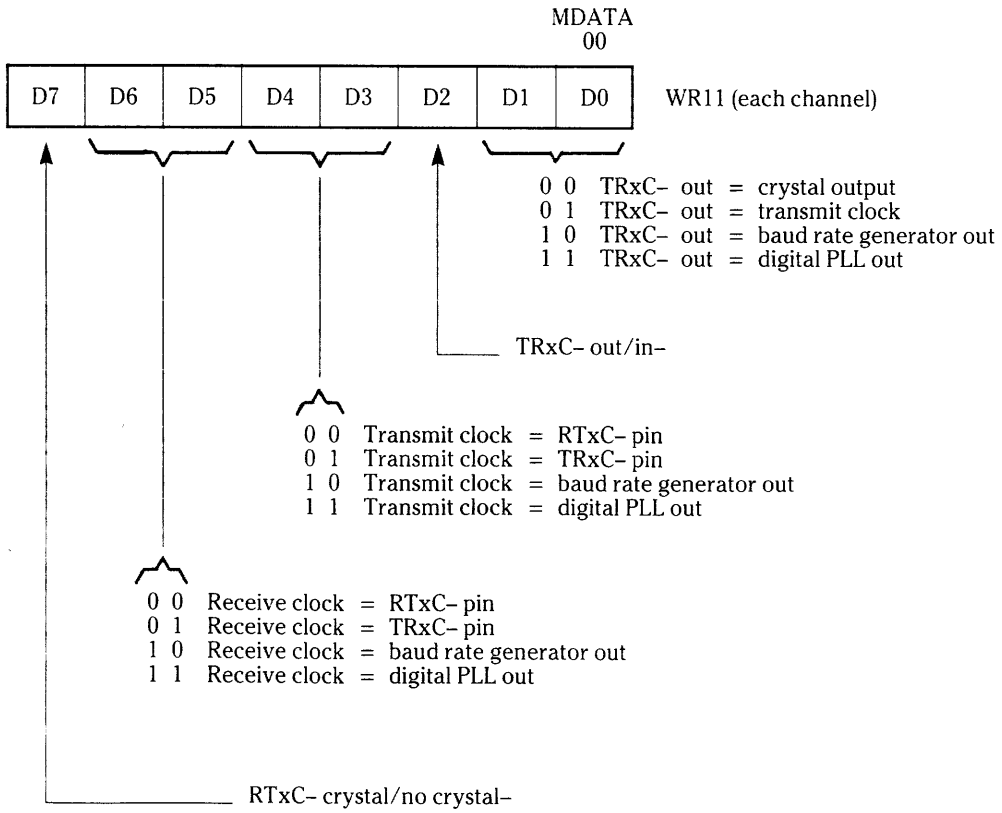


Figure 4-54 Write Register 12 Bit Assignments

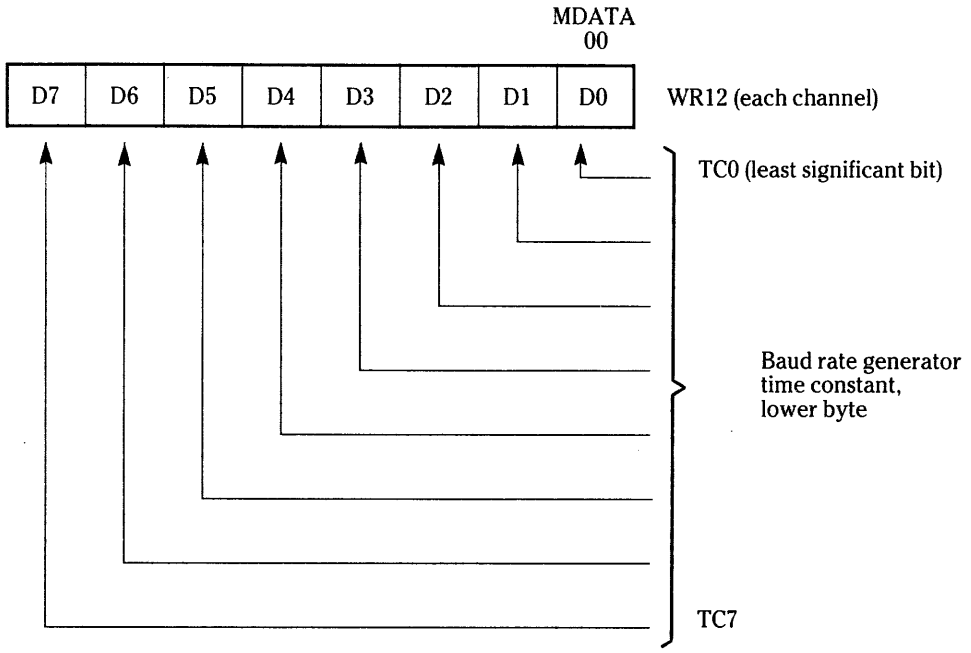


Figure 4-55 Write Register 13 Bit Assignments

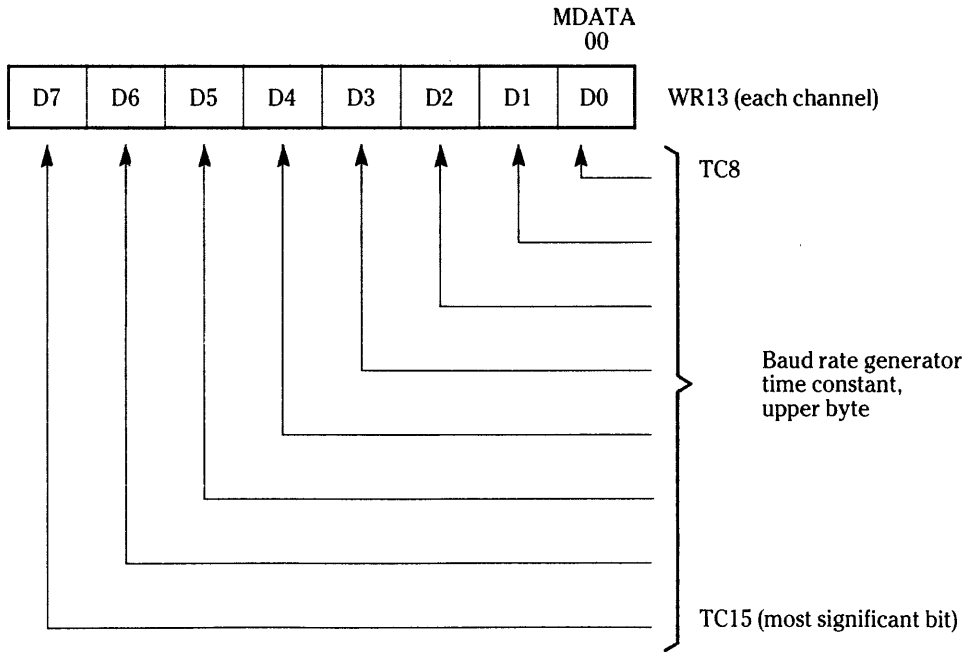


Figure 4-56 Write Register 14 Bit Assignments

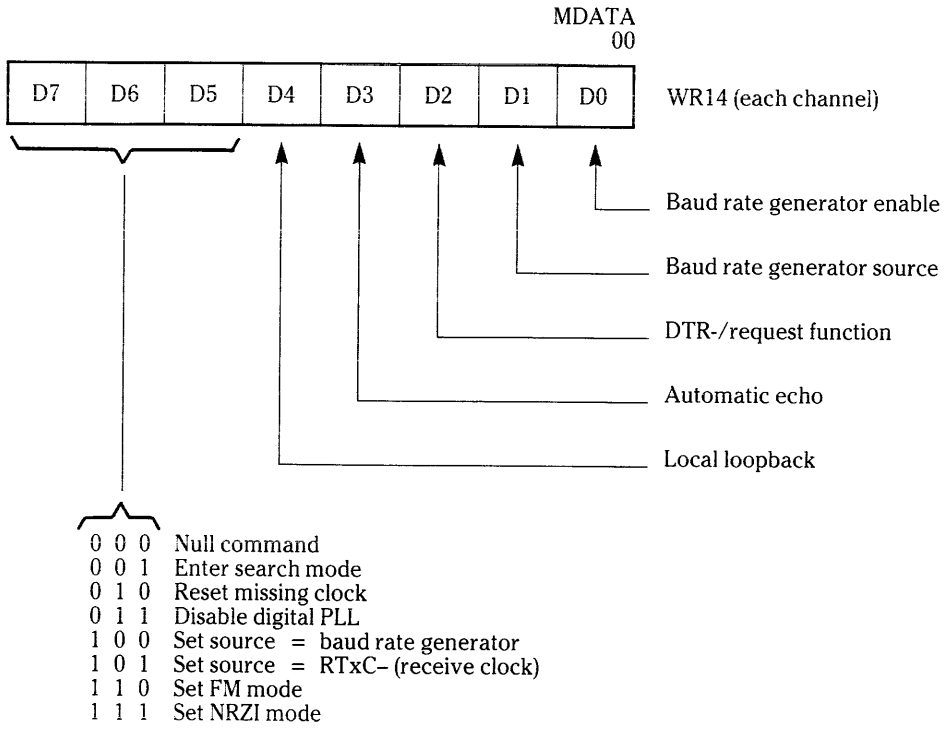
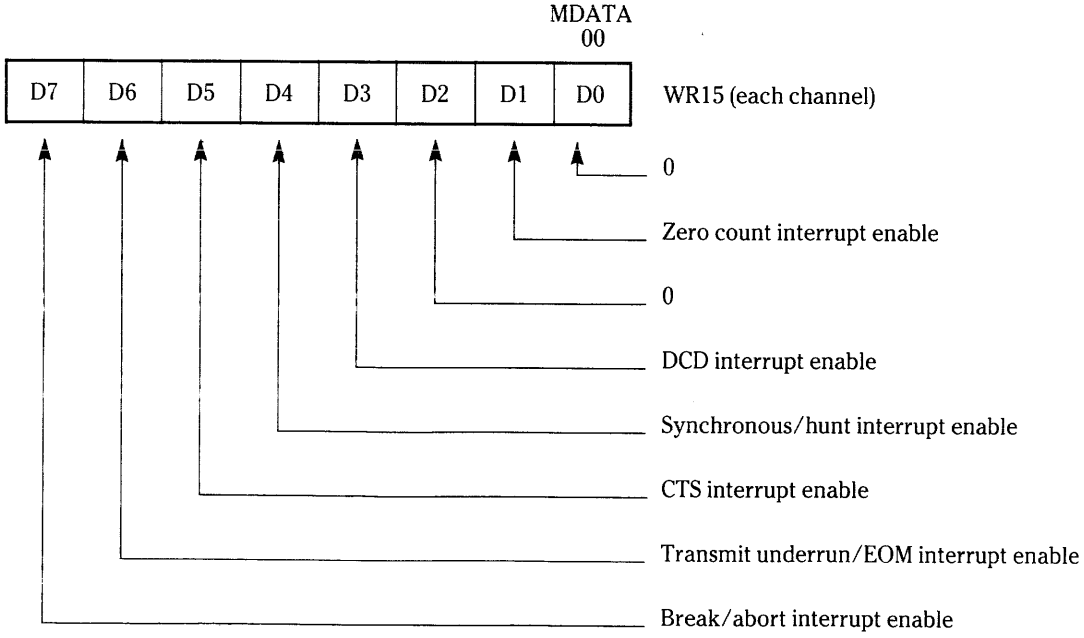


Figure 4-57 Write Register 15 Bit Assignments



Typical Programming Example

4.4.15.3 The following steps indicate the necessary procedure required to program the RS232-C port for asynchronous communications at 4800 baud, 1 stop bit, internal transmit and receive interrupts enabled so that the chip can be polled, external interrupts disabled, and DTR, RTS, and transmit data enabled. Refer to the *Z8030/Z8530 SCC Serial Communications Controller Technical Manual* for detailed information. In the following procedural steps, > is used to indicate hexadecimal values.

1. Write > C0 to WR9 to reset the Z8530
2. Write > 04 to WR4 to set the chip for asynchronous communications using 1 stop bit
3. Write > 12 to WR1 to enable internal interrupts
4. Write > 50 to WR11 to enable the internal baud rate clock
5. Write > FE to WR12 to specify 4800 baud (low byte)
6. Write > 00 to WR13 to specify 4800 baud (high byte)
7. Write > 63 to WR14 to disable sync comm and enable baud generator
8. Write > 00 to WR15 to disable external interrupts
9. Write > C1 to WR3 to enable the receiver at 8 bits
10. Write > EA to WR5 to enable modem control bits

After the Z8530 has been initialized, read register RR2 can be polled to determine the status of the transmit and receive buffer. Characters can be sent or received by read and write operations to the receive and transmit data buffers.

Board Initialization Procedure

4.4.16 Since the SI board is a multifunction board that handles many different types of inputs from several sources, the board initialization sequence must be carefully followed to ensure that a board reset does not destroy a previously initialized function. Perform the following procedure in the order shown to properly initialize the board from power-up or after a full system reset.

1. Initialize the real-time clock as described in paragraph 4.4.7. Disable and clear any pending interrupts.
2. Initialize the interval timers as described in paragraph 4.4.8.
3. Perform a software reset by writing a 1 to bit 0 of the configuration register as described in paragraph 4.4.4.
4. Initialize the CRT controller as described in paragraph 4.4.10.7.

5. Write the event generator vector addresses into the register file for each of the desired events. As an absolute minimum, the power failure warning, the overtemperature, and optic link interrupt addresses must be written early in order to protect system integrity during the power-up cycle.

6. Enable the event generator by writing to the configuration register as described in paragraph 4.4.4.

7. Run any desired power-up testing and reset the fault lamps by writing to their configuration register locations as described in paragraph 4.4.4.

8. Clear the bit-mapped memory and enable the monitor display by writing to the video attributes register as described in paragraph 4.4.10.7.

9. Enable keyboard operation by initializing the keyboard USART as described in paragraph 4.4.12.2.

   The SI board is now fully operational. Note that once the CRT controller has been initialized, a software reset will clear the controller registers and halt operation of the fiber-optic link to the monitor. The CRT controller must be reinitialized to resume fiber-optic link operation.

Fiber-Optic Board Functional Description

4.5 The following paragraphs contain an operational and physical description of the fiber-optic board.

General Description

4.5.1 The fiber-optic board is electrically and physically very simple. The fiber-optic board provides the interface from the SI board to the video display and acts as a passive adapter to extend the RS-232C and parallel printer port signals of the SI board to corresponding I/O cable connectors. There are four connectors on the fiber-optic board. Connector P1 is a 96-pin DIN female connector that mates with an extended-pin DIN connector on the backplane at the connector P3 position of the SI board. Through wiring on the fiber-optic board, the RS-232C signals at connector P1 of the fiber-optic board are transferred to 18-pin connector P3 on the board. Similarly, the parallel printer signals at connector P1 are brought to 40-pin connector P4 of the board. Logic on the fiber-optic board forms a part of a duplex fiber-optic communication link between the computer system at connector P1 and the monitor enclosure at connector P2. Both interfaces are described in the following paragraphs.

RS-232C Interface

4.5.2 The functions of each of the EIA input and output signals that are generated or received by the RS-232C port are described in Table 4-17. Signals and pin numbers for connector P3 and at the connecting cable are shown in Figure 4-58.

NOTE: Pin numbers on the connecting cables to connectors P3 and P4 do not correspond to those of the connectors. The connector orientation in the system enclosure must allow the cable to dress downward. To allow this, the cable connector must be oriented 180 degrees relative to the fiber-optic board connectors.

This numbering discontinuity is not visible when the connector shield is soldered into place. However, the schematic shows the reversed pin numbering.

Parallel Printer Interface

4.5.3 The functions of each of the input and output signals that are generated or received by the parallel printer port are described in Table 4-9. Signals and pin numbers for connector P4 and at the connecting cable are shown in Figure 4-59.

Fiber-Optic Logic and Interface

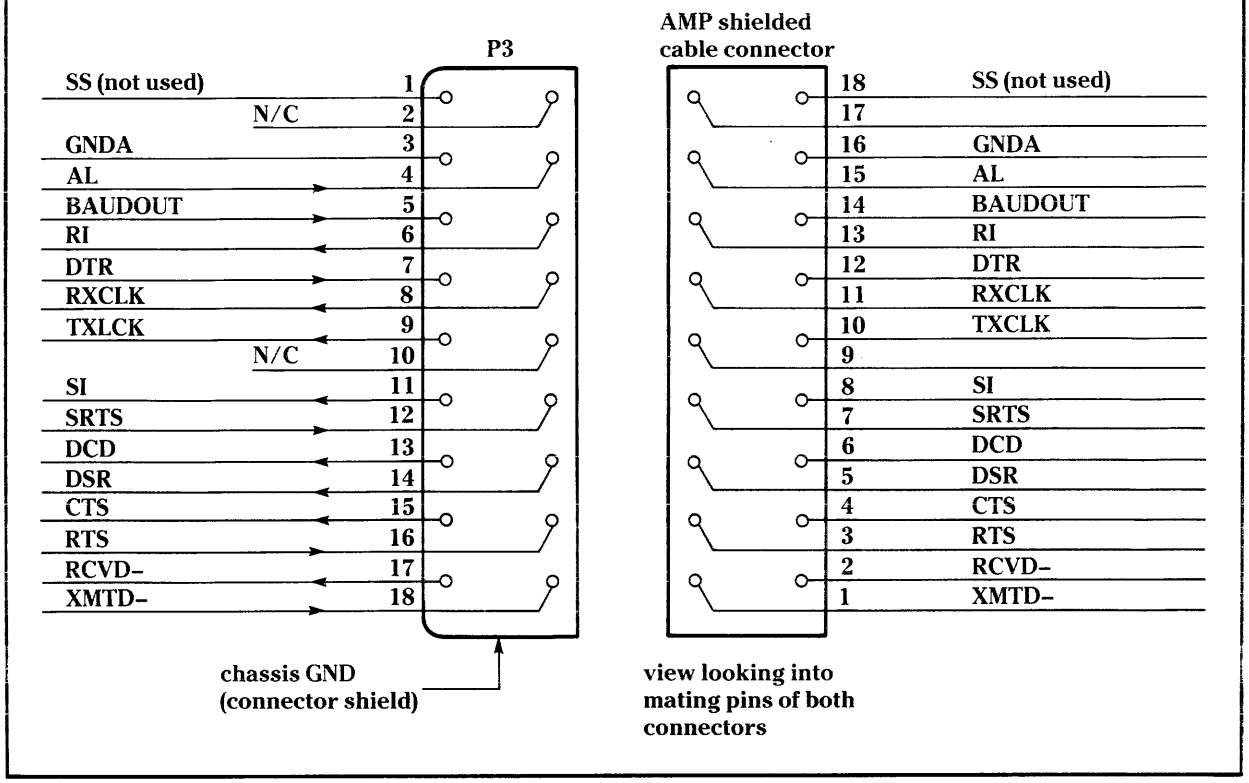
4.5.4 Electronic circuitry on the fiber-optic board converts the transistor-transistor logic (TTL) signals at the output of the SI board to an optical signal for transmission to the monitor via channel A of the connecting fiber-optic cable. Circuitry also converts the optical signal received from the monitor via channel B of the cable to the TTL signal level required for application to the SI board.

The transmit channel TTL-to-optical converter and optical transmitter consists of two buffer/drivers and a light-emitting diode (LED) optic emitter. The LED is mounted within the optical I/O connector to launch optical pulses into the channel A (transmit) fiber of the fiber-optic cable. The transmitting channel has a high bandwidth, specified at a signaling rate of 34 megahertz.

The receive channel optical receiver and optical-to-TTL converter consist of an integrated detector/preamplifier device mounted within the optical I/O connector and a voltage comparator. The received data is TTL-compatible at the output of the voltage comparator and is applied to the SI board.

A three-terminal regulator circuit on the fiber-optic board develops a clean +5-volt supply from a +12-volt input from the SI board for use of the logic on the fiber-optic board.

Figure 4-58 Connector P3 and Connecting Cable Pin Assignments



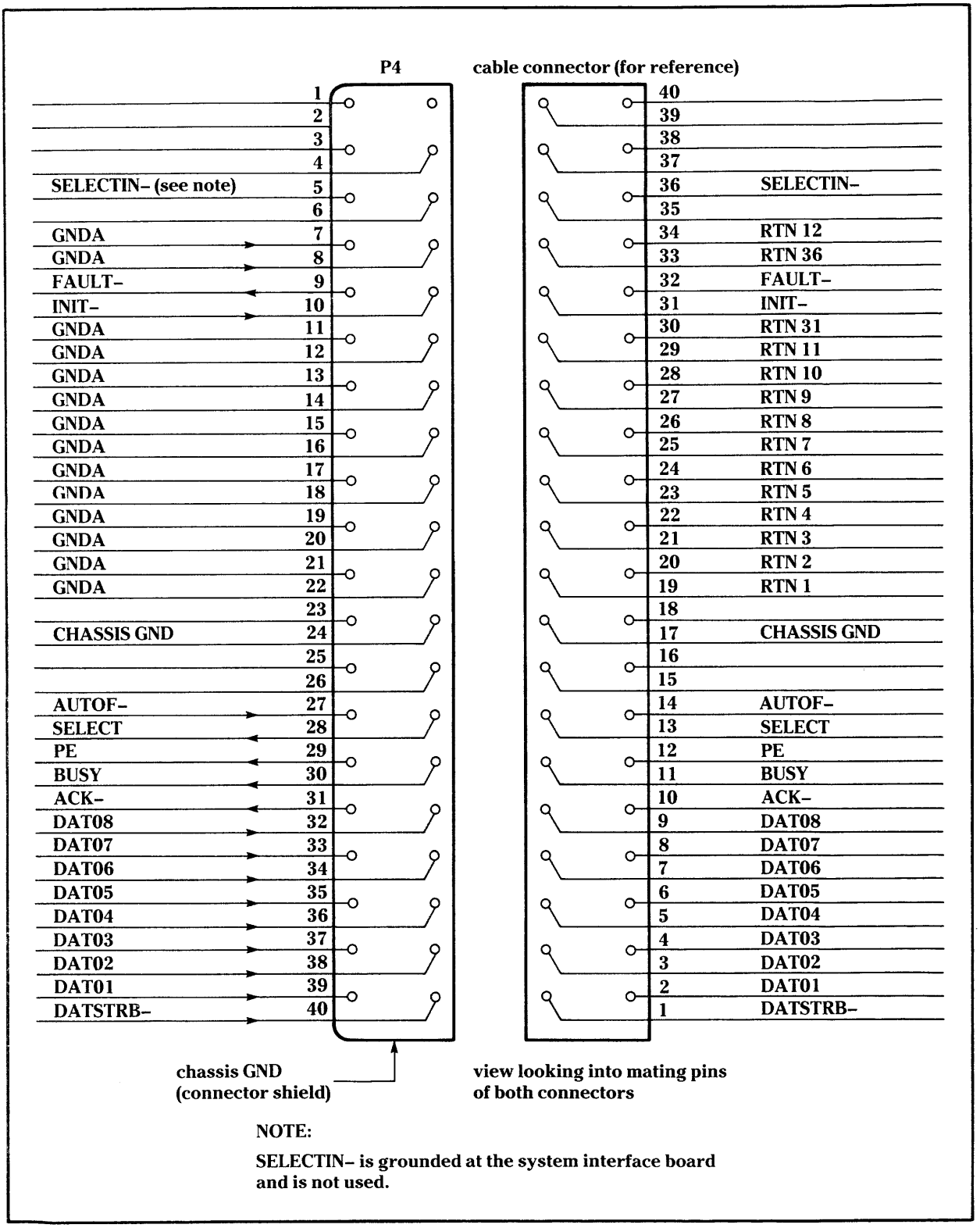
TTL Signal-Level Interface

4.5.4.1 The TTL signals at the interface between the SI board (connector P3) and the fiber-optic board (connector P1) are described in Table 4-23.

Optical Interfaces

4.5.4.2 The optical power-out and power-in interfaces occur at the optical connector. Both the transmitting optic emitter LED and the receiving integrated detector/preamplifier are mounted within the optical connector. As described in Table 1-2, the optical power-out signal is a high data rate optical output to channel A of the fiber-optic cable, and the optical power-in signal from channel B of the fiber-optic cable is at a lower data rate than channel A.

Figure 4-59 Connector P4 and Connecting Cable Pin Assignments



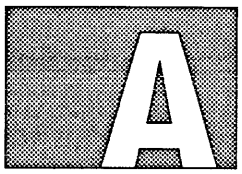
NOTE:

SELECTIN– is grounded at the system interface board and is not used.

Table 4-23 **SI Board and Backplane Signals**

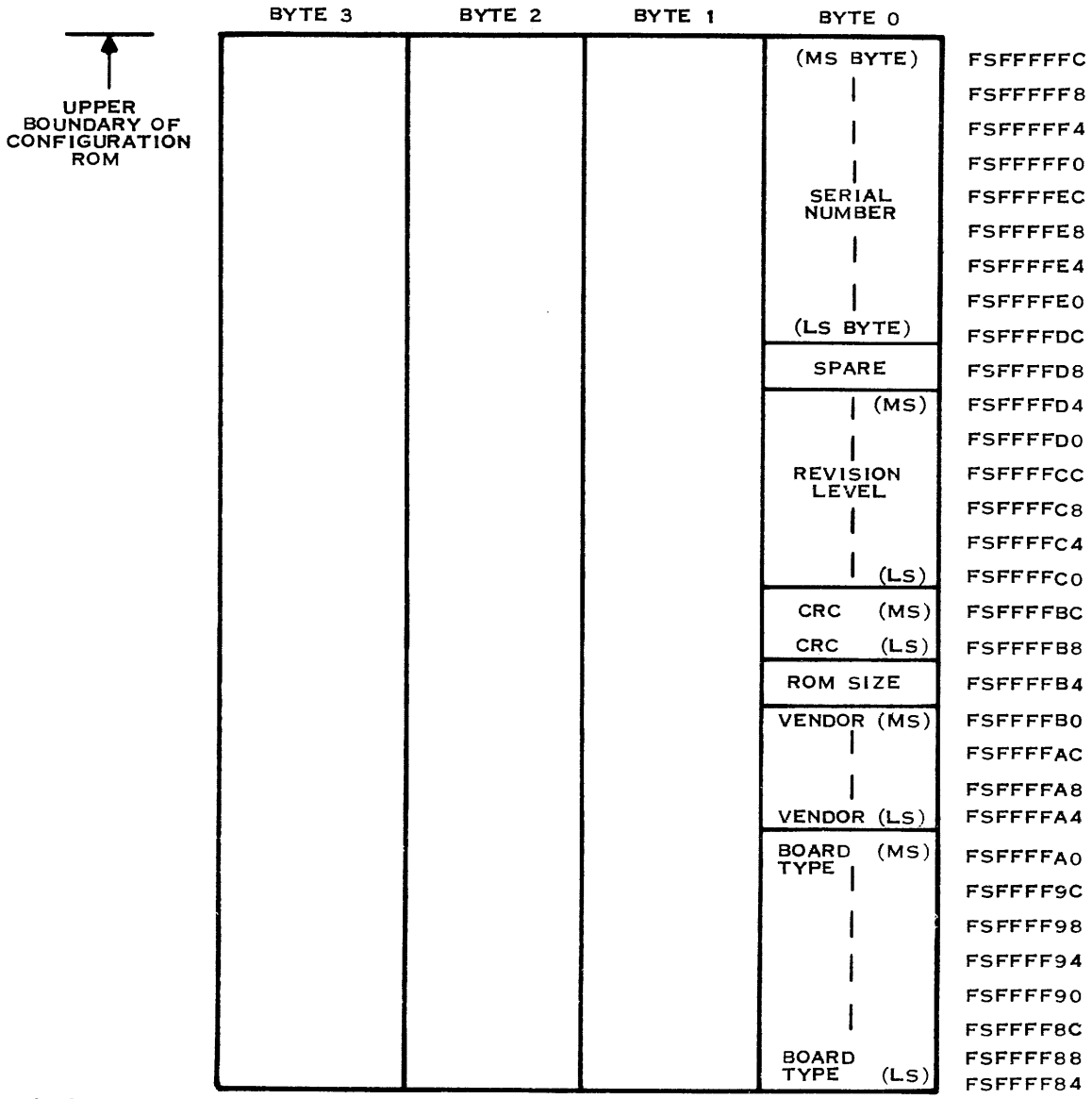
Signal Name	*Function*
+5 Vdc	Source of +5 volts for the TTL devices.
+12 Vdc	Source to develop clean +5-volt operating power for low-level optic-to-TTL conversion circuits.
GND, GNDA	Power/signal returns. GND and GNDA are kept separate on the fiber-optic board and connected together at the backplane.
AXMIT	Channel A transmit data.
BCVDAT	Channel B receive data.
BRCVRTN	Channel B receive data return.

MEMORY MAPS



This appendix contains detailed memory maps of major devices on the SI board. The content and significance for the base address and additional memory locations for each device are described in detail in Section 4 and are repeated here for easy reference.

Figure A-1 Configuration ROM Memory Map (Sheet 1 of 3)



NOTES:

MS = MOST SIGNIFICANT.

LS = LEAST SIGNIFICANT.

Figure A-1 Configuration ROM Memory Map (Sheet 2 of 3)

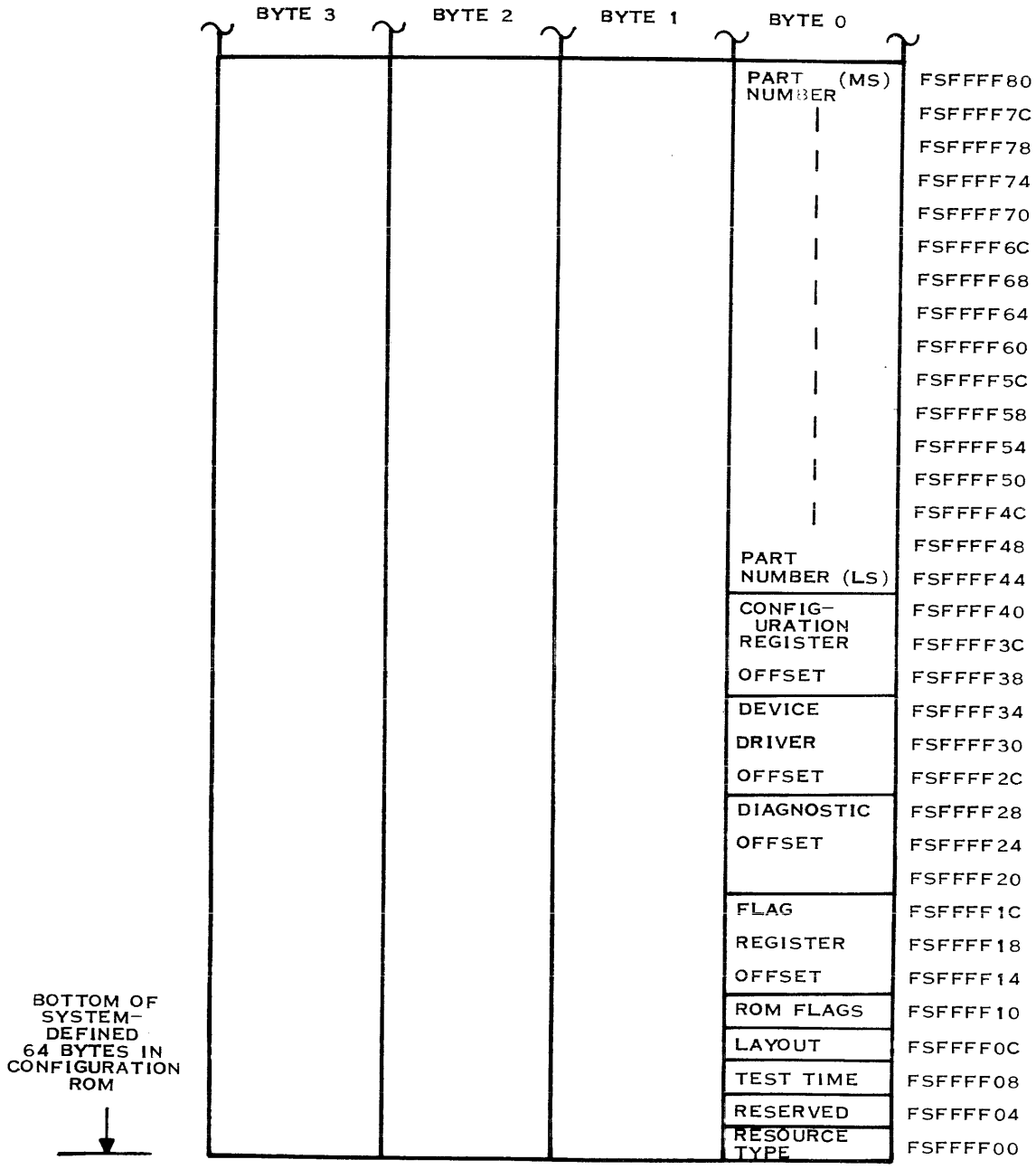


Figure A-1 Configuration ROM Memory Map (Sheet 3 of 3)

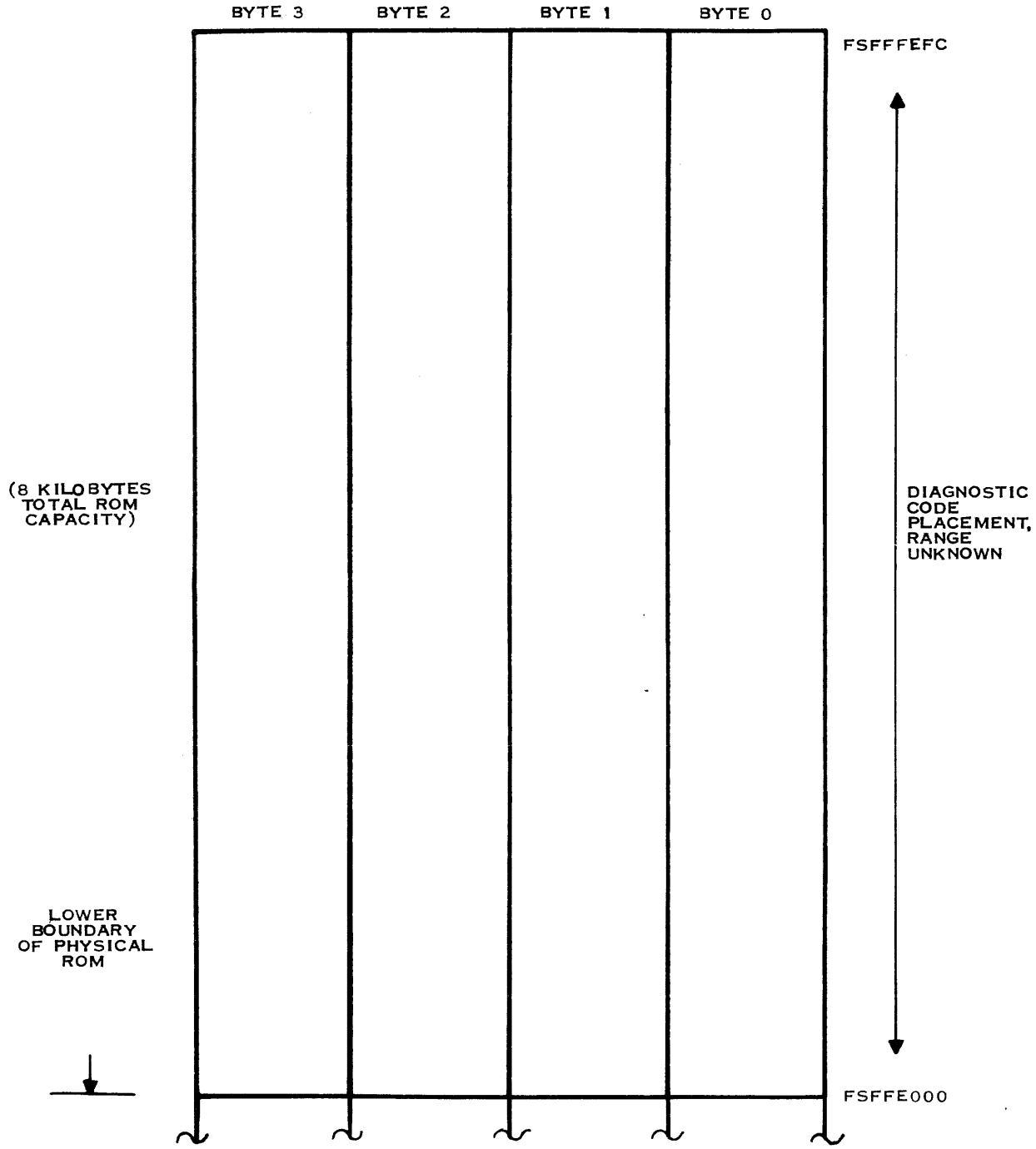


Figure A-2 Keyboard USART Memory Map

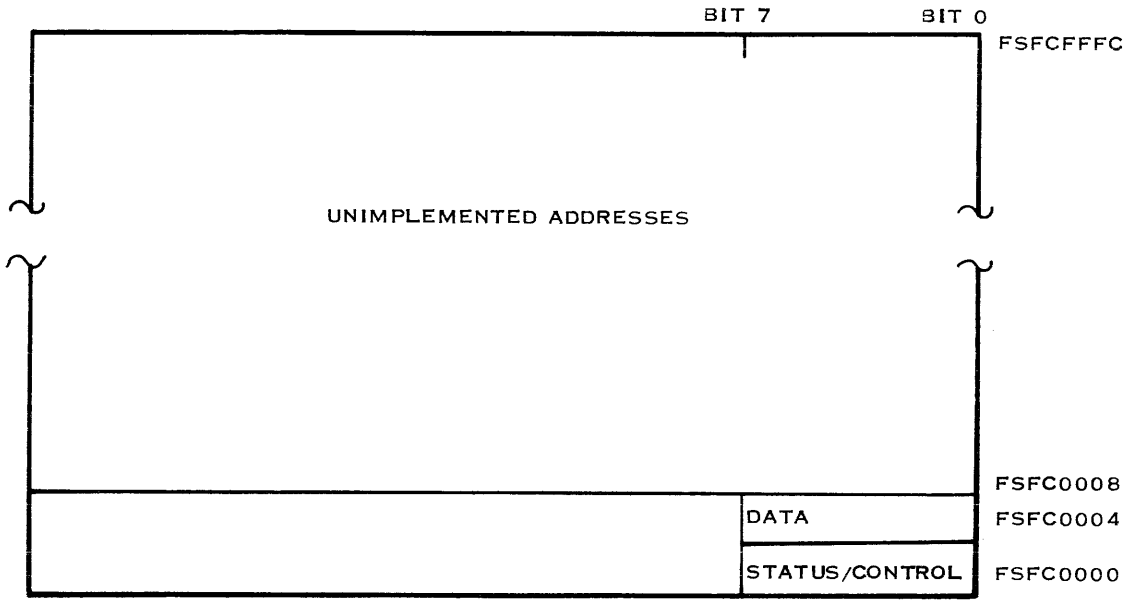
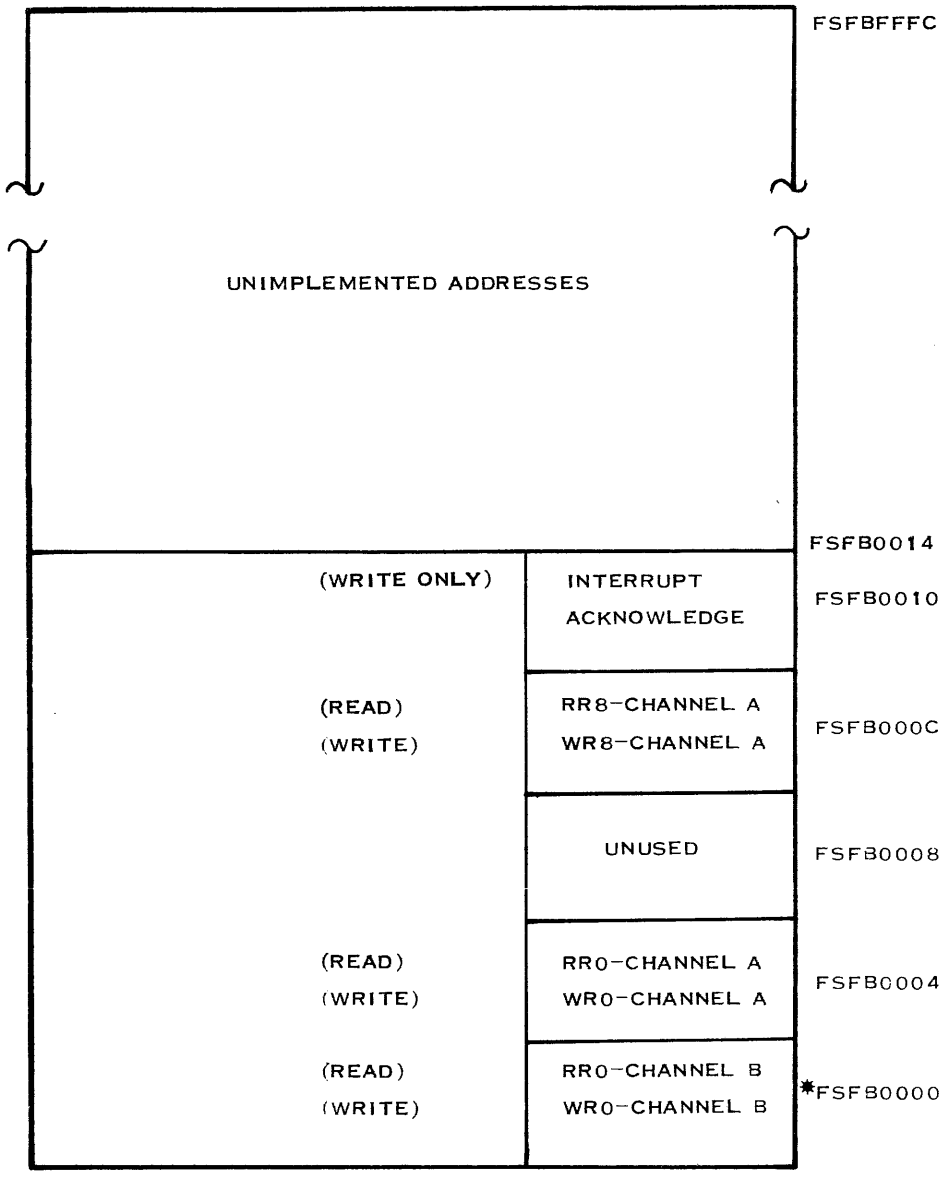


Figure A-3 RS-232C Port Memory Map



NOTE:

* ANY NUMBERED INTERNAL REGISTER IS ACCESSIBLE VIA A 2-STEP PROCESS. THE FIRST STEP WRITES THE DESIRED REGISTER NUMBER IN WR0. THE NEXT ACCESS (AT THE SAME ADDRESS) IS DIRECTED TO THE INTERNAL REGISTER.

Figure A-4 Nonvolatile RAM Memory Map

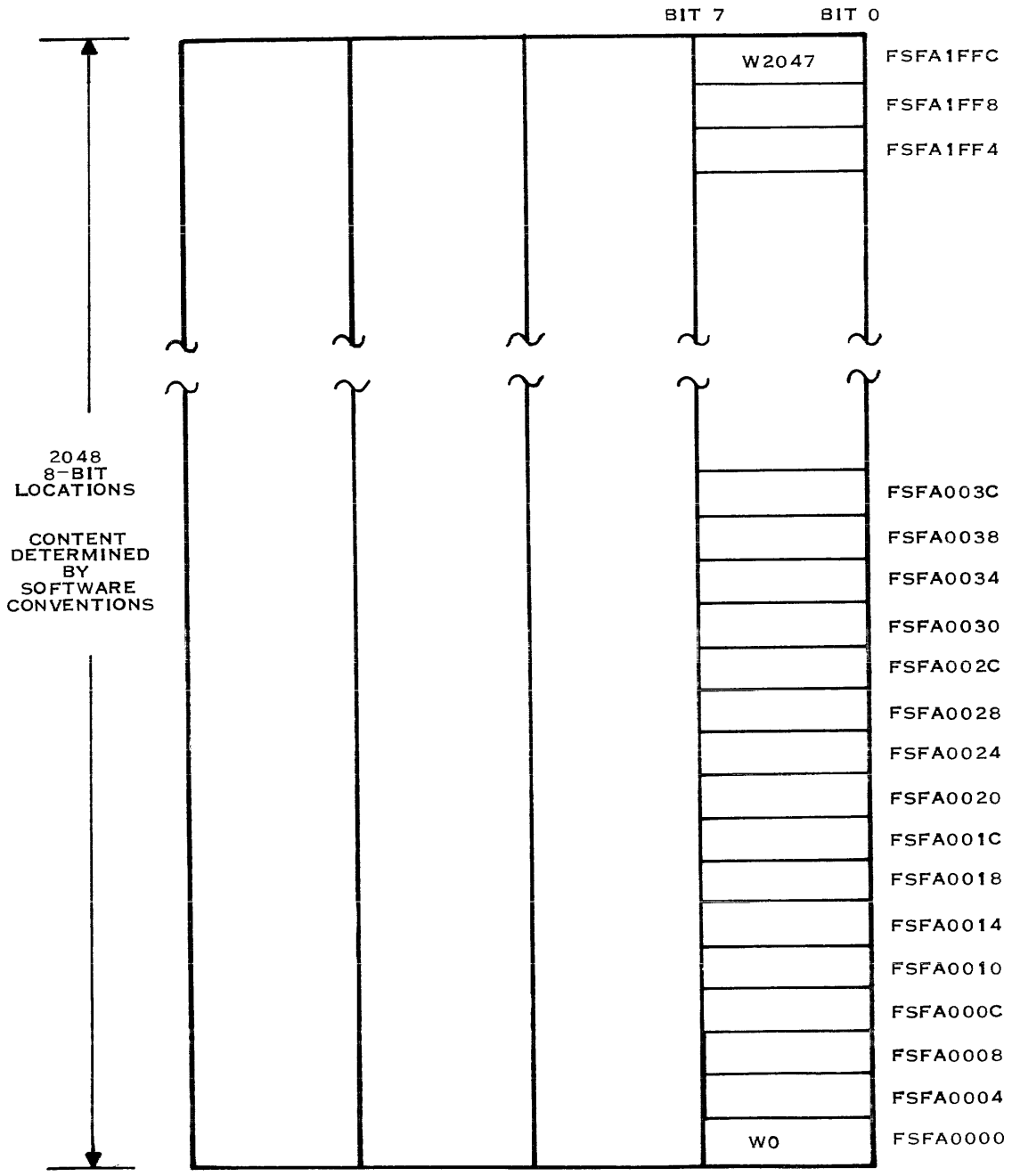


Figure A-5 Interval Timers Memory Map

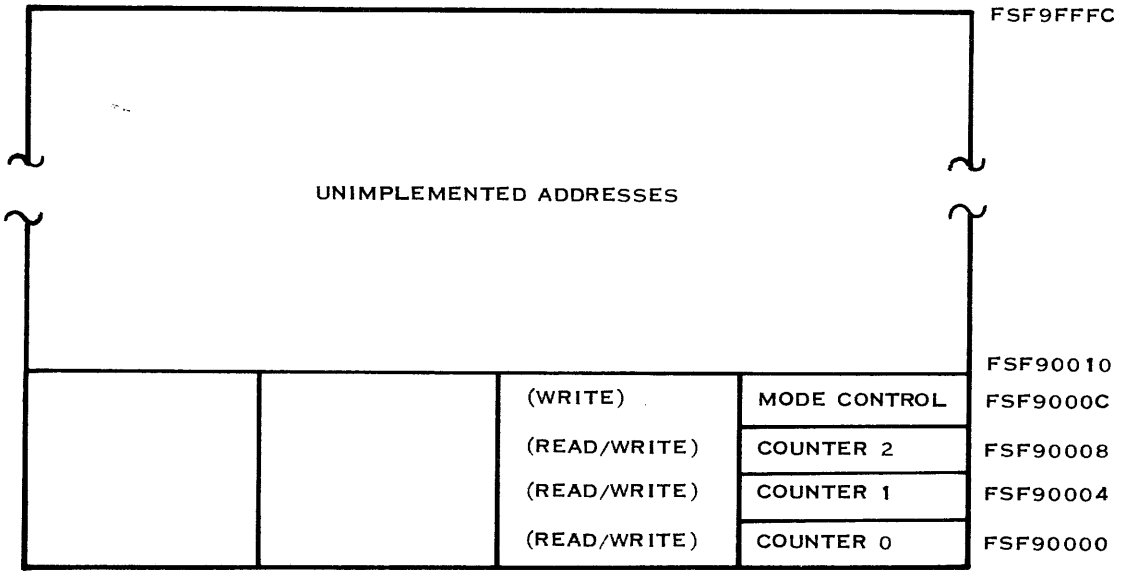


Figure A-6 Real-Time Clock Memory Map

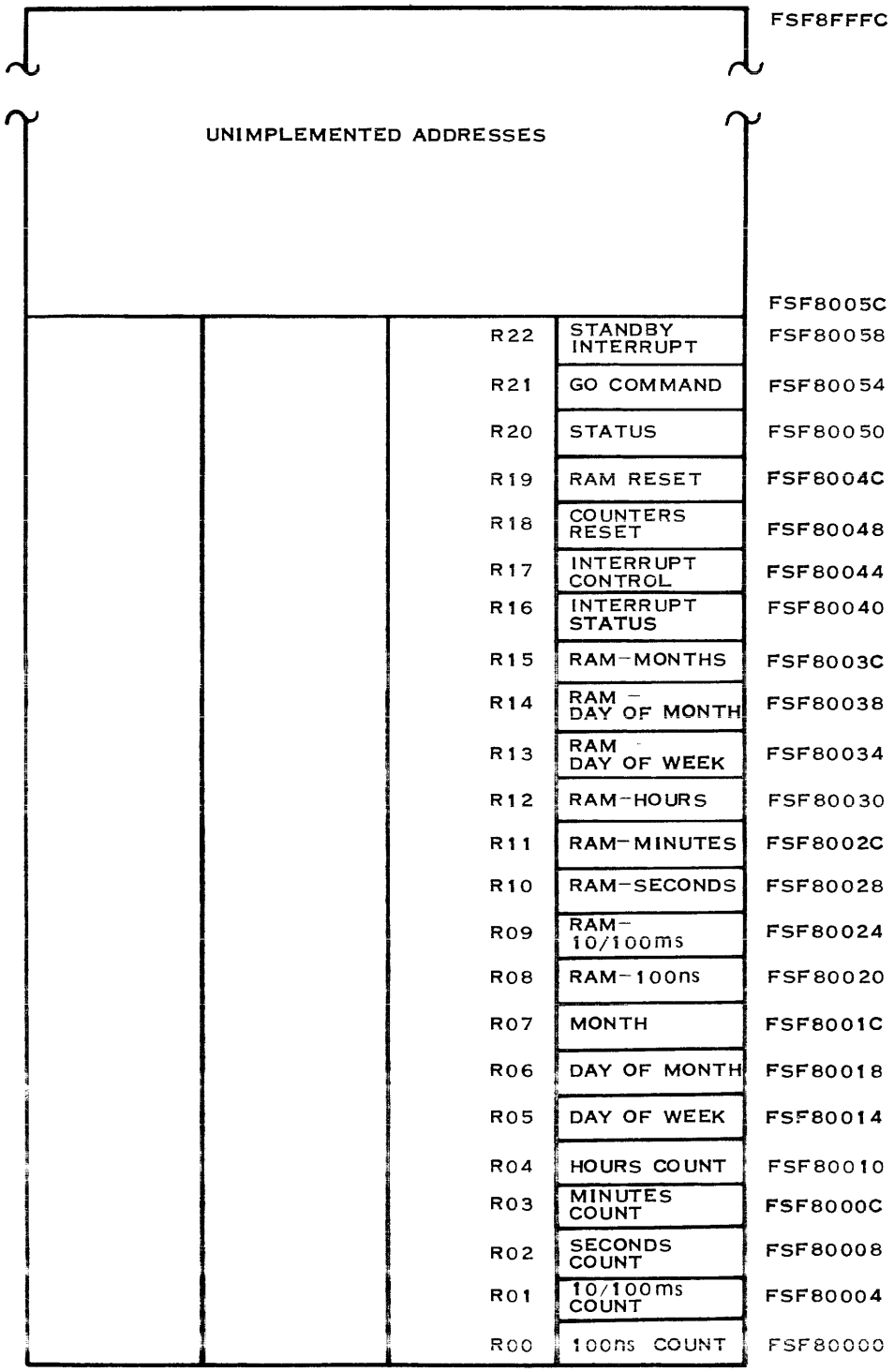


Figure A-7 Parallel Printer Port, Mouse, and Audio Registers Memory Map

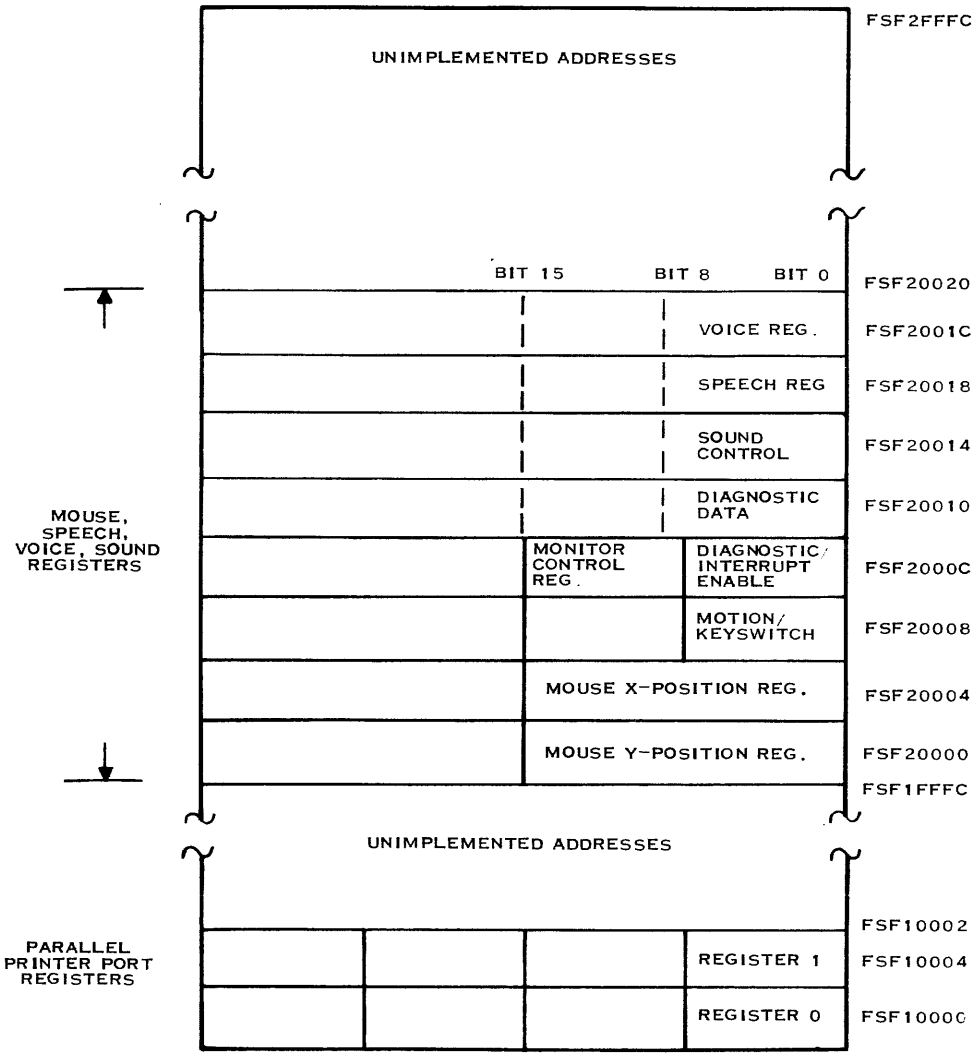
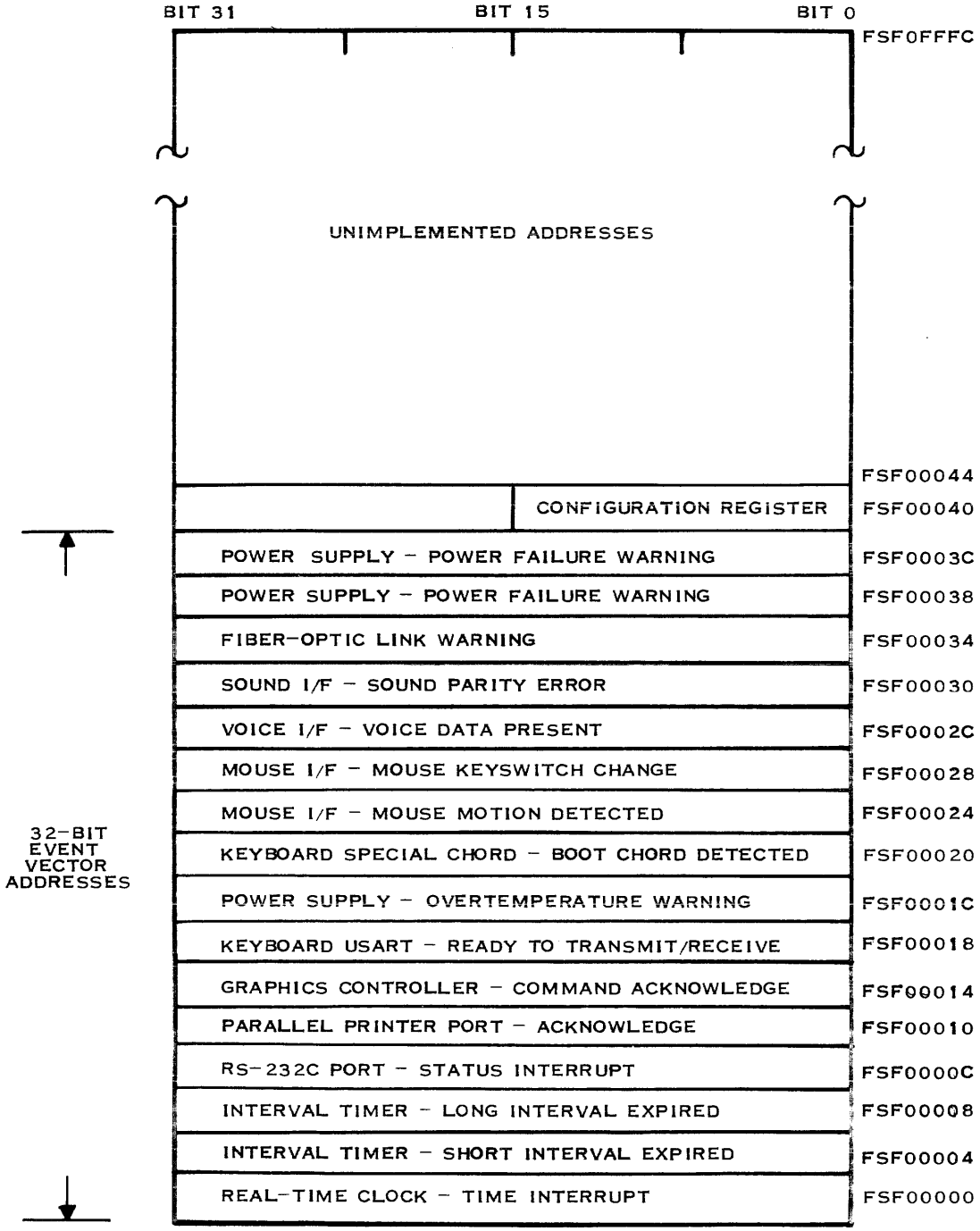


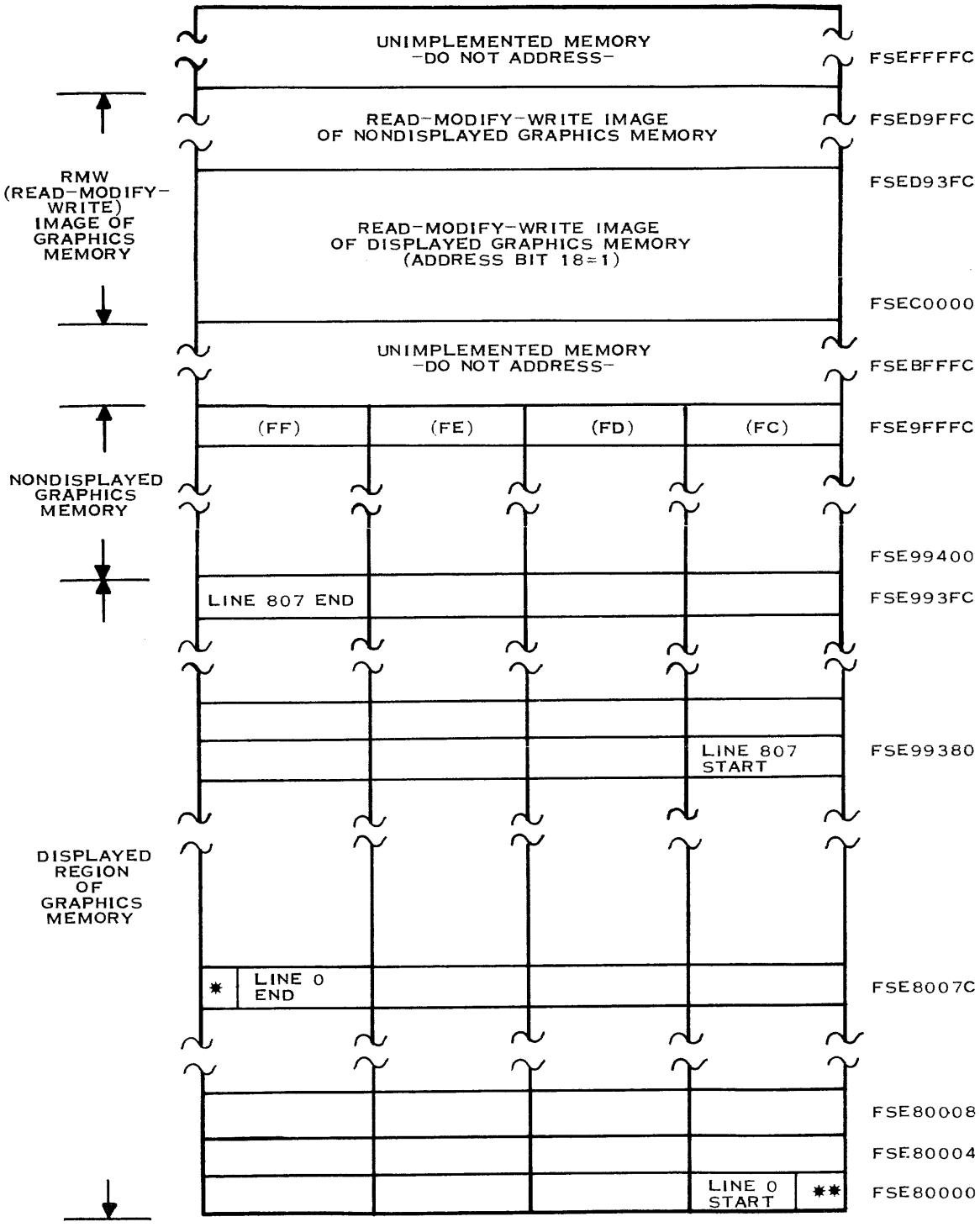
Figure A-8 Event Generator Vectors and Configuration Register Memory Map



NOTE:

EVENT ADDRESSES STORED IN THESE VECTORS MUST BE ALIGNED ON WORD BOUNDARIES (THAT IS, MUST BE MOUDLE-4 BYTE ADDRESSES).

Figure A-9 Memory Map for Graphics Bit-Mapped Memory



NOTES:

* EQUALS THE LAST PIXEL ON LINE 0.

**EQUALS THE FIRST PIXEL ON LINE 0.

Figure A-10 Graphics Control Memory Map (Sheet 1 of 2)

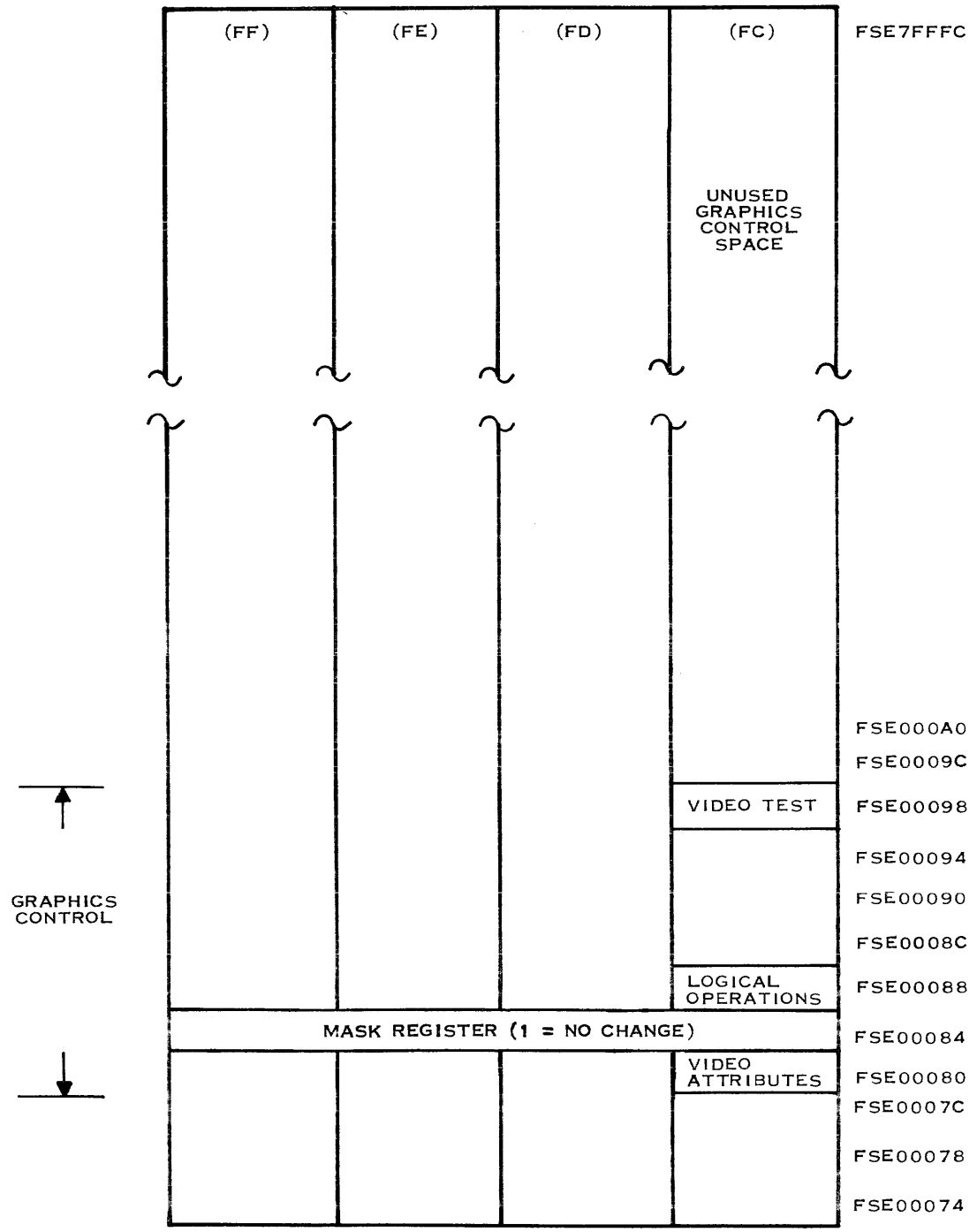
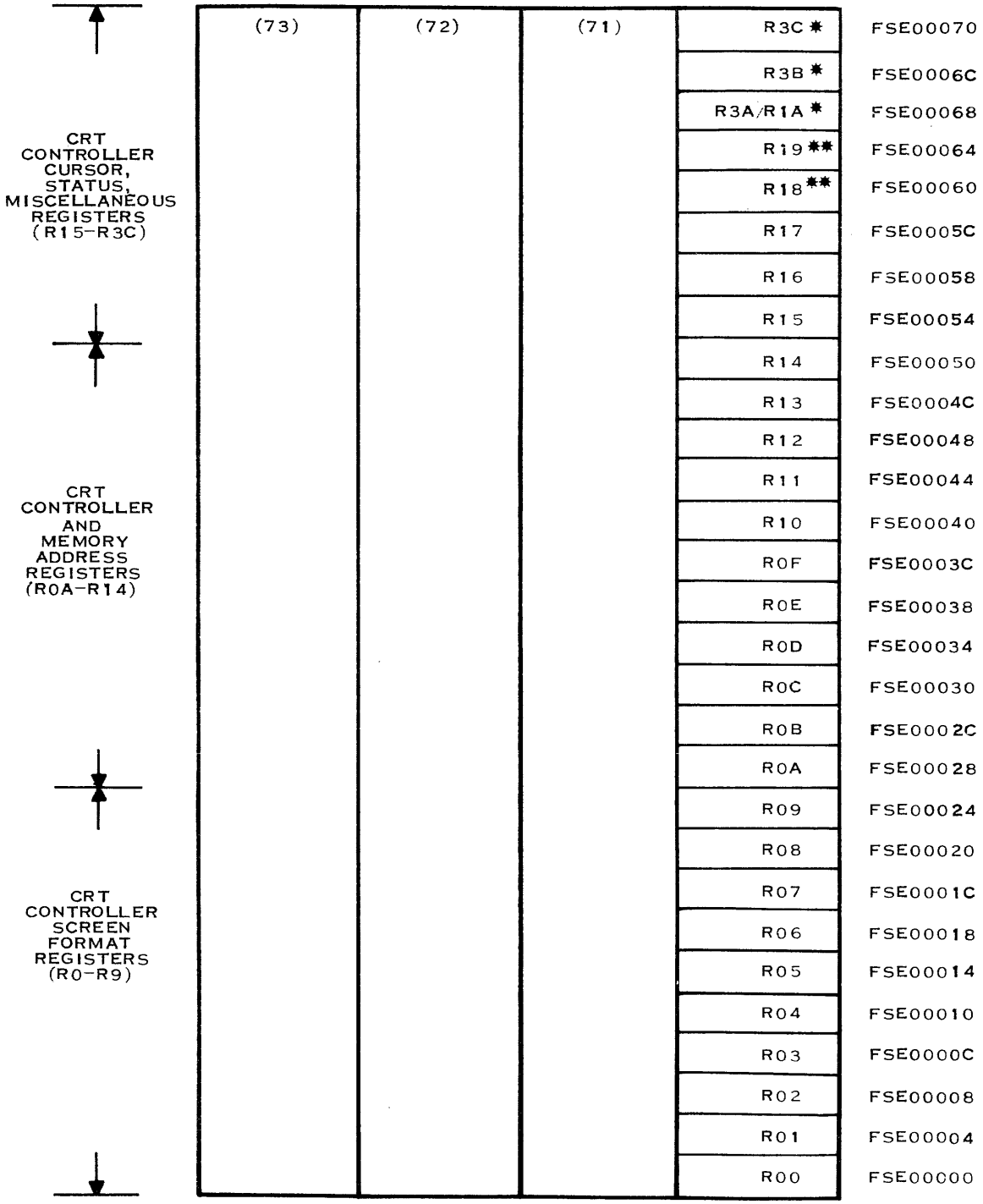


Figure A-10 Graphics Control Memory Map (Sheet 2 of 2)



NOTE:

*RESERVED (DO NOT USE). R3 IS READ-ONLY REGISTER.

**R18 AND R19 ARE READ/WRITE REGISTERS.

INDEX

A

Acknowledge Generator, NuBus 4-11
Address:
  Configuration, Board 1-8
  Decoders, Mouse/Keyboard/Audio Interface 4-48
  Map:
    Audio Registers 4-50
    Diagnostic Registers 4-50
    Mouse Registers 4-50
Addresses and Functions, USART Registers 4-62
Addressing Scheme, Bit-Mapped Memory 4-44
Audio Registers Address Map 4-50

B

Base Addresses, SI Board Device 4-6
Bit Assignments:
  Channel A Multiplexed Data 4-73
  Configuration Register 4-15
  Diagnostic Control Field 4-58
  Diagnostic Data Field 4-59
  Mouse Motion/Keyswitch Data Register 4-51
  Mouse Position Register 4-52
  Sound Control Register 4-56
  Speech Register 4-57
  Voice Register 4-58
  Z8530:
    Read Register 0 4-88
    Read Register 1 4-88
    Read Register 10 4-91
    Read Register 12 4-91
    Read Register 13 4-92
    Read Register 15 4-92
    Read Register 3 4-90
    Write Register 0 4-93
    Write Register 1 4-94
    Write Register 10 4-98
    Write Register 11 4-99
    Write Register 12 4-100
    Write Register 13 4-100
    Write Register 14 4-100
    Write Register 15 4-100
    Write Register 2 4-94
    Write Register 3 4-95
    Write Register 4 4-95
    Write Register 5 4-96
    Write Register 6 4-97
    Write Register 7 4-97
    Write Register 9 4-98
Bit Assignments Channel A, Z8530
  Read Register 2 4-89
Bit Assignments Channel B, Z8530
  Read Register 2 4-90
Bit-Mapped Graphics:
  CRT Controller 4-39
  Display Controller 4-35
    Block Diagram 4-37
    Programming 4-41
  Logical Operation Unit 4-39
  Shift Register Latch 4-39
  Shift Registers 4-39
  Video Generator 4-39
Bit-Mapped Memory 4-3
  Addressing Scheme 4-44
  Functional Description 4-35
  Operation 4-41
Block Diagram:
  Bit-Mapped Graphics Display Controller 4-35
  Explorer System 1-7
  SI Board 4-4
    Functional 4-7
Board:
  Address Configuration 1-8
  Slot Assignments 1-8
Bus Structure, SI Board 4-11

C

Channel A:
  Data Format 4-67, 4-89
  Description 4-68
  Multiplexed Data 4-73
  Transmit Section 4-74, 4-75
Channel B:
  Data Format 4-75, 4-76
  Description 4-74
  Receive Section 4-77
Characteristics:
  Fiber-Optic Board 1-12
  SI Board 1-10
Chord Detector 4-5, 4-66
Clock, NuBus 4-3, 4-10
Clock Timing, NuBus 4-11
Configuration Register 4-15
  Bit Assignments 4-15

Configuration ROM 4-12
Contents 4-12
Connector P3, Fiber-Optic Board 4-105
Connector P4, Fiber-Optic Board 4-106
Counter and RAM Reset Registers,
Real-Time Clock 4-22
CRT Controller:
Bit-Mapped Graphics 4-39
Initialization and Interrupts 4-42
Register Functions 4-43

D

Data Formats 4-70, 4-75
Detector, Mouse Motion 4-51
Diagnostic:
Control Field Bit Assignments 4-58
Data Field Bit Assignments 4-59
Register 4-57
Registers Address Map 4-50
Display Controller:
Bit-Mapped Graphics 4-35
Block Diagram 4-37
Programming 4-41
Graphics 4-3

E

Event:
Causes 4-17
Generator 4-5, 4-16
Initialization 4-18
Programming 4-17
Power-Failure 4-5
Explorer System 1-5, 1-6
Block Diagram 1-7

F

Fault Light Locations 3-4
Features:
Fiber-Optic Board 1-11
SI Board 1-9
Fiber-Optic:
Board 1-4
Characteristics 1-12
Connector P3 4-105
Connector P4 4-106
Features 1-11
Functional Description 4-103
General Description 1-3, 4-103
Installation Procedures 2-7
Interface 4-3, 4-67, 4-104
Logic 4-104
Parallel Printer Interface 4-104
Removal Procedures 2-8
RS-232C Interface 4-104
Unpacking Procedures 2-4
Functional Description:
Bit-Mapped Memory 4-35
Fiber-Optic Board 4-103
SI Board 4-5
Block Diagram 4-7

G

General Description:
Fiber-Optic Board 1-3, 4-103
SI Board 1-3
GO Command Register,
Real-Time Clock 4-23
Graphics Display Controller 4-3

I

Initialization:
At Power-Up, Interval Timers 4-29
Event Generator 4-18
Interrupts, CRT Controller 4-42
Interval Timers 4-28
Procedures, SI Board 4-102
Input/Output Connections, Z8530 4-80
Installation Procedures:
Fiber-Optic Board 2-7
SI Board 2-5
Interface:
Address Decoders,
Mouse/Keyboard/Audio 4-49
Fiber-Optic 4-3, 4-67, 4-104
Board:
Parallel Printer 4-104
RS-232C 4-104
Keyboard 4-5
Asynchronous Communication 4-60
Local Bus 1-7, 4-39
Mouse 4-3
Mouse/Keyboard/Audio 4-49
NuBus 1-7, 4-5, 4-37
Optical 4-105
Signals:
NuBus 4-10
Parallel Printer Port 4-32
TTL 4-107
Signal-Level 4-105
Interrupt:
Enable Register 4-52, 4-54
Handshake Controller 4-53
Status Register, Real-Time Clock 4-22
Interrupts, Real-Time Clock 4-22
Interval Timers 4-3, 4-23
Initialization 4-28
At Power-Up 4-29
Loading and Starting 4-29
Mode 0 Operation 4-24
Mode 3 Operation 4-25

Programming 4-25
Register Addresses and Functions 4-27
Square-Wave Rate Generator 4-30
Start-up Example 4-31

K

Keyboard Interface 4-5
Keyboard Asynchronous Communication Interface 4-60
Keyswitch Detector 4-52

L

Loading and Starting, Interval Timers 4-29
Local Bus:
Interface 1-7, 4-39
Signal Functions 4-40
Logical Operation Unit:
Bit-Mapped Graphics 4-39
Logical Functions 4-45
Logical Functions Example 4-46

M

Manchester Encoding 4-68, 4-71
Memory 1-8
Memory Map 4-5
Nonvolatile RAM 4-19
Mode Selection, USART 4-61
Mode 0 Operation, Interval Timers 4-24
Mode 3 Operation, Interval Timers 4-25
Monitor Control Register 4-52, 4-54
Monitor Timing 4-69
Mouse:
Interface 4-3
Motion and Position Logic 4-49
Motion Detector 4-51
Motion/Keyswitch Data Register 4-49
Bit Assignments 4-51
Position Counters and Registers 4-51
Position Register Bit Assignments 4-52
Registers Address Map 4-50
Mouse/Keyboard/Audio:
Interface 4-48
Address Decoders 4-49

N

Nonvolatile RAM 4-3, 4-18
Memory Map 4-19
Explorer Computer System Overview 1-5
NuBus:
Acknowledge Generator 4-11
Clock 4-3, 4-10
Timing 4-11
Interface 1-7, 4-5, 4-39
Signals 4-9
Time-Out Logic 4-10

O

Operation, Bit-Mapped Memory 4-41
Optical Interface 4-105

P

Parallel Printer Interface, Fiber-Optic Board 4-104
Parallel Printer Port 4-5, 4-31
Interface Signals 4-32
Programming 4-33
Register 0 Functions 4-33
Register 1 Functions 4-34
Port:
Input and Output Signals, RS-232C 4-81
Programming, RS-232C 4-83
Example, RS-232C 4-102
Power Failure Event 4-5
Programmable Baud Rate Generator, Z8530 4-86
Baud Rate Generator Divisors, Z8530 4-87
Programming:
Bit-Mapped Graphics Display Controller 4-41
Event Generator 4-17
Interval Timers 4-25
Parallel Printer Port 4-33
Real-Time Clock 4-20
RS-232C Port 4-83
Example 4-102
USART 4-61

R

Raster-Scan Display Example 4-69
Read Register Functions, Z8530 4-84
Read Register 0:
Bit Assignments, Z8530 4-88
Z8530 4-87
Read Register 1:
Bit Assignments, Z8530 4-88
Z8530 4-87
Read Register 10:
Bit Assignments, Z8530 4-91
Z8530 4-89
Read Register 12:
Bit Assignments, Z8530 4-91
Z8530 4-89
Read Register 13:
Bit Assignments, Z8530 4-92
Z8530 4-90

Read Register 15:
Bit Assignments, Z8530 4-92
Z8530....4-90
Read Register 2:
Bit Assignments Channel A, Z8530 4-89
Bit Assignments Channel B, Z8530....4-90
Z8530....4-89
Read Register 3:
Bit Assignments, Z8530 4-90
Z8530....4-89
Real-Time Clock....4-3, 4-20
Counter and RAM Reset Registers....4-23
GO Command Register 4-23
Interrupt Status Register....4-22
Interrupts....4-22
Programming 4-20
Registers....4-21
Status Bit Register 4-23
Timing and Registers Format 4-21
Real-Time Counter....4-20
Receive (Channel A)....4-68
Register:
Addresses and Functions,4-27
Interval Timers....4-27
File Storage Locations....4-17
Functions, CRT Controller 4-43
Video Attributes....4-47
Video Test 4-49
0 Functions, Parallel Printer Port 4-33
1 Functions, Parallel Printer Port 4-33
Registers:
Addresses and Functions, USART 4-62
Real-Time Clock....4-21
Removal Procedures:
Fiber-Optic Board....2-9
SI Board....2-6
RS-232C:
Interface, Fiber-Optic Board....4-104
Port....4-3, 4-77
Input and Output Signals....4-81
Programming 4-83
Programming Example....4-102
Register Addresses....4-86

S

Self-Test Procedures, SI Board 3-3
Shift Register Latch, Bit-Mapped
Graphics 4-39
Shift Registers, Bit-Mapped Graphics 4-39
SI Board....1-4
Block Diagram 4-4
Bus Structure....4-11
Characteristics....1-10
Device Base Addresses 4-1
Features....1-9
Functional Block Diagram....4-7
Functional Description....4-5
General Description 1-3
Initialization Procedures....4-102
Installation Procedures 2-5
Removal Procedures....2-6
Self-Test Procedures 3-3
Unpacking Procedures....2-3
Signal Functions, Local Bus 4-40
Slot Assignments, Board....1-8
Sound Control Logic 4-3
Sound Control Register....4-55
Bit Assignments....4-56
Speech Register....4-56
Bit Assignments....4-57
Square-Wave Rate Generator,
Interval Timers....4-30
Start-Up Example, Interval Timers....4-31
Status Bit Register, Real-Time Clock 4-22

T

Time-Out Logic, NuBus....4-10
Timing and Registers Format,
Real-Time Clock....4-21
Timing, Monitor 4-68, 4-69
Transmit (Channel A)....4-74
TTL Interface 4-107
TTL Signal-Level Interface 4-105

U

Unpacking Procedures:
Fiber-Optic Board....2-4
SI Board....2-3
USART:
Command Instruction Control
Format....4-64
Mode Instruction Control Format....4-63
Mode Selection....4-61
Operating Parameters Selection....4-62
Operations....4-60
Programming 4-61
Receive Operation 4-61
Registers Addresses and Functions 4-62
Status Byte Format....4-65
Status Reporting....4-63
Transmit and Receive Data Byte
Format....4-66
Transmit Operation 4-61

V

Video Attributes Register 4-47
Video Generator, Bit-Mapped
Graphics 4-39
Video Requirements 4-68
Video Test Register....4-39, 4-49
Voice Interrupt Generator 4-57

Voice Register 4-57
Bit Assignments 4-58
Voice/Speech I/O 4-3

W

Write Register Functions, Z8530 4-85
Write Register 0:
Bit Assignments, Z8530 4-93
Z8530 4-93
Write Register 1:
Bit Assignments, Z8530 4-94
Z8530 4-93
Write Register 10:
Bit Assignments, Z8530 4-98
Z8530 4-96
Write Register 11:
Bit Assignments, Z8530 4-99
Z8530 4-99
Write Register 12:
Bit Assignments, Z8530 4-100
Z8530 4-99
Write Register 13:
Bit Assignments, Z8530 4-100
Z8530 4-99
Write Register 14:
Bit Assignments, Z8530 4-101
Z8530 4-99
Write Register 15:
Bit Assignments, Z8530 4-101
Z8530 4-99
Write Register 2:
Bit Assignments, Z8530 4-94
Z8530 4-93
Write Register 3:
Bit Assignments, Z8530 4-95
Z8530 4-93
Write Register 4:
Bit Assignments, Z8530 4-95
Z8530 4-93
Write Register 5:
Bit Assignments, Z8530 4-96
Z8530 4-96
Write Register 6:
Bit Assignments, Z8530 4-97
Z8530 4-96
Write Register 7:
Bit Assignments, Z8530 4-97
Z8530 4-96
Write Register 9:
Bit Assignments, Z8530 4-98
Z8530 4-96

Z

Z8530:
Input/Output Connections 4-80
Programmable Baud Rate Generator 4-86
Divisors 4-87
Read Register Functions 4-84
Read Register 0 4-87
Bit Assignments 4-88
Read Register 1 4-87
Bit Assignments 4-88
Read Register 10 4-89
Bit Assignments 4-91
Read Register 12 4-89
Bit Assignments 4-91
Read Register 13 4-90
Bit Assignments 4-92
Read Register 15 4-90
Bit Assignments 4-92
Read Register 2 4-89
Bit Assignments Channel A 4-89
Bit Assignments Channel B 4-90
Read Register 3 4-89
Bit Assignments 4-90
Write Register Functions 4-85
Write Register 0 4-93
Bit Assignments 4-93
Write Register 1 4-93
Bit Assignments 4-94
Write Register 10 4-96
Bit Assignments 4-98
Write Register 11 4-99
Bit Assignments 4-99
Write Register 12 4-99
Bit Assignments 4-100
Write Register 13 4-100
Bit Assignments 4-99
Write Register 14 4-99
Bit Assignments 4-101
Write Register 15 4-99
Bit Assignments 4-101
Write Register 2 4-93
Bit Assignments 4-94
Write Register 3 4-93
Bit Assignments 4-95
Write Register 4 4-93
Bit Assignments 4-95
Write Register 5 4-96
Bit Assignments 4-96
Write Register 6 4-96
Bit Assignments 4-97
Write Register 7 4-96
Bit Assignments 4-97
Write Register 9 4-96
Bit Assignments 4-98

USER RESPONSE SHEET

Our manuals are written for you. Please help us improve them by completing and mailing this form. List any discrepancy or other problem that you found in this manual by page, paragraph, figure, or table number in the space provided, and add any suggestions you have. Thank you.

Manual Title: _____

Manual Date: _____ **Today's Date:** _____

Your Name: _____ **Telephone:** _____

Company: _____ **Department:** _____

Company Address: _____

Product Purchased From: _____

Company **City** **State**

Location in Manual	**Comment/Suggestion**
_____	_____
	_____
	_____
_____	_____
	_____
	_____
_____	_____
	_____
	_____

The manual's good points: _____

_____

What needs to be improved: _____

_____

No postage necessary if mailed in U.S.A.
Fold on two lines (located on reverse side), tape and mail

TAPE EDGE TO SEAL

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

TEXAS INSTRUMENTS INCORPORATED
DATA SYSTEMS GROUP
ATTN: PUBLISHING CENTER
P.O. Box 2909 M/S 2146
Austin, Texas 78769-9990

FOLD