MOTOROLA

M68SFDC2(D)

M68SFDC2
EXORdisk II
FLOPPY DISK CONTROLLER MODULE

User's Guide

SYSTEMS

MICROSYSTEMS

M68SFDC2(D)
JUNE 1978

M68SFDC2

FLOPPY DISK CONTROLLER MODULE
USER'S GUIDE

First Edition

TABLE OF CONTENTS

		Page
CHAPTER 1:	GENERAL INFORMATION	
1.1	INTRODUCTION	1-1
1.2	FEATURES	1-1
1.3	SPECIFICATIONS	1-1
1.4	GENERAL DESCRIPTION	1-1
1.5	EQUIPMENT SUPPLIED	1-3
CHAPTER 2:	INSTALLATION INSTRUCTIONS, PROGRAMMING, FAULT ISOLATION, AND INTERCONNECTION CONSIDERATIONS	
2.1	INTRODUCTION	2-1
2.2	UNPACKING INSTRUCTIONS	2-1
2.3	INSPECTION	2-1
2.4	HARDWARE PREPARATION	2-1
2.4.1	Memory Map Option	2-1
2.4.2	Clock Speed	2-2
2.4.3	Dynamic System Bus (DSB)	2-2
2.4.4	Multiple Head/Quad Drive Option	2-2
2.5	INSTALLATION INSTRUCTIONS	2-2
2.6	PROGRAMMING CONSIDERATIONS	2-2
2.7	FAULT ISOLATION	2-3
2.8	MODULE INTERCONNECTIONS	2-6
CHAPTER 3:	THEORY OF OPERATION	
3.1	INTRODUCTION	3-1
3.2	BLOCK DIAGRAM DESCRIPTION	3-1
3.2.1	ROM Resident Driver Firmware (E800 to EBFF)	3-1
3.2.1.1	Initialization	3-1
3.2.1.2	Error Messages	3-1
3.2.1.3	Resident Driver Firmware Entry Points	3-6
3.2.2	Recording Format	3-10
3.2.3	Read Operation	3-11
3.2.3.1	PLL Data Recovery Circuit	3-11
3.2.3.2	CRC Read Error Check Circuit	3-13
3.2.4	Write Operation	3-14
CHAPTER 4:	PARTS	
4.1	INTRODUCTION	4-1

LIST OF ILLUSTRATIONS

FIGURE		
1-1	Floppy Disk Controller Module	1-2
3-1	Floppy Disk Controller Module Block Diagram	3-2
3-2	Read Data Circuit Block Diagram	3-12
3-3	Write Data Circuit Block Diagram	3-15
3-4	Floppy Disk Controller Module Schematic Diagram	3-16

LIST OF TABLES

TABLE		Page
1-1	Floppy Disk Controller Module Specifications	1-3
2-1	Fault Isolation	2-3/2-5
2-2	Connector P1 Bus Interface Signals	2-6/2-8
2-3	Connector P2 Interconnection Signals	2-8/2-9
3-1	Initialization Parameters for a User Prepared DOS Program	3-3
3-2	Disk Error Messages	3-4/3-5
3-3	Resident Driver Firmware Program Entry Points	3-6/3-9
4-1	Floppy Disk Controller Module (M68SFDC2) Parts List	4-1/4-2
4-2	EXORdisk II to EXORciser Interconnecting Cable Assembly (M68SFD1C) Parts List	4-2

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

This manual provides general information, preparation for use, installation instructions, programming considerations, theory of operation adjustments, and illustrated parts list for the M68SFDC2 Floppy Disk Controller Module. A typical module is shown in Figure 1-1. All address references within this manual are shown in hexadecimal unless otherwise indicated.

1.2 FEATURES

The features of the Floppy Disk Controller Module include:

- Compatible with EXORciser I and II, and EXORterm.
- Operable from 880 KHz to 2 MHz with Static, or 910 KHz to 2 MHz with Dynamic Memory using cycle stealing or refresh.
- Direct control of the EXORdisk II Disk Drive Unit via Resident Driver Firmware.
- Formatting and control of synchronous serial data transferred between the EXORdisk II Disk Drive Unit and the EXORciser or EXORterm.
- Direct interface with the EXORciser bus or the EXORterm bus.
- Fully compatible with the Motorola Disk Operating System (MDOS) software.
- Series II DSB (Dynamic System Bus) provides Page Enable and Priority Interrupt control.
- Enable jumpers allow module to be addressed via VUA, VXA, or Page Enable.
- TTL Voltage compatible high impedance inputs.

1.3 SPECIFICATIONS

The specifications for the Floppy Disk Controller Module are identified in Table 1-1.

1.4 GENERAL DESCRIPTION

The Floppy Disk Controller Module provides the disk drive control and data transfer functions for the EXORdisk II Floppy Disk System. This system includes the EXORdisk II Disk Drive Unit and the MDOS Disk Operating System software. This module is bus compatible with both the EXORciser and the EXORterm family.

The address allocation for the Floppy Disk Controller Module extends from E800 to EC07, inclusively. This range of memory addresses is divided into the following areas:

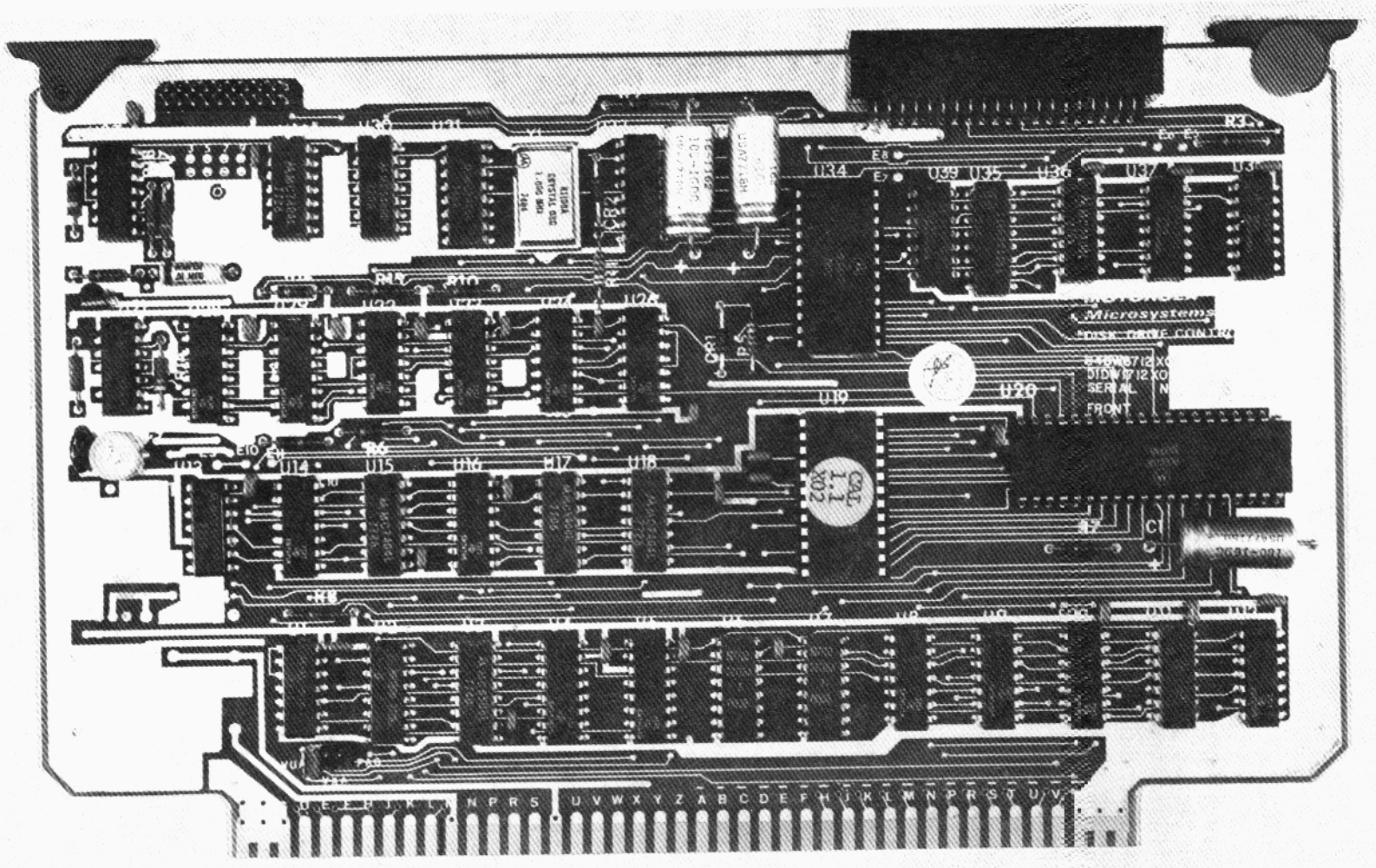


FIGURE 1-1. Floppy Disk Controller Module

- E800 through EBFF are assigned to the Resident Driver Firmware
- EC00 through EC03 are assigned to control the operation of the Disk Drive Unit.
- EC04 through EC07 are assigned to format and control the flow of data to and from the Disk Drive Unit.

The M68SFDC2 Floppy Disk Controller Module consists of a Peripheral Interface Adapter (PIA) used to control the operation of the Disk Drive Unit, a Synchronous Serial Data Adapter (SSDA) used to format and control the flow of data to and from the Disk Drive Unit, a 1024 x 8-bit Read Only Memory (ROM) containing the Resident Driver Firmware program, a CRC generation circuit, three-state bus interface circuitry, and the necessary address decoding logic to permit accessing from either the EXORciser bus or EXORterm bus.

1.5 EQUIPMENT SUPPLIED

The Floppy Disk Controller Module may be shipped separately or as part of the EXORdisk II Floppy Disk System. An Interface Cable is supplied in either case.

TABLE 1-1. Floppy Disk Controller Module Specifications

CHARACTERISTICS	SPECIFICATIONS
Power Requirements	+5 Vdc @ 2.75 A +12 Vdc @ 20 mA -12 Vdc @ 45 mA (max)
Bus Interface Signals	
Address Bus	TTL compatible
Data Bus	Bi-directional three-state TTL voltage compatible
Control Bus	TTL voltage compatible
EXORdisk II Interface Signals	TTL voltage compatible
Operating Temperature	0^{o} to 70^{o}C.
Dimensions	
Width x Height	9.75 in. x 6.15 in.
Board Thickness	0.062 in.
Connector Type	
Disk Drive Unit Connector (P2)	Standard Applied Engineering 3M 3417-3000 or equivalent

CHAPTER 2

INSTALLATION INSTRUCTIONS, PROGRAMMING, FAULT ISOLATION, AND INTERCONNECTION CONSIDERATIONS

2.1 INTRODUCTION

This chapter provides the unpacking, inspection, installation, programming, fault isolation, and interconnection instructions for the M68SFDC2 Floppy Disk Controller Module. This chapter also discusses the module's interconnection signals.

2.2 UNPACKING INSTRUCTIONS

Unpack the Floppy Disk Controller Module from its shipping carton and, referring to the packing list, verify that all of the items are present. Save the packing materials for storing and reshipping of the module. If the shipping carton is damaged upon receipt, request that the carrier's agent be present while the module is being unpacked and inspected.

2.3 INSPECTION

The Floppy Disk Controller Module should be inspected upon receipt for broken, damaged, or missing parts and physical damage to the printed circuit board.

2.4 HARDWARE PREPARATION

The Floppy Disk Controller Module has option features which the user may select through jumper installation. The module is shipped with jumpers installed which configure the module to operate in an EXORciser or EXORterm system.

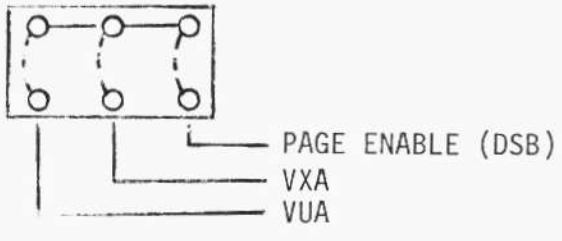
2.4.1 Memory Map Option

The user has the option, through a jumper, of selecting one of the three memory map modes:

VUA - Valid User Address
VXA - Valid Executive Address
PAGE ENABLE - For multiple "pages" of 64K bytes

The module is shipped with VUA selected. Since the Controller operates with MDOS and EXbug, the selection will usually be VUA for Single Map mode, or VXA for Dual Map mode.

A jumper located in the lower left-hand corner of the module is used for the option selection, as shown below:



2.4.2 Clock Speed

The Controller contains a 1.0 MHz time base which allows it to operate with less dependency on system clock. The user may select to operate the controller from the memory clock input on the system bus (900 KHz to 2.0 MHz). At the center left-hand portion are three pads: E9, E10, and E11. To connect to memory clock, the track must be cut between pins E10 and E11. A jumper must be installed between E9 and E10.

2.4.3 Dynamic System Bus (DSB)

Only one meaningful connection is available to the Dynamic System Bus header. PAGE ENABLE is connected to DSB, pin 19 with ground on pin 20. (See paragraph 2.4.1.)

Provisions are made on the module for $\overline{\text{IRQ}}$ input to the DSB. This option, however, would require changes to the ROM firmware, and is reserved for advanced system designs.

2.4.4 Multiple Head/Quad Drive Option

Pads E5, E6, E7, E8, E12, and E13 are provided on the module for jumper connection of $\overline{\text{SEL 2/3}}$ and $\overline{\text{HEAD 1}}$ (double side disk). These connections are provided for future system expansion.

2.5 INSTALLATION INSTRUCTIONS

The Floppy Disk Controller Module can be used with either the EXORciser or the EXORterm. To install the module, proceed as follows:

a. Turn power OFF on the equipment in which module is being installed.

CAUTION

Inserting a Floppy Disk Controller Module while power is applied to the system may result in damage to components on the module.

b. Install the module in the selected card slot. (This module may be installed in any of the card slots. However, it is desirable to keep this module as close to the power connector as possible.)

c. Install the interconnecting cable (observe pin index mark on connector, also pin 10 on cable connector is plugged and pin 10 on module connector is cut off as a means of ensuring proper connection).

d. Turn power ON.

2.6 PROGRAMMING CONSIDERATIONS

The EXORdisk II Floppy Disk Drive Unit is controlled by the Resident Driver Firmware located on the Floppy Disk Controller Module. This firmware program is used to control the two disk drive units and to verify disk operation. However, the data stored on the diskette and the file management of recorded data are controlled by the Motorola Disk Operating System (MDOS) software. When the EXORdisk II Floppy Disk System is used with the EXORciser or the EXORterm, 16K of contiguous working memory is required for address 0000-3FFF. MDOS software is available to the user to control all file management operations. When the EXORdisk II Floppy Disk System is incorporated into a user system design, the MDOS software cannot be used. Therefore, those users must write their own Disk Operating System software.

2.7 FAULT ISOLATION

Should a fault occur, the information in Table 2-1 will enable the user to determine if the fault is malfunctioning module, component, disk drive, diskette, or operator error. The user may use Table 2-1 by starting at a point in the flow chart which best describes the particular problem.

TABLE 2-1. Fault Isolation

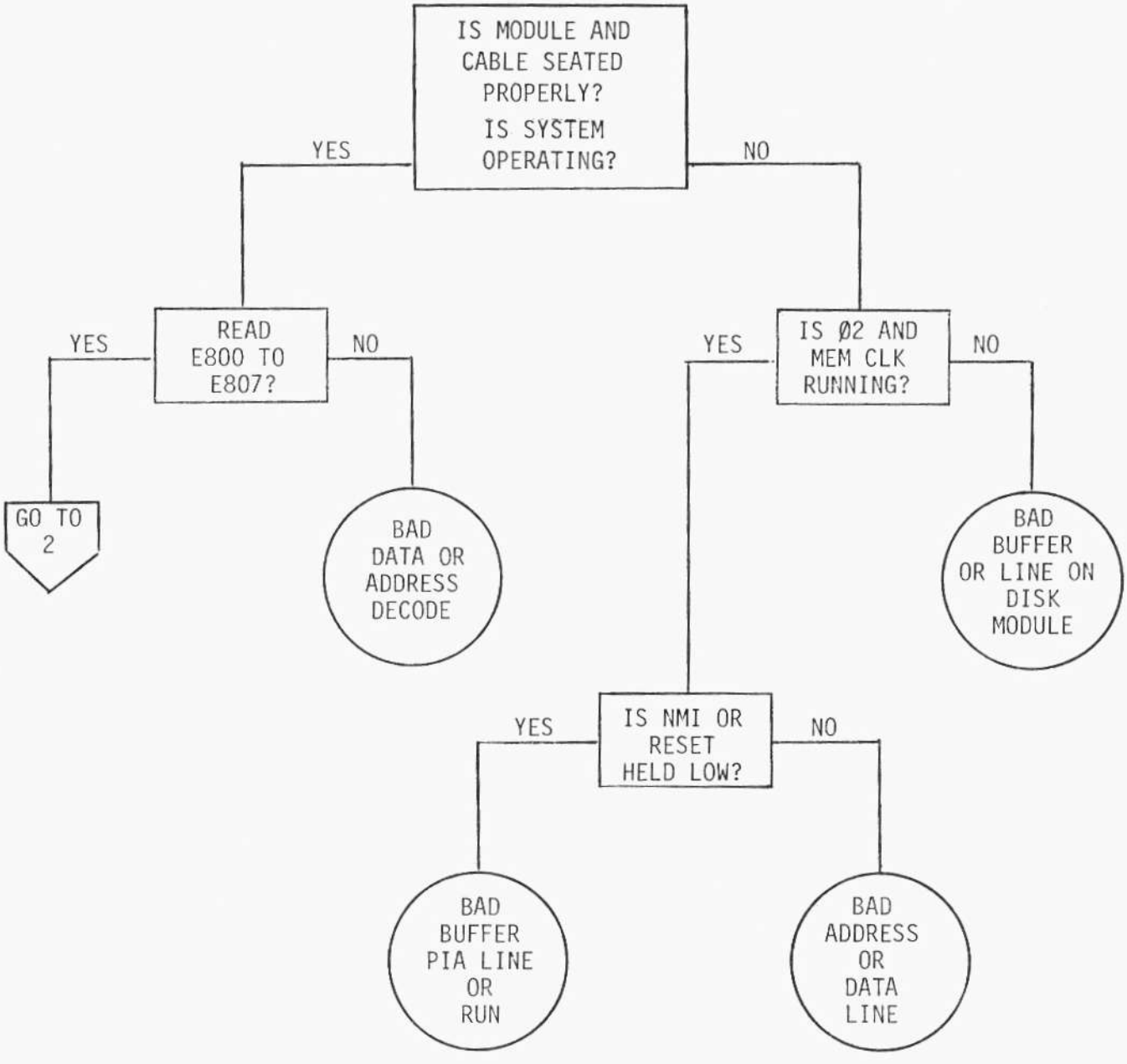


TABLE 2-1. Fault Isolation (cont'd)

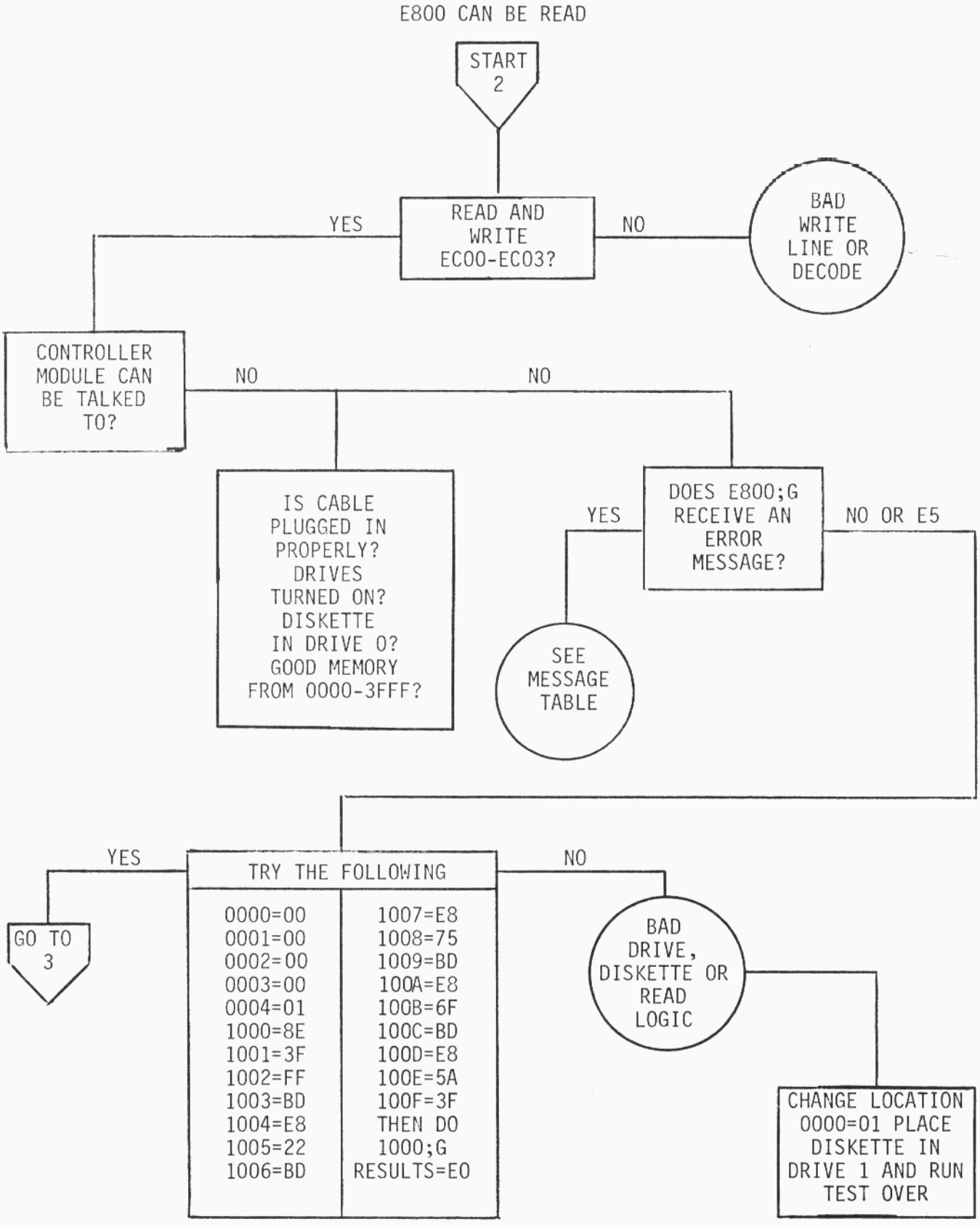
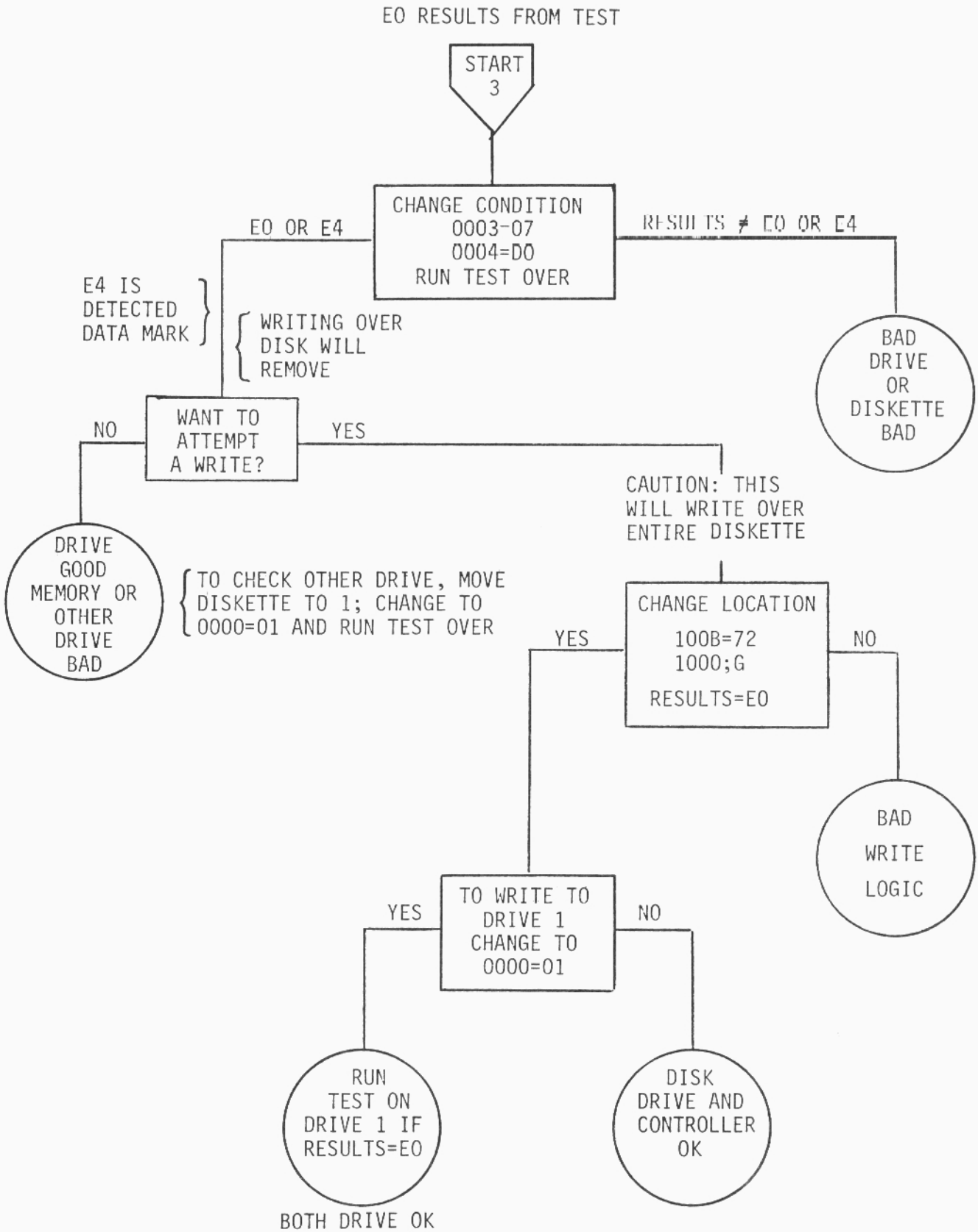


TABLE 2-1. Fault Isolation (cont'd)



2.8 MODULE INTERCONNECTIONS

The Floppy Disk Controller Module is bus compatible with either the EXORciser bus or the EXORterm bus. The bus signals used on this module are identified in Table 2-2. This table lists each pin connection, signal mnemonic, and signal characteristic. Table 2-3 identifies the various EXORdisk II Floppy Disk System interface signals.

TABLE 2-2. Connector P1 Bus Interface Signals

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
A,B,C	+5V	+5 Vdc - Used for module's logic circuits.
D	$\overline{IRQ}$	INTERRUPT REQUEST - This low level output signal to the MPU requests that an interrupt sequence be initiated. The MPU will wait until it completes the instruction being executed before it recognizes the request. At that time, if the interrupt mask bit in the MPU condition code register is not set, the MPU will begin the interrupt sequence.
E	$\overline{NMI}$	NON-MASKABLE INTERRUPT - This low-going output signal requests that a non-maskable interrupt be generated within the MPU. The MPU will wait until it completes the instruction being executed before it recognizes the NMI signal. At that time, regardless of the setting of the interrupt mask bit in the MPU, the MPU will begin the interrupt sequence.
F,H		Not used.
J	ø2	Phase 2 clock signal.
K		Not used.
L	MEM CLK	MEMORY CLOCK - This signal is an ungated TTL level ø2 clock used by the module to generate the Write Clock signal.
M	-12V	-12Vdc - Used to generate the -5Vdc required by the Read Only Memory (ROM) containing the Resident Driver Firmware. (Not used with 82S2708 ROM device.)
N to S		Not used.
T	+12V	+12Vdc - Used by the Read Only Memory (ROM) containing the Resident Driver Firmware. (Not used with 82S2708 ROM device.)
U to Z		Not used.
$\overline{A}$ to $\overline{F}$		Not used.
$\overline{H}$	$\overline{D3}$	DATA bus (bit 3) - This bi-directional line, when enabled, provides a two-way data transfer between this module and the module containing the MPU.

TABLE 2-2. Connector P1 Bus Interface Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
J	D7	DATA bus (bit 7) - Same as D3 on P1-H.
K	D2	DATA bus (bit 2) - Same as D3 on P1-H.
L	D6	DATA bus (bit 6) - Same as D3 on P1-H.
M	A14	ADDRESS bus (bit 14) - One of 16 input address lines used to select a specific disk operation.
N	A13	ADDRESS bus (bit 13) - Same as A14 on P1-M.
P	A10	ADDRESS bus (bit 10) - Same as A14 on P1-M.
R	A9	ADDRESS bus (bit 9) - Same as A14 on P1-M.
S	A6	ADDRESS bus (bit 6) - Same as A14 on P1-M.
T	A5	ADDRESS bus (bit 5) - Same as A14 on P1-M.
U	A2	ADDRESS bus (bit 2) - Same as A14 on P1-M.
V	A1	ADDRESS bus (bit 1) - Same as A14 on P1-M.
W,X,Y	GND	GROUND
1,2,3	+5V	+5Vdc - Used for module's logic circuits.
4		Not used.
5	RES	RESET - This buffered input signal permits the module to be restarted by either an external switch closure to ground or by the low level RESET signal from the module containing the MPU.
6	R/W	READ/WRITE - This signal controls whether data is to be read (high) from the module or written (low) into the module. The normal standby state of the signal is read (high). This signal will also be in the read state whenever the MPU is halted.
7 to 9		Not used.
10	VUA	VALID USER'S ADDRESS - This signal, jumper selectable, is produced by the DEbug module. When high, this signal indicates that the address on the address bus is valid and the MPU is NOT addressing the EXbug program.
11	-12V	-12Vdc - Used to generate the -5Vdc required by the Read Only Memory (ROM) containing the Resident Driver Firmware. (Same as P1-M)
12 to 15		Not used.
16	+12V	+12Vdc - Used by the Read Only Memory (ROM) containing the Resident Driver Firmware. (Same as P1-T)
17,18		Not used.
19	VXA	VALID EXECUTIVE ADDRESS - A high level signal, jumper selectable, generated by the DEbug module in place of the VUA signal when the system is operating in the Dual Map mode and the EXbug program is addressing the Executive portion of the memory map.

TABLE 2-2. Connector P1 Bus Interface Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
20 to 28		Not used.
29	$\overline{D1}$	DATA bus (bit 1) - Same as $\overline{D3}$ on P1-$\overline{H}$.
30	$\overline{D2}$	DATA bus (bit 2) - Same as $\overline{D3}$ on P1-$\overline{H}$.
31	$\overline{D0}$	DATA bus (bit 0) - Same as $\overline{D3}$ on P1-$\overline{H}$.
32	$\overline{D4}$	DATA bus (bit 4) - Same as $\overline{D3}$ on P1-$\overline{H}$.
33	A15	ADDRESS bus (bit 15) - Same as A14 on P1-$\overline{M}$.
34	A12	ADDRESS bus (bit 12) - Same as A14 on P1-$\overline{M}$.
35	A11	ADDRESS bus (bit 11) - Same as A14 on P1-$\overline{M}$.
36	A8	ADDRESS bus (bit 8) - Same as A14 on P1-$\overline{M}$.
37	A7	ADDRESS bus (bit 7) - Same as A14 on P1-$\overline{M}$.
38	A4	ADDRESS bus (bit 4) - Same as A14 on P1-$\overline{M}$.
39	A3	ADDRESS bus (bit 3) - Same as A14 on P1-$\overline{M}$.
40	A0	ADDRESS bus (bit 0) - Same as A14 on P1-$\overline{M}$.
41,42,43	GND	GROUND

TABLE 2-3. Connector P2 Interconnection Signals

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
1	+5V	+5 Vdc - Output voltage supplied to Disk Drive Unit for logic circuits.
3	+5V	+5 Vdc - Output voltage supplied to Disk Drive Unit for logic circuits.
5	$\overline{DIRECTION}$	DIRECTION - This signal output is used in conjunction with the STEP signal to move the recording head from track to track. When this signal is a logic low level, the recording head is moved to the lower numbered tracks (out). When this signal is a logic high level, the head moves to the higher numbered tracks (in). This signal must remain in the desired logic state during the duration of the STEP signal.
7	$\overline{STEP}$	STEP - This signal output is used in conjunction with the DIRECTION signal to move the recording head from track to track. A logic low level pulse causes the head to be moved one track (step) in the direction indicated by the DIRECTION signal.
9,11,13		Not used.

TABLE 2-3. Connector P2 Interconnection Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
15	$\overline{\text{SELECT 0}}$	SELECT 0 - The logic low level output signal used to select disk drive 0.
17	$\overline{\text{TRK 00}}$	TRACK 00 - Logic low level input signal used to indicate when the recording head is positioned over track 00. When low, the stepper motor drive circuits are inhibited to prevent further outward movement of the head.
19	$\overline{\text{HEAD LOAD}}$	HEAD LOAD - This logic low level output signal is used to position the flexible diskette against the recording head.
21	$\overline{\text{READY 0}}$	READY 0 - This logic low level input signal is used to indicate that the flexible diskette is inserted correctly into disk drive 0 and that the dc voltage levels and disk speed in this drive are correct.
23	$\overline{\text{TRK 43}}$	ABOVE TRACK 43 - This output signal is used to control the amplitude of the write current in the recording head. When recording on tracks 0 through 43, this signal must be a logic high level. Conversely, when recording on tracks 44 through 76, this signal must be a logic low level.
25	$\overline{\text{INDEX}}$	INDEX - A logic low level input pulse used to indicate the beginning of a track. This pulse occurs once per revolution of the diskette.
27	$\overline{\text{HEAD 1}}$	HEAD 1 - An optional signal, jumper selectable, used in advanced design systems to select the Drive Head.
29	$\overline{\text{WRITE ENABLED}}$	WRITE ENABLED - A logic low level input signal used to indicate that writing may take place.
31	$\overline{\text{WRITE DATA}}$	WRITE DATA - This output signal consists of logic low level pulses representing data to be recorded on the flexible diskette. Write current reverses direction on the leading edge of each pulse.
33	$\overline{\text{EN WRITE}}$	ENABLE WRITE - A logic low level output signal used to enable recording of data on the flexible diskette. When this signal is a logic high level, reading data from the flexible disk is enabled.
35	$\overline{\text{SEL 2/3}}$	SELECT 2 or 3 - An optional signal, jumper selectable, used in advanced design systems to select Drive 2 or 3.
37	$\overline{\text{RAW DATA}}$	RAW DATA - This input signal contains the unseparated data read from the flexible diskette.
39	$\overline{\text{SELECT 1}}$	SELECT 1 - The logic low output signal used to select disk drive 1.
2-40	GND	GROUND (all even numbered pins are GROUND)

CHAPTER 3

THEORY OF OPERATION

3.1 INTRODUCTION

This chapter provides a block diagram description of the M68SFDC2 Floppy Disk Controller Module. An overall block diagram of this module is shown in Figure 3-1. Separate block diagrams are presented for both the Read Data Circuit and Write Data Circuit portions of this overall diagram in Figures 3-2 and 3-3, respectively. The schematic diagram for this module is provided in Figure 3-4.

3.2 BLOCK DIAGRAM DESCRIPTION

The Floppy Disk Controller Module occupies 1032 address locations, divided as follows: ROM Resident Driver Firmware addresses E800 to EBFF, PIA addresses EC00 to EC03, and SSDA (Serial Synchronous Data Adapter) addresses EC04 to EC07. The address decoding logic decodes these addresses to generate the necessary select signals within the module to read and write data from and to the selected Disk Drive Unit. The module contains the circuits needed to control the reading and writing operations of the Disk Drive Unit, plus the necessary control bus buffers, data bus buffers, and disk drive buffers. The following paragraphs describe the major circuits used to perform these functions.

3.2.1 ROM Resident Driver Firmware (E800 to EBFF)

The Resident Driver Firmware stored in the Read Only Memory (ROM) device is used to control all of the EXORdisk II Floppy Disk System hardware operations.

3.2.1.1 Initialization. When the Floppy Disk System is initially accessed by the user, a defined initialization procedure must be used. When the Floppy Disk System is used with the EXORciser, this initialization procedure is automatically performed by the Motorola Disk Operating System (MDOS) program when the command E800;G is entered by the user. However, if the Floppy Disk System is used in conjunction with a user designed system, the user must include this initialization procedure when preparing his own Disk Operating System (DOS) program. Parameters for the initialization procedure are stored in nine sequential bytes as described in Table 3-1.

3.2.1.2 Error Messages. The ninth byte (FDSTAT) of the initialization procedure contains a hexadecimal error message (from 30_{16} to 39_{16}). If no error occurred during the disk operation, then the carry bit will be reset and the FDSTAT byte will contain the hexadecimal number 30 (ASCII 0): no error. However, if an error does occur, then the carry bit will be set and the FDSTAT byte will contain a hexadecimal number of 31 to 39 (ASCII 1 to 9) that will relate to a specific error message. If an error does occur, any disk operation in progress is halted and control will be returned to the user. Each of the error messages (and their corresponding hexadecimal and ASCII characters) is explained in Table 3-2.

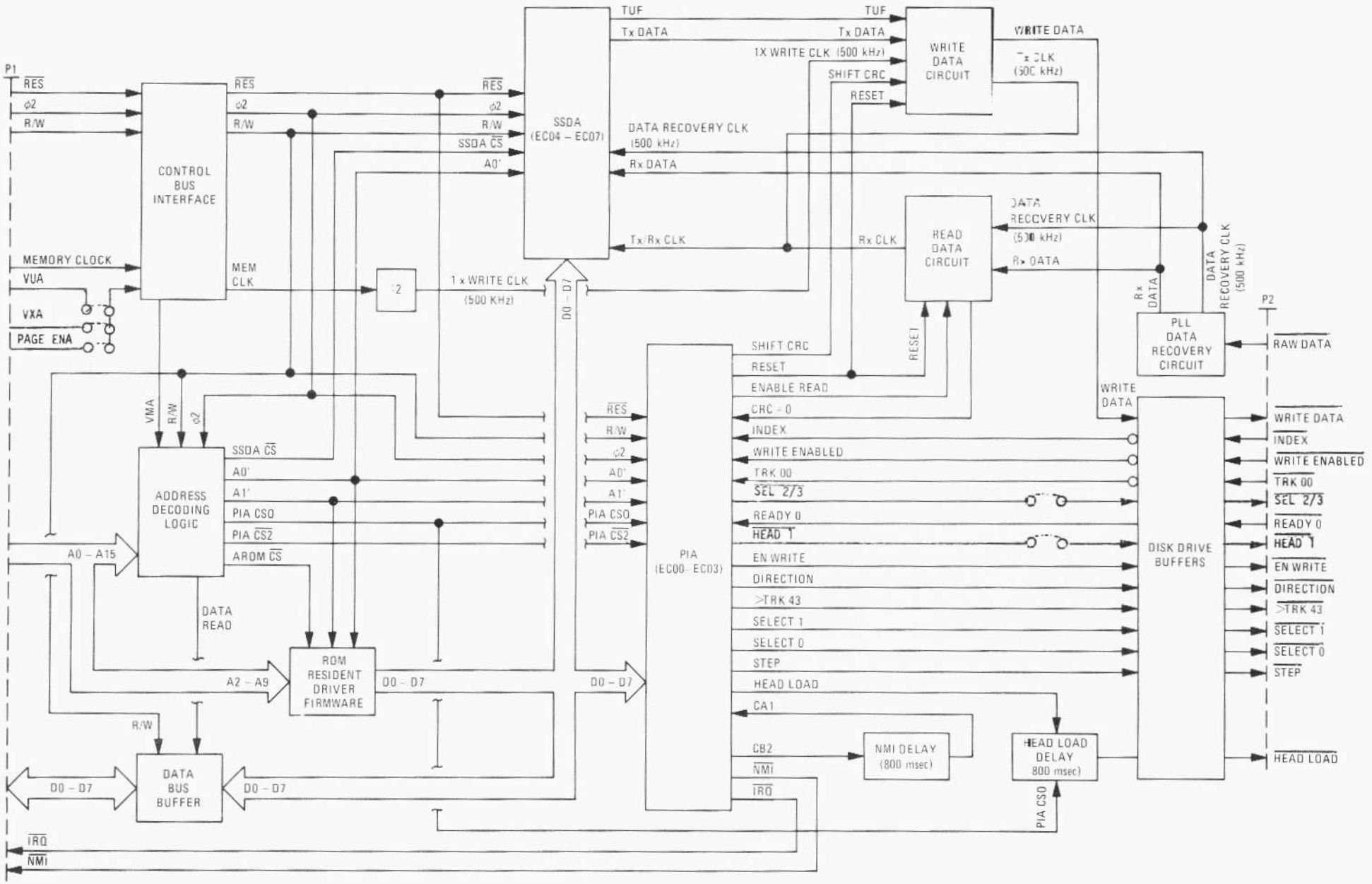


FIGURE 3-1. Floppy Disk Controller Module Block Diagram

TABLE 3-1. Initialization Parameters for a User Prepared DOS Program

BYTE	NAME	DEFINITION
0	CURDRV	CURrent DRiVe - This byte contains the number of the selected drive unit (0 or 1).
1,2	STRSCT	STaRting SeCTor - These two bytes contain the physical sector number of the first sector to be used (starting sector). For all single density disks, this starting sector number must be between 0 and 2001, inclusively.
3,4	NUMSCT	NUMber of SeCTors - These two bytes contain the number of sectors to be used. This number includes a partial sector read, if one is requested. For all single density disks, the sum of the numbers contained in the STRSCT byte and the NUMSCT byte cannot be greater than 2002 when read or write operations are requested.
5	LSCTLN	Last SeCTor LeNgth - This byte, during read into memory operations, contains the number of bytes to be read from the last sector to be used. This number should be between 1 and 128, since each sector contains 128 bytes. If this number is not between 1 and 128 (inclusively), a CRC error will result when the last sector is read. The Resident Driver Firmware may not stop on the byte specified, since the firmware segments the last sector into 8 byte blocks and stops loading after reading the block containing the last byte requested.
6,7	CURADR	CURrent ADdRess - These two bytes contain the first address from/to which data is to be read/written during disk read/write operations. The Resident Driver Firmware automatically updates this entry after each sector is read/written. During write test operations, these two bytes contain the address of a two byte data buffer.
8	FDSTAT	Floppy Disk STATus - This byte contains a status indication returned from the Resident Driver Firmware ROM. If an error occurred during a disk operation, the carry bit will be set on return to the caller, and this byte will contain a number indicating the type of error. If no error occurs, then carry will be reset and this byte will contain the hexadecimal number 30 (ASCII 0).

TABLE 3-2. Disk Error Messages

FDSTAT (HEX)	ASCII	CARRY BIT	DESCRIPTION/POSSIBLE CAUSE
30	0	RESET	NO ERRORS. This status indication is returned when no errors have occurred in the disk operation. On return to the user, the carry bit is cleared.
31	1	Set	DATA CRC ERROR. This status is returned when the CRC following the data is in error. This error would occur after the sector has been read and, if appropriate, written into memory. The CURADR byte will not be updated for the sector with the error. In multiple sector operations, the equation listed at the end of this table can be used to determine the sector number of the physical sector in which the error occurred. Possible causes of this error include miswriting and misreading the data and/or CRC. If the error occurred during WRVERF or RWTEST, the sector should be rewritten. Otherwise, another attempt should be made to read the sector. The ROM will attempt to read the sector 5 times before returning this error.
32	2	Set	DISK WRITE PROTECTED. This status is returned whenever an attempt is made to write to a diskette that is protected. In order to write to a protected diskette, a piece of opaque tape must be placed over the write protect hole in the disk's envelope.
33	3	Set	DISK NOT READY. This status is returned when an operation is attempted with a disk that is not ready. Possible causes of the not ready status include the drive unit door is not closed and the diskette is not up to speed.
34	4	Set	READ DELETED DATA MARK. This status is returned when an attempt is made to read a sector that is prefaced by a deleted data mark. The sector will not be read into memory or written onto the diskette, and the CURADR byte will not be updated for the sector in error. The equation listed at the end of this table can be used to determine the physical sector number of the sector in error when this error occurs during a multiple sector operation. A possible cause of this error is that a deleted data mark was intentionally written to the sector.
35	5	Set	TIMEOUT. This error occurs whenever a disk operation is not completed before the hardware timer on the interface board times out. This error is also produced if any non-maskable interrupt (such as an ABORT) is generated during a disk operation.

TABLE 3-2. Disk Error Messages (cont'd)

FDSTAT (HEX)	ASCII	CARRY BIT	DESCRIPTION/POSSIBLE CAUSE
35	5	Set	Other possible causes of a disk timeout error include attempting to read or write a bad track or sector or an unformatted disk. Reformatting the disk may eliminate this error. The disk timeout may also occur as a result of the head being positioned at the wrong track.
36	6	Set	INVALID DISK ADDRESS. This error occurs when the sum of the STRSCT and NUMSCT bytes are greater than the number of sectors on the disk. The RESTOR command does not check for this error.
37	7	Set	SEEK ERROR. This error occurs if a restore is completed incorrectly or track 0 is found before a seek operation is completed. This error may occur during a restore if the drive is not connected to the controller.
38	8	Set	DATA MARK ERROR. This error occurs if a valid data mark for the sector being read has not been found. This error message occurs before the sector is read, and prevents the sector from being read. The CURADR byte is not updated for the sector in error. The equation provided at the end of this table can be used to determine the number of the physical sector in error, when this error occurs during a multiple sector operation. The Resident Driver Firmware will attempt to read the sector in error 5 times before returning this error message. Possible causes of this error include misreading and miswriting the disk.
39	9	Set	ADDRESS MARK CRC ERROR. This error occurs when the CRC of an address mark is incorrect. This error occurs before the sector is read or written, and stops the operation in progress. The CURADR byte is not updated for the sector in error. The equation provided at the end of this table can be used to determine the sector number of the physical sector in error, when this error occurs during a multiple sector operation. Possible causes of this error include miswriting the address mark or its CRC when formatting the disk, and misreading the address mark or its CRC. The Resident Driver Firmware will attempt to read the sector in error 5 times before returning this error message.

EQUATION: PSNE = STRSCT + NUMSCT - SCTCNT - 1

where: PSNE is the sector number of the physical sector in error.
STRSCT is the contents of the STRSCT byte.
NUMSCT is the contents of the NUMSCT byte.
SCTCNT is the 2 byte value contained in locations $B and $C. This value is set equal to the contents of the NUMSCT byte at the beginning of a disk read or write operation, and is then decremented before each sector operation.

3.2.1.3 Resident Driver Firmware Entry Points. Various entry points (addresses) are available to the user to perform specific operations contained within the ROM's Resident Driver Firmware. These entry points are provided in Table 3-3. This table is divided into three parts: the initialization and error check routines part, the disk operation routines part, and the line printer driver routines part. In these routines, all register contents are unspecified (except where otherwise indicated) on entry to and return.

In addition, the ten entry points in the disk operation routines part are called by a Jump to Subroutine (JSR) and require the indicated parameters. The user parameter name is used to represent its contents. Upon entry to a disk routine, the user's interrupt mask is saved and the mask bit is set. The NMI vector is also saved and the Resident Driver Firmware timeout routine's entry address is substituted for the NMI vector. (Thus, an ABORT or any other NMI operation during a disk routine will appear as a timeout error.) The FDSTAT byte is then initialized to indicate that no errors occurred. Before returning from a disk routine, the NMI vector and the user's interrupt mask are restored. If an error occurred, the FDSTAT byte will be changed accordingly, and the carry bit will be set.

TABLE 3-3. Resident Driver Firmware Program Entry Points

NAME	LOCATION	FUNCTION
		INITIALIZATION AND ERROR CHECK ROUTINES
OSLOAD	E800	This entry bootloads the disk operating system, initializes the stack pointer and drive electronics, and restores the head position of drive 0 to track 0. The bootloader and the operating system's retrieval information block from sectors 23_{10} and 24_{10}, respectively, of drive 0 are loaded into memory beginning at location 32_{10}. Control is then passed to the bootloader by jumping to that location. If a disk error occurs, the error number is printed at the system console and control is returned to EXbug. No user parameters need to be specified when the Resident Driver Firmware is entered at this entry point. The firmware will initialize all the required parameters.
FDINIT	E822	This subroutine initializes the PIA and SSDA. No user parameters are required by this subroutine and none are modified by it. This subroutine does not change location FDSTAT or the state of the carry bit.
CHKERR	E853	This subroutine checks for a disk error if called immediately after return from a disk operation by checking the carry flag. The subroutine just returns to the user if no error occurred (carry clear). If an error did occur (carry set), then the subroutine prints an E followed by the contents of FDSTAT (in ASCII) and two spaces at the system console. It then gives control to EXbug. Other than FDSTAT, no user parameters are required. (If a disk error occurred, the Resident Driver Firmware will load the appropriate data into FDSTAT). CHKERR does not modify any user parameters.

TABLE 3-3. Resident Driver Firmware Program Entry Points (cont'd)

NAME	LOCATION	FUNCTION
		INITIALIZATION AND ERROR CHECK ROUTINES (cont'd)
PRNTER	E85A	This subroutine prints an E at the system console followed by the contents of FDSTAT (in ASCII) and two spaces. FDSTAT is the only user parameter required by PRNTER. It does not modify any user parameters.
		DISK OPERATION ROUTINES
READSC	E869	This entry causes the number of sectors (beginning with STRSCT of CURDRV) to be read from the NUMSCT byte into memory beginning at CURADR. The CURADR byte is updated to the next address to be written to after each sector is read. This entry point initializes the LSCTLN byte to 128 (decimal) so that all of the last sector read will be written to memory. This routine does not change CURDRV, STRSCT, or NUMSCT.
READPS	E86D	This entry causes the number of sectors (beginning with STRSCT of CURDRV) to be read into memory beginning at CURADR. The CURADR byte is updated to the next address to be written to after each sector is read. This entry point does not change LSCTLN, so that only a portion of the last sector read may be written to memory. (See LSCTLN under user parameters for a description of last sector segmentation and valid values of LSCTLN.) This routine does not change CURDRV, STRSCT, NUMSCT, or LSCTLN.
RDCRC	E86F	This entry causes NUMSCT sectors beginning with STRSCT of CURDRV to be read to check their CRC's. The sectors are not written to memory. This routine does not change CURDRV, STRSCT, NUMSCT, LSCTLN, or CURADR.
RWTEST	E872	This entry point causes the data at CURADR and CURADR+1 to be written to alternating bytes, respectively, of NUMSCT sectors beginning with STRSCT of CURDRV. After all of the sectors have been written, they are read back to check their CRC's. This routine does not change CURDRV, STRSCT, NUMSCT, LSCTLN, or CURADR.
RESTOR	E875	This entry point causes the head of CURDRV to be restored to track 0. The head is unloaded from the disk. The drive must be ready or restore will return an error. RESTOR does not verify that STRSCT and NUMSCT are valid. RESTOR is used to position the drive's head at a known track before using the drive. (The OSLOAD routine restores drive 0.) RESTOR does not change CURDRV, STRSCT, NUMSCT, LSCTLN, or CURADR.

TABLE 3-3. Resident Driver Firmware Program Entry Points (cont'd)

NAME	LOCATION	FUNCTION
		DISK OPERATION ROUTINES (cont'd)
SEEK	E878	This entry point causes the head of CURDRV to be positioned at the track containing STRSCT. The head is loaded on the disk. This routine does not change CURDRV, STRSCT, NUMSCT, LSCTLN, or CURADR.
WRTEST	E87B	This entry point causes the two bytes of data pointed to by the address in CURADR and the address + 1 to be written to alternating bytes, respectively, of NUMSCT sectors beginning with STRSCT of CURDRV. This routine does not change CURDRV, STRSCT, NUMSCT, LSCTLN, or CURADR.
WRDDAM	E87E	This entry point causes a deleted data address mark to be written to NUMSCT sectors beginning with STRSCT of CURDRV. This routine does not change CURDRV, STRSCT, NUMSCT, LSCTLN, or CURADR.
WRVERF	E881	This entry point causes NUMSCT sectors beginning with STRSCT of CURDRV to be written from memory beginning at CURADR. CURADR is updated to the address of the next byte to be read from after each sector has been written. After all of the sectors have been written, they are read back and their CRC's are verified as in RDCRC. This routine does not change CURDRV, STRSCT, NUMSCT, or LSCTLN.
WRITSC	E884	This entry point causes NUMSCT sectors beginning with STRSCT of CURDRV to be written from memory beginning at CURADR. CURADR is updated to the address of the next byte to be read from after each sector has been written. This routine does not change CURDRV, STRSCT, NUMSCT, or LSCTLN.
CLOCK	E887	(PROM version Cal 1.2 only) This entry point calculates a timing parameter as a function of the system clock frequency for the PROM routines. This is done automatically if the system is started at the OSLOAD entry point. If OSLOAD is not used to bootload the system, then this routine must be called after FDINIT has been called and before any other disk operation is attempted. If CLOCK returns an error, no further disk operations should be attempted. This routine only requires the CURDRV parameter. It does not change CURDRV, STRSCT, NUMSCT, CURADR, or LSCTLN.

TABLE 3-3. Resident Driver Firmware Program Entry Points (cont'd)

NAME	LOCATION	FUNCTION
		LINE PRINTER DRIVER ROUTINES
LPINIT	EBC0	This subroutine initializes the PIA interfacing with the line printer from a reset condition.
LIST	EBCC	This subroutine sends the contents of the A accumulator to the line printer. If a printer error occured, carry is set on return to the caller. The LIST routine detects the paper empty and printer not selected conditions as printer errors.
LDATA	EBE4	This subroutine sends a character string pointed to by the index register and terminated by a 04 to the line printer. Prior to the string, it sends a carriage return, then a line feed, to the printer. LDATA uses the LIST subroutine. Thus, any printer errors that occur while using LDATA will be detected by LIST. LDATA will sit in a loop until aborted or the error is corrected.
LDATA1	EBF2	This subroutine performs the same as LDATA except that this subroutine does not print a carriage return and line feed prior to the string. It also uses LIST to send characters to the printer.

3.2.2 Recording Format

The format of the data recorded on the diskette is a function of the Floppy Disk Controller Module and is compatible with the IBM 3740 format. Data is recorded on the diskette using frequency modulation techniques (each data bit recorded on the diskette has an associated clock bit recorded with it). These clock and data bits (if present) are interleaved. By definition, a Bit Cell is the period between the leading edge of one clock bit and the leading edge of the next clock bit. (Thus, the Bit Cell is one clock bit and one data bit if the data bit is present.)

When referring to serial data being written onto or read from the Disk Drive Unit, a byte is defined as eight consecutive Bit Cells. The most significant Bit Cell is defined as Bit Cell 0, and the least significant Bit Cell is defined as Bit Cell 7. During a write operation, Bit Cell 0 of each byte is transferred to the diskette first with Bit Cell 7 being transferred last. Correspondingly, the most significant byte of data is transferred first, while the least significant byte is transferred last. During read operation, Bit Cell 0 of each byte will be read from the diskette first with Bit Cell 7 last. As with writing, the most significant byte will also be read from the diskette first.

The EXORdisk II is capable of recording up to 77 tracks of data. The tracks are numbered 0 to 76. Each track is made available to the recording head by moving the head with a stepper motor and carriage assembly controlled by the PIA located on the Floppy Disk Controller Module. The diskette is rotated by the drive motor at a speed of 360 rpm.

When the Motorola Disk Operating System (MDOS) is used (EXORciser only), the 77 tracks are formatted in the following manner: the first track (Track 0) contains the disk directory, and the remaining 76 tracks contain data information. Each track is divided into 26 sectors containing 128 bytes each. Preceding each sector is a 6 byte ID Field. This ID Field is formatted as follows:

```
Byte 1   - Track #
Byte 2   - Blank
Byte 3   - Sector #
Byte 4   - Blank
Byte 5&6 - Cyclic Redundancy Code (CRC)
```

Each field of data recorded on a track is separated from adjacent fields by a number of bytes containing no data bits. These areas are referred to as gaps and are provided to allow updating of one field without affecting adjacent fields. Four different types of gaps are used: Post-Index Gap, ID Gap, Data Gap, and Pre-Index Gap. The Post-Index Gap is defined as the 32 bytes between Index Address Mark and the ID Address Mark for sector 1 (excluding the address mark bytes). This gap is always 32 bytes in length and is not affected by any updating process. The ID Gap consists of 17 bytes between the ID Field and the Data Field. This gap may vary in size slightly after the Data Field has been updated. The Data Gap consists of 33 bytes between the Data Field and the next ID Field. This gap may also vary slightly after the Data Field has been updated. The Pre-Index Gap is a space of 320 bytes between the last data field on the track and the Index Address Mark. This gap may also vary slightly in length.

Address Marks are unique bit patterns, one byte in length, and used to identify the beginning of ID and Data Fields and to synchronize the deserializing circuitry with the first byte of each field. Address Mark bytes are unique from all other data bytes in that certain Bit Cells do not contain a clock bit (all other data bytes have clock bits in every Bit Cell). Four different types of Address Marks are used to identify the different types of fields: Index Address Mark, ID Address Mark, Data Address Mark, and Deleted Data Address Mark. The Index Address Mark is located at the beginning of each track and is a fixed number of bytes in front of the first record. The ID Address Mark is located at the beginning of each ID Field on the diskette. The Data Address Mark is located at the beginning of each nondeleted Data Field on the diskette. The Deleted Data Address Mark is located at the beginning of each deleted Data Field on the diskette.

Each field written on the diskette is appended with two Cyclic Redundancy Check (CRC) bytes. These two CRC bytes are generated from a cyclic permutation of the data bits starting with bit zero of the address mark and ending with bit 7 of the last byte within a field (excluding the CRC bytes). When a field is read back from a diskette, the data bits (from bit zero of the address mark to bit 7 of the second CRC byte) are divided by the CRC Check circuit located on the Floppy Disk Controller Module. A non-zero remainder indicates an error within the data read back from the disk, while a remainder of zero indicates that the data has been read correctly.

3.2.3 Read Operation

The SSDA is used to synchronize read operations by testing the incoming data stream for the first half clock and data pattern of the desired address mark. (The incoming data stream is clocked at 500 KHz - 2 X Read Clock signal.) When a match is found, the external circuitry is released by the Syn Match (SM) output signal, and the second half of the address mark (clock and data) is read by the SSDA and tested for a match with the desired type. If a match does not occur, the sequence is restarted. However, if the second half of the address mark does match, then the desired data transfer is initiated. Circuitry within the Read Data Circuit switches the SSDA read clock (R X CLK) to 250 KHz after the second half of the address mark has been received, permitting only the data portion of the remaining information (R X D) to be recovered. This circuitry also controls the Cyclic Redundancy Code (CRC) generator timing so that only the data portion of the recovered information is clocked into the generator.

After the data block has been transferred, the CRC status (CRC=0) is made available to the MPU for 32 microseconds at PIA peripheral line PB7.

The following paragraphs provide an operational description of the circuitry used to read data from the diskette. During this description, refer to the block diagram provided in Figure 3-2 and the schematic diagram provided in Figure 3-4.

3.2.3.1 PLL Data Recovery Circuit. The Phase Locked Loop Data Recovery Circuit consists of a frequency/phase detector circuit with filter network, a voltage controlled oscillator (VCO), two synchronous 4-bit counters, an 8-bit shift register, and four flip-flops (refer to Figure 3-4). The center frequency of the VCO is nominally 8.0 MHz with a lock range of 7.6 MHz to 8.4 MHz. Lockup time will be less than 192 microseconds.

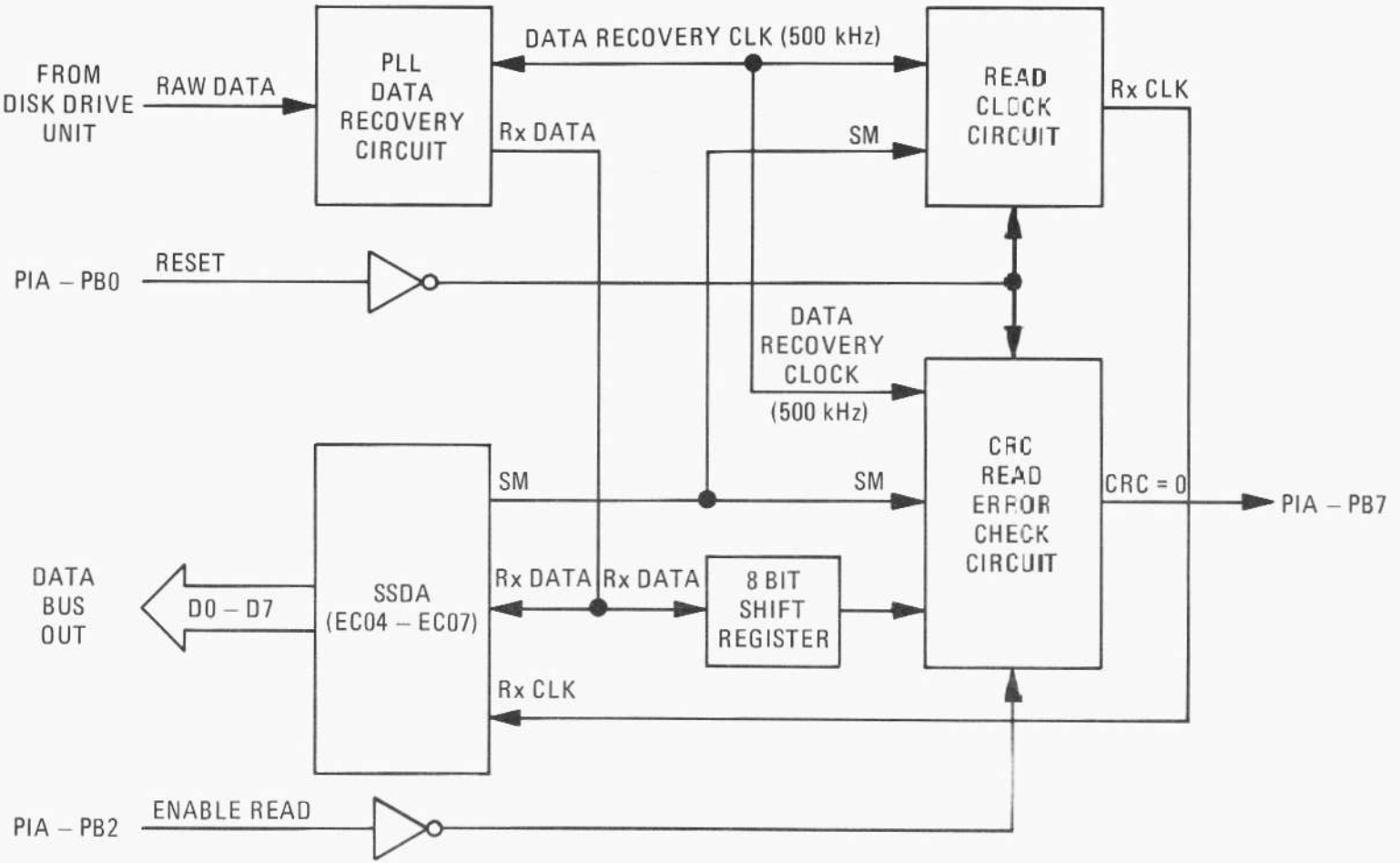


FIGURE 3-2. Read Data Circuit Block Diagram

The Raw Data signal from the Disk Drive Unit is a combination of a 500 KHz clock signal and data. This signal is applied to an input flip-flop circuit consisting of two serially connected flip-flops. This circuit then generates a negative output pulse with a pulse width equal to one VCO time period (approximately 0.125 microseconds). This negative pulse is used to preset the first reference counter with a 9 and to set the output of the first data flip-flop to a logic high level.

In the data format used in the disk system (IBM 3740 format), the incoming data stream can have only one consecutive pulse missing. By loading the first reference counter with a 9, it will produce a positive output transition within 15 VCO pulse periods (1.875 microseconds), thus generating a clock edge even if the data pulse is missing. This reference counter will also produce a carry-out pulse on the 16th VCO pulse (2 microseconds). This pulse is then compared with the carryout output of the second reference counter by the frequency/phase detector, thus providing a reference for the VCO.

The negative output transitions of the first reference counter are inverted and used to clock the first data flip-flop, causing the output to change to a logic low level. If another data pulse is present in the incoming data stream, then the first data flip-flop is once again set by the input flip-flop circuit. However, if no data is present, then the output of the first data flip-flop will remain at a logic low level until set by a data pulse, which must occur within 32 microseconds of the last data pulse in order to avoid initiating an error message. When the data pulse is set and the first data flip-flop is set, the next output pulse produced by the reference counter causes the second data flip-flop to toggle, producing the NRZ (Non-Return to Zero) data (R X D) required by the SSDA.

3.2.3.2 CRC Read Error Check Circuit. Separated data and the 500 KHz Data Recovery Clock are provided to the CRC Read Error Circuit from the PLL Data Recovery Circuit. When the second data flip-flop in the PLL Data Recovery Circuit produces the R X D data signal for the SSDA, the R X D data signal is also applied to the D input of the 8-bit shift register. This shift register delays the R X D data pulse by 16 microseconds and then applies it to the CRC Read Error Check Circuit.

When the SSDA sync code match occurs, the SSDA outputs a high level SM signal. The first positive edge of the 500 KHz Data Recovery Clock to occur after sync match sets a sync match latch. This enables both a frequency divider ($\div$ 16) and the CRC. The frequency divider produces two different output frequencies: 250 KHz ($\div$ 2) and 31.25 KHz ($\div$ 16). The 250 KHz output is then applied to the CRC.

At the 250 KHz clock rate, only the data bits from the read data are loaded into the CRC. This data, however, is delayed eight bits (four data bits) behind the read data by the 8-bit shift register. This permits the CRC to receive the first half of the address mark which occurs just before the sync match and before the CRC is enabled. The first half of the address is included in the cyclic permutation of data bits which generate the two CRC bytes. Two CRC bytes append every ID and data field.

If the complete address mark and ID or data field has been read correctly, the CRC=0 output will go low after the last CRC byte for that field has been read. The positive transition of the next 31.25 KHz output will reset the CRC=0 output signal to a logic low level, where it will remain until changed to a logic high level by the next low to high transition of the 250 KHz clock pulse. This occurs one byte time after the last byte of the data field. The software test for a CRC error is made during this one byte period. (The CRC=0 output is detected by the software via PB7 of the PIA.) If a detectable read error occurs, the CRC=0 output will remain high during the one byte test time.

After completing a CRC check of a single ID or data field, the CRC read error logic is reinitialized before reading the next field by the reset output of the PIA (pin PB0).

3.2.4 Write Operation

The write sequence is described in the following paragraphs. During this description, refer to the block diagram provided in Figure 3-3 and the schematic diagram provided in Figure 3-4.

The Transmitter Underflow (TUF) output of the SSDA is used to synchronize write operations by resetting the frequency divider ($\div$ 16) while writing the pre-address mark gap from the CRC at a 500 KHz rate provided by the 1 X Write Clock. After counting 11 TUF pulses and 5.5 bytes of pre-address gap, the first half of the desired address mark is stored in the SSDA.

When the first half of the address mark is completed, no TUF output pulse will be produced by the SSDA. The second half of the address mark may then be loaded, followed by the data to be written on the diskette. The switch clock rate flip-flop is then toggled after the address mark is written, causing the data selector to select a 250 KHz clock signal. This clocks the data portion of the information to be written into the CRC. When the data transfer is completed, two dummy bytes are stored in the SSDA. AT the same time, the frame check sequence is appended by the CRC on command of the Shift CRC signal from the PIA. A register in the CRC will then be loaded with a postamble (written at the 250 KHz clock rate) after the frame clock sequence has been appended and the first TUF pulse has occurred. On the next TUF pulse (second), the SSDA clock is switched back to 500 KHz, permitting the data to be written to the diskette.

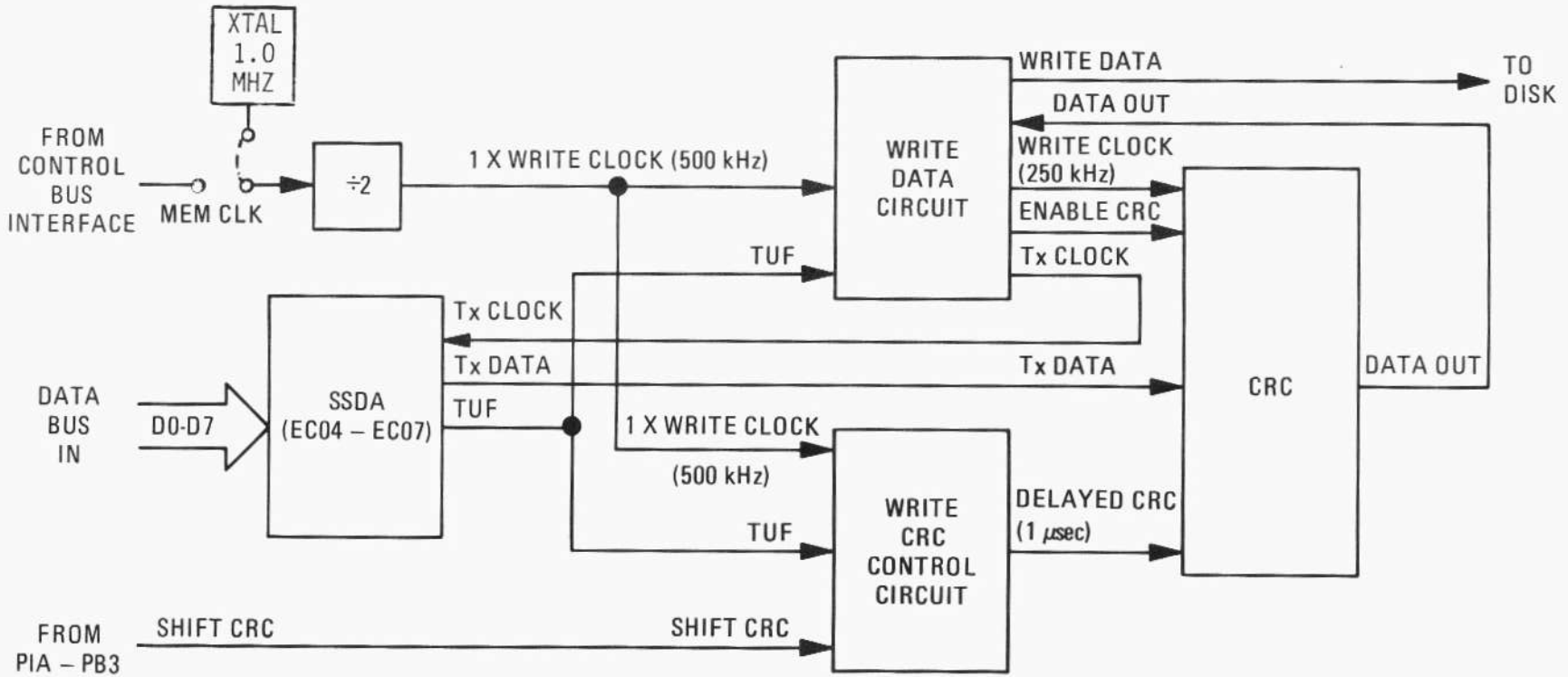
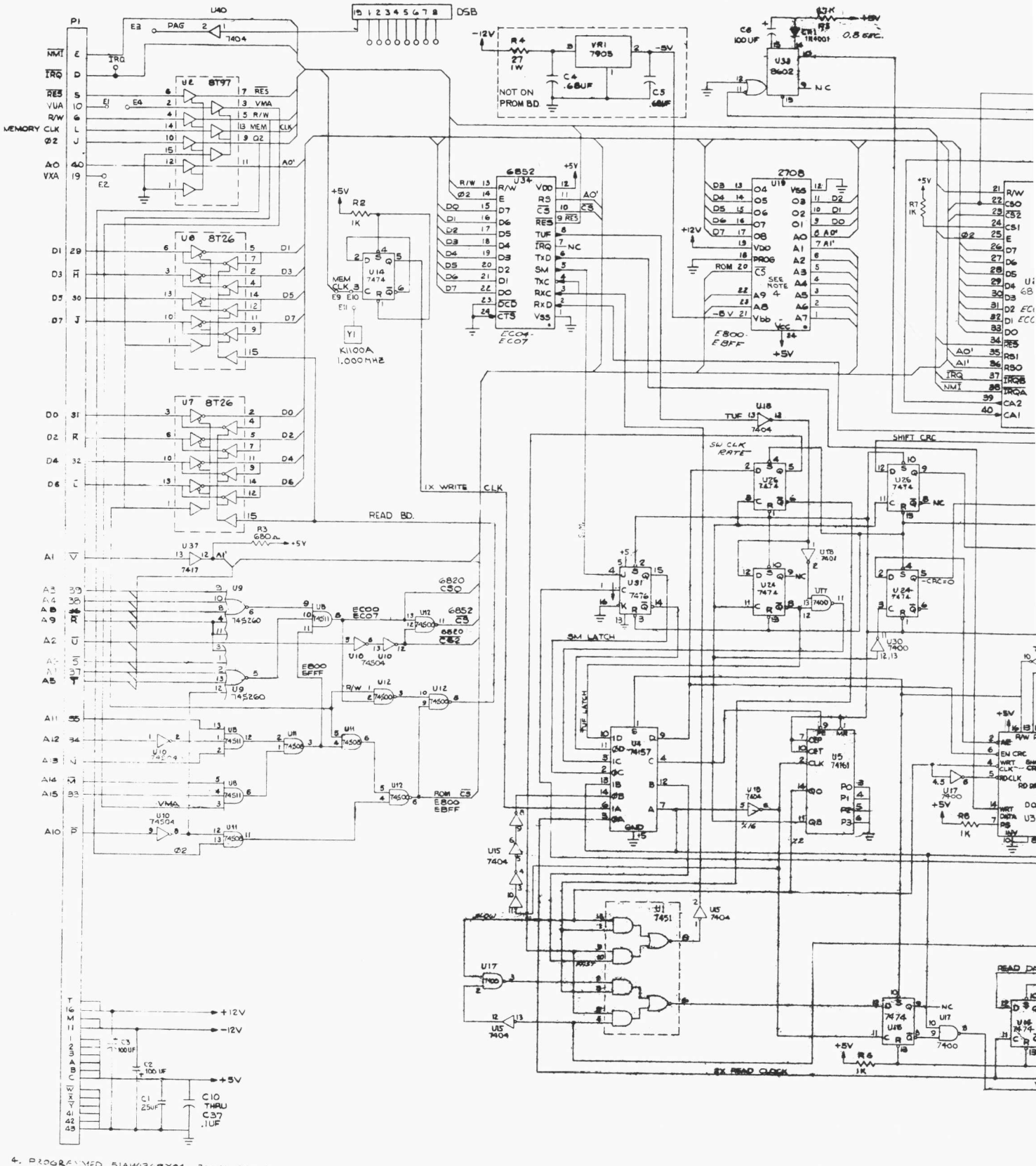


FIGURE 3-3. Write Data Circuit Block Diagram



4. PROGRAMMED 51AW1369X04 2MHz PROM.
3. C1, C3, C4, C5, R4 AND VR1 ARE OPTIONAL
2. ALL RESISTORS ARE IN OHMS ± 10 PERCENT, 1/4 WATT
   ALL CAPACITORS ARE IN UF.
1. FOR ASSEMBLY DRAWING REFER TO: 01EW122IX01

EXCEPT AS NOTED:

FIGURE 3-4. Floppy Disk Controller Module

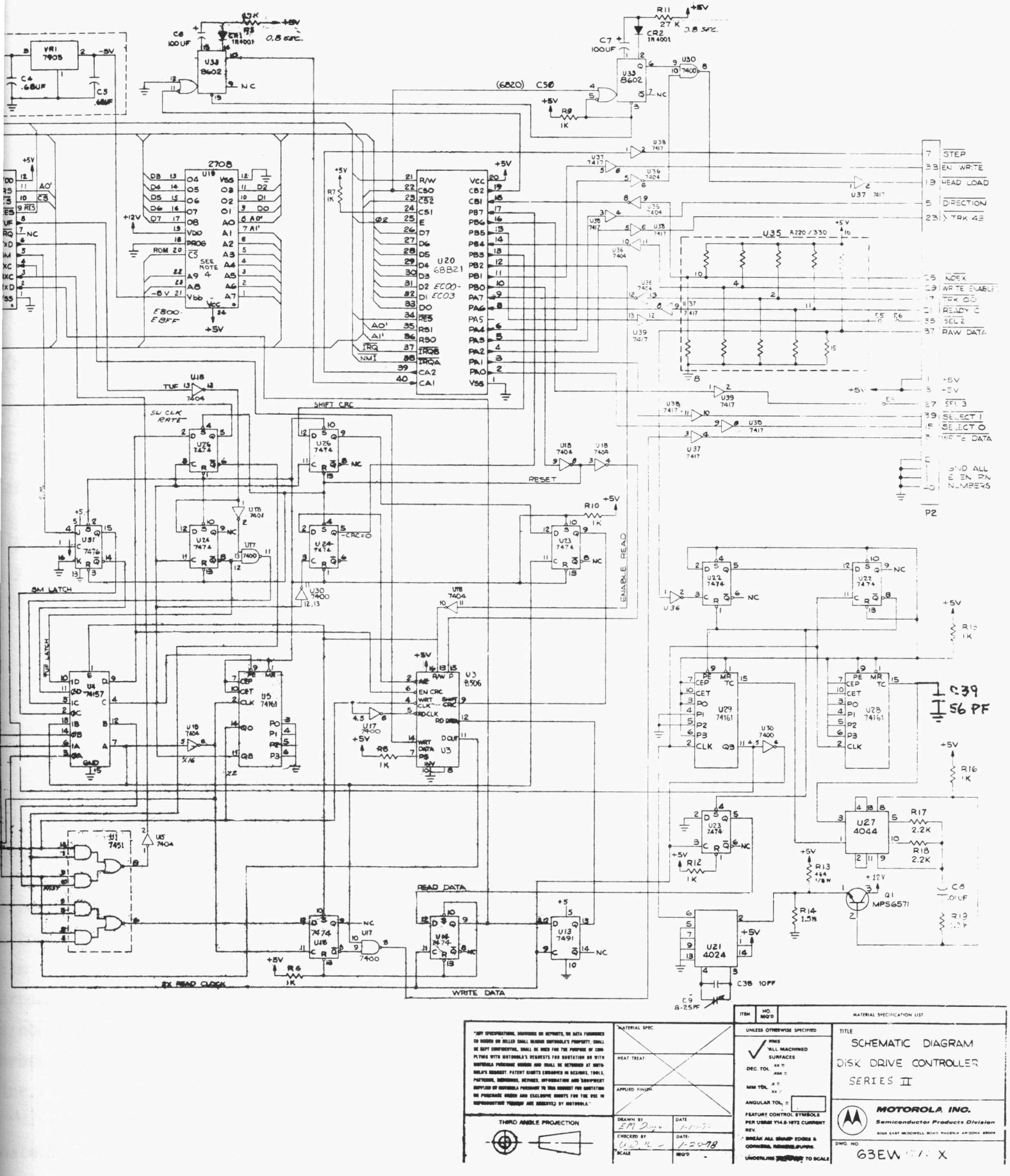


FIGURE 3-4. Floppy Disk Controller Module Schematic Diagram

CHAPTER 4

PARTS

4.1 INTRODUCTION

This chapter provides the parts lists for the Floppy Disk Controller Module (M68SFDC2), Table 4-1, and Interconnecting Cable Assembly (M68SFD1C), Table 4-2. Each of these parts lists reflects the latest issue of the hardware at the time of printing.

TABLE 4-1. Floppy Disk Controller Module (M68SFDC2) Parts List

REFERENCE DESIGNATOR	MOTOROLA PART NUMBER	DESCRIPTION	EFFEC-TIVITY
	84EW6712X01	Printed Circuit Board, Floppy Disk Controller	M
	55NW9403A10	Ejector, Circuit Card with Roll Pin Attachment (2 required)	M
	23NW9618A33	Capacitor, Electrolytic - 25MF @ 16WVDC	M
C6, C7	23NW9618A09	Capacitor, Electrolytic - 100MF @ 16WVDC	M
C8	08NW9621A03	Capacitor, Poly Film - 0.1MF @ 100WVDC	M
C9	20NW9628A02	Capacitor, Ceramic Trim - 8 to 25PF @ 350WVDC	M
C10-C37	21NW9702A09	Capacitor, Ceramic - 0.1MF @ 50WVDC	M
C38	21NW9605A20	Capacitor, Fixed Mica - 10PF @ 500WVDC	M
C39	21NW9605A11	Capacitor, Fixed Mica - 56PF @ 100 WVDC	M
CR1, CR2	48NW9607A01	Diode - 1N4001	M
P2	28NW9802A35	Connector, Right Angle Solder Tail - 40 pin	M
Q1	48NW9610A21	Transistor - MPS6571	M
R2,R6-R10, R12,R15,R16	06SW-124A49	Resistor, Fixed Carbon - 1KΩ, 5%, ¼W	M
R3	06SW-124A45	Resistor, Fixed Carbon - 680Ω, 5%, 1/4W	M
R5, R11	06SW-124A83	Resistor, Fixed Carbon - 27KΩ, 5%, 1/4W	M
R13	06NW9602A23	Resistor, Fixed Film - 464Ω, 1%, 1/8W	M
R14	06SW-124A53	Resistor, Fixed Carbon - 1.5KΩ, 5%, 1/4W	M
R17,R18	06SW-124A57	Resistor, Fixed Carbon - 2.2KΩ, 5%, 1/4W	M
R19	06SW-124A55	Resistor, Fixed Carbon - 1.8KΩ, 5%, 1/4W	M
-	28NW9802B08	Socket, Dual In-Line I.C. - 24 pin (2 required)	M
U1	51NW9615A44	I.C. - SN7451N	M
U2	51NW9615B71	I.C. - 8T95 Alternate Device	M
U3	51NW9615C79	I.C. - MC8506L	M
U4	51NW9615C78	I.C. - SN74157N	M

TABLE 4-1. Floppy Disk Controller Module (M68SFDC2) Parts List (cont'd)

REFERENCE DESIGNATOR	MOTOROLA PART NUMBER	DESCRIPTION	EFFEC-TIVITY
U5,U28,U29	51NW9615B17	I.C. - SN74161N	M
U6, U7	51NW9615F19	I.C. - 8T26A	M
U8	51NW9615D90	I.C. - SN74S11N	M
U9	51NW9615E67	I.C. - SN74S260N	M
U10	51NW9615C96	I.C. - SN74S04N	M
U15,U18, U36,U40	51NW9615A35	I.C. - MC7404P	M
U11	51NW9615C56	I.C. - SN74S08N	M
U12	51NW9615C94	I.C. - SN74S00N	M
U17, U30	51NW9615A32	I.C. - MC7400P	M
U13	51NW9615C78	I.C. - MC7491AP	M
U14,U16 U22-U24,U26	51NW9615A47	I.C. - SN7474N	M
U19	51AW1369X04	I.C. - Programmed 2 MHz	M
U19	51NW9615D15	I.C. - 82S2708	M
U20	51NW9615D85	I.C. - MC68B21P	M
U21	51NW9615A28	I.C. - MC4024P	M
U27	51NW9615B31	I.C. - MC4044P	M
U31	51NW9615C77	I.C. - MC7476P	M
U33	51NW9615B39	I.C. - MC8602P	M
U34	51NW9615F20	I.C. - MC68B52P	M
U35	51NW9626A20	Resistor Network - 220/330 , 16 pin	M
U37,U38,U39	51NW9615C75	I.C. - SN7417N	M
Y1	48NW9606A04	Crystal Oscillator 1 MHz, K1100A	M
	28NW9802C12	Header, Double Row Post, 20 pin	M
	28NW9802B21	Header, Double Row Post, 6 pin	M
	29NW9805A91	Jumper, 2 position	M
	28NW9802B07	Socket DIL-16 pin (use at U28,U29)	M

TABLE 4-2. EXORdisk II to EXORciser Interconnecting Cable Assembly (M68SFD1C) Parts List

REFERENCE DESIGNATOR	MOTOROLA PART NUMBER	DESCRIPTION	EFFEC-TIVITY
-	28NW9802A36	Connector, 40 Pin PCB	M
-	30NW9302A20	Cable, Flat 40 Conductor - 45", 28 AWG, Stranded	M
-	43NW9002A53	Insert, Key Polarizing	

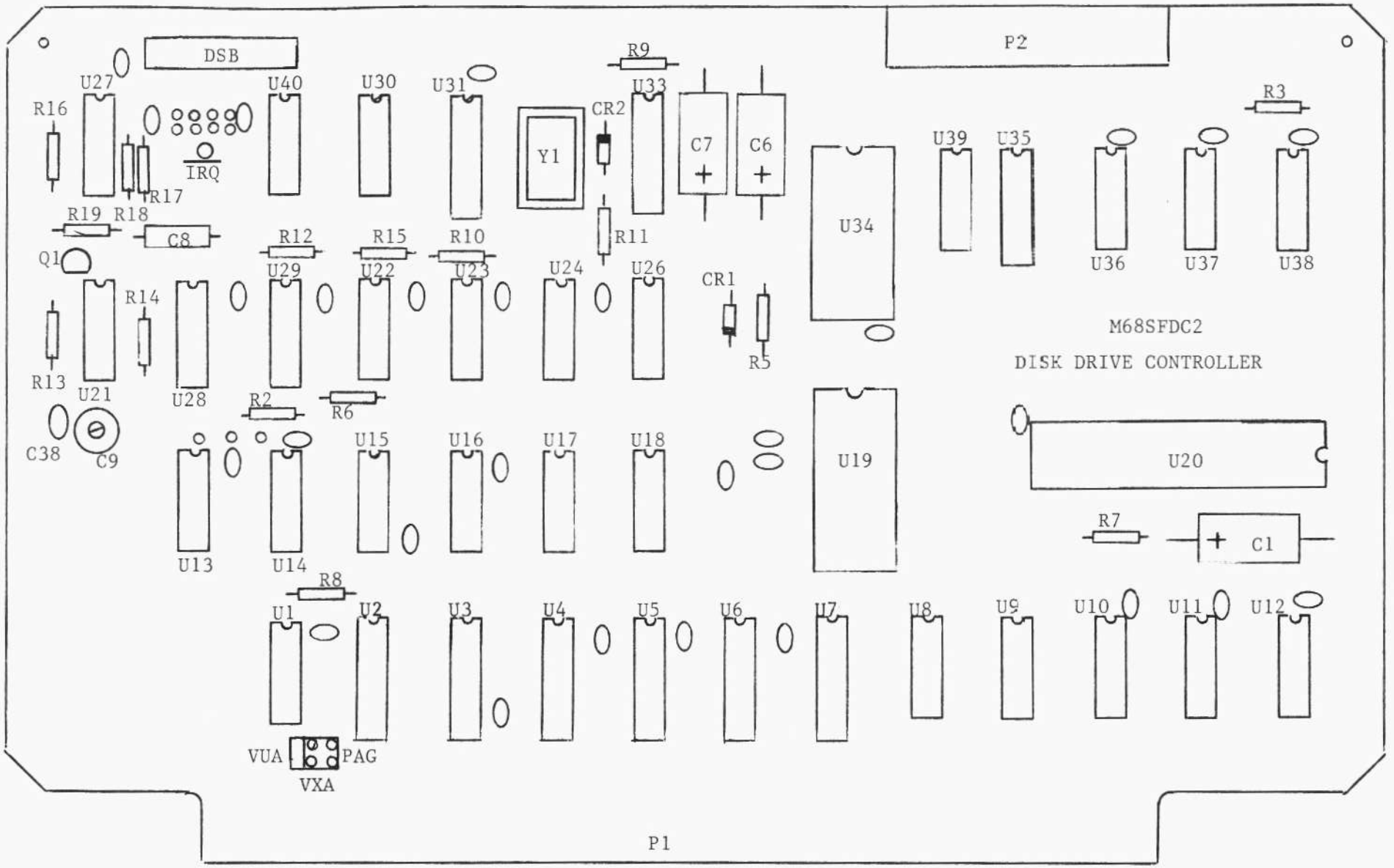


FIGURE 4-1. Floppy Disk Controller Module, Parts Location

MOTOROLA INC.
Integrated Circuits Division

Microsystems • 3102 North 56th Street • Phoenix, Arizona 85018

10946-1 PRINTED IN USA (6/78) MPS 2M