CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

FIGURES

FIGURES (Cont)

FIGURES (Cont)

TABLES

PREFACE

This manual is intended for use by service personnel who have previously been trained in the KL10 and its associated core memory subsystems (MA20, MB20). It assumes the reader is familiar with the KL10 architecture, SBus protocol, and use of the KLAD diagnostic system to load and run diagnostics. It also assumes familiarity with running SYSERR and user mode diagnostics under control of a TOPS-10 and TOPS-20 operating system. Although formal MF20/MG20 training is desirable, the experienced DECsystem-10/DECSYSTEM-20 service engineer should be able to use this manual to service the MF20 and MG20 without special training.

This manual contains ten chapters and three appendices as described in the following paragraphs.

Chapter 1 - Introduction
This chapter provides an overview of the MF20 MOS Memory Subsystem. It includes a general description, a summary of the technical specifications, and references to other related documentation. This chapter answers the question: What is an MF20 and how does it tie into a KL10-based DECSYSTEM-20?

Chapter 2 - Site Preparation and Planning
This chapter contains a detailed physical description of all the major components and subassemblies that comprise the MF20. The information presented is primarily used for site planning and preparation.

Chapter 3 - Installation Procedures
This chapter describes the detailed step-by-step procedures for installing an MF20 option either as a new system installation or as an add-on to an existing configuration. It includes the required checkout procedures to be performed after installing the unit.

Chapter 4 - Operation Programming
This chapter describes the steps required to place the MF20 in a state where it can be placed on-line, ready to be used by the operating system. Unlike previous KL10-based memory systems, the MF20 does require a moderate amount of program intervention before it can be used to store and retrieve data.

Chapter 5 - Technical Description
This chapter includes a functional description of each major area within the MF20 subsystem. The theory of operation is discussed using block diagrams, tables, and simplified logic diagrams to emphasize how the functional areas of logic interact over the backplane. This treatment is consistent with the prime maintenance philosophy - fault isolation to the module level.

Chapter 6 - Preventive Maintenance
This chapter describes the recommended preventive maintenance procedures and schedules.

Chapter 7 - Corrective Maintenance
This chapter describes the use of hardware and software tools to repair a failing MF20. It emphasizes the use of both system and diagnostic software to perform fault isolation. It also includes descriptions of any required removal, replacement, and adjustment procedures.

Chapter 8 - MF20F Site Preparation and Installation
This chapter describes the external MOS memory cabinet layout. It also includes the mechanical, power, and environmental data sheet, service area requirements, and option variations for the MF20F. The installation procedures are also outlined in this chapter.

Chapter 9 - MG20 Specific Information
This chapter provides an overview of the MG20 internal memory and external cabinet. Site preparation and installation procedures are described.

Chapter 10 - TGHA (The Great Heuristic Algorithm)
This chapter explains the use of TGHA with the MF20 and MG20 MOS memories. Examples are provided for both memories.

Appendix A - M8579 Storage Module Geography
This appendix describes the physical layout of the 16K MOS chips on the storage module and how to determine chip position if the physical address is known.

Appendix B - Complex IC Descriptions
This appendix describes the operation of the data multiplexer and MOS RAM chips used on the M8579 storage modules.

Appendix C - 11-Based 10 Diagnostic Test Structure
This appendix briefly describes the basic operations involved with using an 11-based program to test a KL10-based hardware subsystem.

CHAPTER 1
INTRODUCTION

1.1 GENERAL DESCRIPTION

The MF20/MF20F is a MOS Memory Subsystem for use in KL10-based computer systems. The MF20F is an external cabinet. It can provide additional storage for up to 3,145,728 44-bit words in increments of 256K (262,144) words. The MF20 may be installed as the only primary storage resource on a Decsystem-10/DECSYSTEM-20, or it can be added on to supplement existing core memory systems using MA20/MB20 memories. The MF20F may be installed on a KL10-E or KL10-R.

1.1.1 Key Features

Large Capacity - Up to 1.5 megawords can exist on the rear door of a DECsystem-10/DECSYSTEM-20.

High Reliability - Automatic error detection and correction for single-bit errors (SBEs) is coupled with a spare bit substitution mechanism.

Increased MTBF - A sophisticated set of diagnostic functions permits software to continue running by patching out hardware faults. This makes it possible to defer corrective maintenance.

Reduced MTTR - Diagnostic hardware permits the diagnostic programs to access detailed control and data path signals. This provides excellent fault resolution to aid the service engineer.

On-Line Diagnosis - Newly developed software coupled with modifications to existing software permits continual on-line error logging and analysis to provide early warning of any deterioration in the MF20's performance.

Battery Backup - A battery backup power supply system provides adequate dc power to the MF20 to preserve data integrity for 30 seconds. This is sufficient time to allow switchover to an uninterruptable power source (UPS) in the event of a primary power outage.

1.1.2 Physical Components (Figure 1-1)

The basic MF20 subsystem consists of several components which combine to provide a variety of memory configurations.

1.1.2.1 MF20 Logic Assembly (Figure 1-2) - This unit houses all the MF20/MF20F logic modules and consists of the following subassemblies.

1. Mounting box complete with printed circuit backplane that can accommodate up to sixteen extended-width, hex-height modules

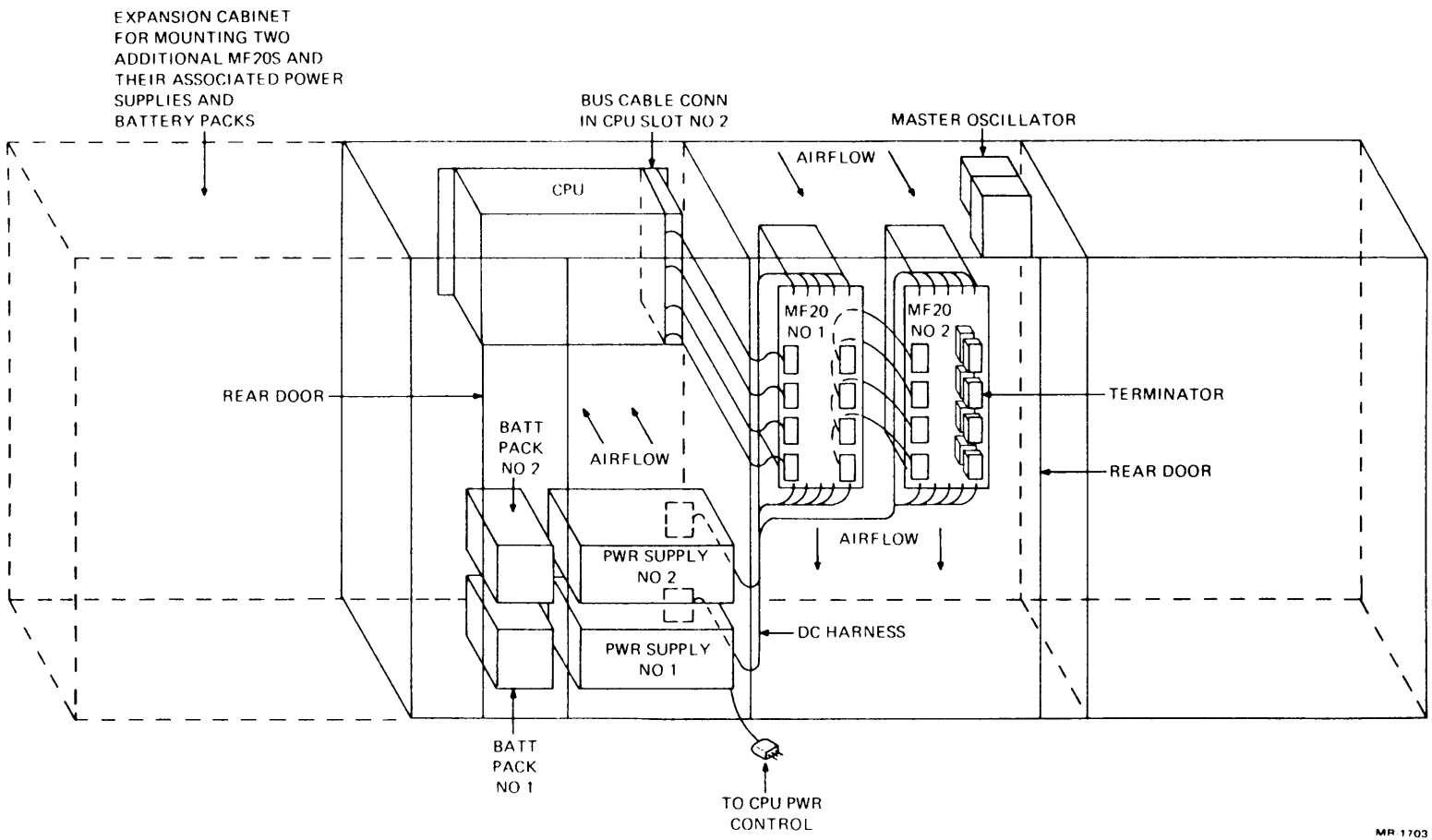


Figure 1-1 MF20 Physical Layout

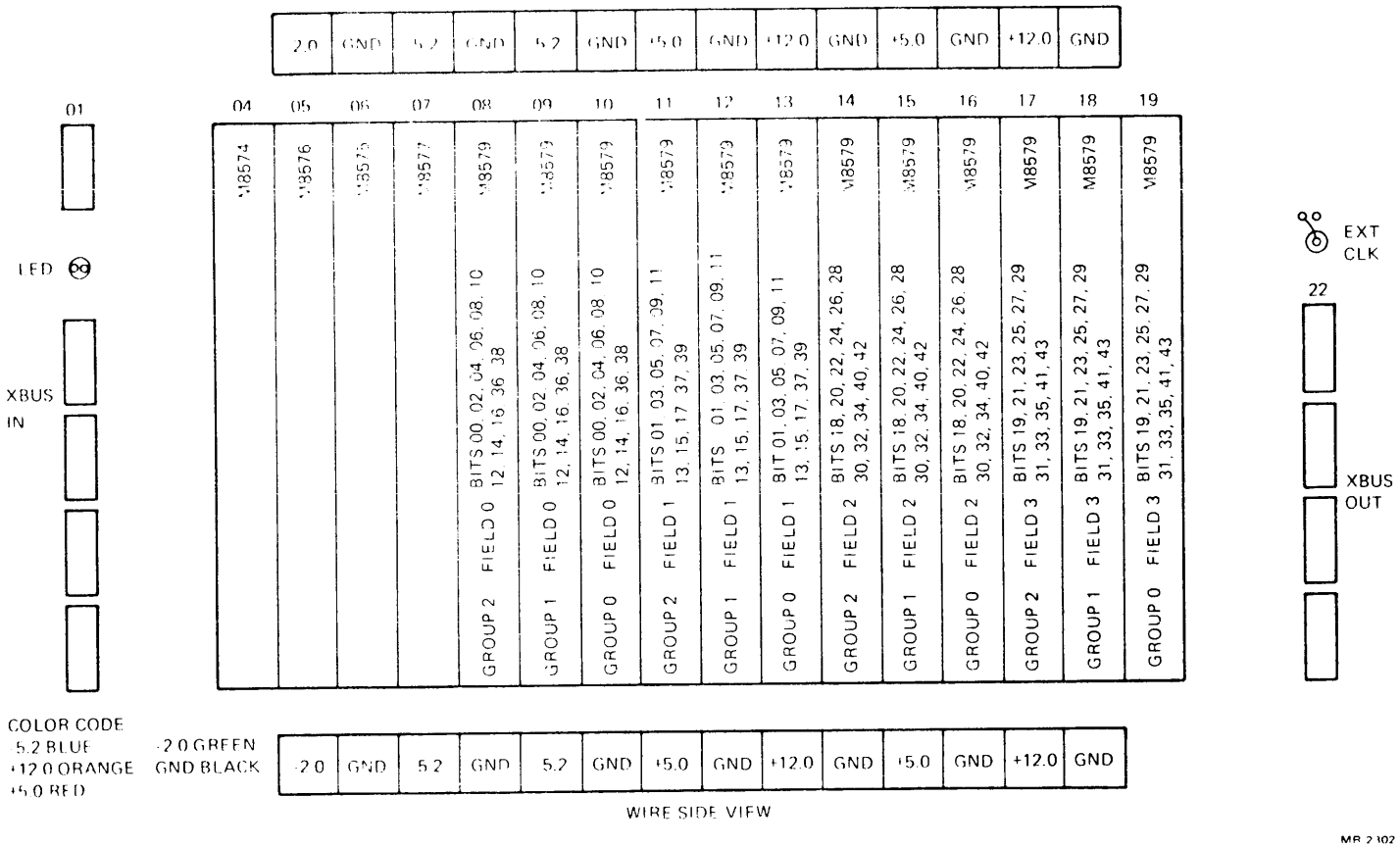


Figure 1-2 MF20 Simplified Module Utilization Diagram

2. Cooling fans

3. The following logic modules:

   a. One M8574 write path module

   b. One M8575 syndrome module

   c. One M8576 control and timing module

   d. One M8577 address and timing module

   e. Four M8579 MOS storage array modules [Each set of four M8579 modules provides a capacity of 262,144 44-bit words; up to three sets (12 boards = 786,432 words) can mount in a single MF20.]

Two MF20 logic assemblies can be mounted on the rear door of the I/O cabinet in a DECSYSTEM-20. In the KL10-R they already exist. An additional pair of MF20 logic assemblies mount in a separate cabinet along with their associated power supplies.

1.1.2.2 KW20 Master Oscillator - This unit mounts in the upper front of the system I/O cabinet (CPU cabinet in the KL10-R) and contains the clock source that provides timing for both the KL10 CPU and the MF20s. A single KW20 can provide all the clocks required to operate up to four MF20 logic assemblies.

1.1.2.3 Power Supply (H7131/H7131-A) - A standalone power unit provides all the necessary dc voltages to operate a single MF20. Each MF20 requires a separate power supply. The power supply includes a battery backup assembly that maintains sufficient power to operate the MF20 for up to 30 seconds following a power failure.

1.1.2.4 KL10 CPU Modifications - The MF20/MF20F uses an improved version of the KL10's SBus called the XBus. To support the XBus the M8519 SBus translator modules in slots 7 and 8 of the CPU backplane must be replaced as follows.

1. Core/MOS mixed configurations use two M8580 modules in place of the M8519s.

2. MOS-only configurations use two M8581 modules in place of the M8519s.

Besides the translator modules, a quad-height XBus cable connector, the M8572, plugs into slot 2 to connect the MF20 to the CPU via the XBus. A second M8572 (YA version) plugs into slot 3 if an expansion cabinet with additional MF20s is installed.

1.1.2.5 Miscellaneous Components - The MF20 list of physical components includes various cables, power harnesses, terminators, etc. These components will be described in Chapters 2 and 3.

1.1.3 Configurations (Figure 1-3)

The MF20 can be configured on single-processor KL10-based systems in integral multiples of 256K (262,144) up to a maximum of 3 megawords (3,145,728). The physical components discussed in Paragraph 1.1.2 are packaged in the following discrete options.

MF20-LA - Basic 256K memory with the master oscillator box (KW20); 60 Hz

MF20-LB - 50 Hz version of MF20-LA

MF20-LC - Additional logic assembly and power supply for expansion from 768K to 1.5 megawords on 60 Hz systems

MF20-LD - 50 Hz version of MF20-LC

MF20-E - A set of four M8579 storage modules (a group) that comprises a 256K expansion unit

MF20-LH - First box of 256K memory packaged in an external cabinet for expansion greater than 1.5 megawords; 60 Hz

MF20-LJ - 50 Hz version of MF20-LH

MF20-LK - Second additional logic assembly and power supply for expansion greater than 2.2 megawords; 60 Hz

MF20-LL - 50 Hz version of MF20-LK

Table 1-1 shows the options that must be ordered to configure an MF20 subsystem of a given capacity. Figure 1-3 shows simplified block diagrams of three possible MF20 configurations.

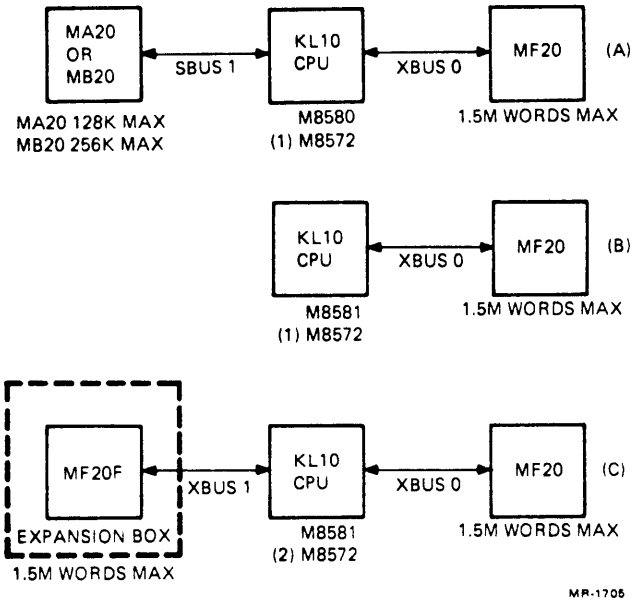


Figure 1-3 MF20 Configuration

Table 1-1 New System Configuration Guide (MOS Only)

Storage Capacity (256K Units)	Options 60 Hz	50 Hz
1	MF20-LA	MF20-LB
2	MF20-E	MF20-E
3	MF20-E	MF20-E
4	MF20-LC	MF20-LD
5	MF20-E	MF20-E
6	MF20-E	MF20-E
7	MF20-LH	MF20-LJ
8	MF20-E	MF20-E
9	MF20-E	MF20-E
10	MF20-LK	MF20-LL
11	MF20-E	MF20-E
12	MF20-E	MF20-E

NOTE

To compute total capacity in 44-bit words (36 data, 7 check, 1 spare), multiply number of units by 262,144.

Example: 3 units
3 X 262,144 = 786,432.

1.1.4 Technical Overview

The MF20 is more than just a replacement for core memory subsystems on the DECsystem-10/DECSYSTEM-20. It is a sophisticated MOS memory storage subsystem with extensive programmable controls that must be set up by the software before it can be used to store and retrieve information. This paragraph provides a brief description of the hardware-software interaction. Refer to Chapter 5 for a complete technical description of the MF20 logic.

1.1.4.1 Timing and Control - Most memory systems are hard-wired to generate fixed sequences of control and timing signals. These perform the basic read, write, and read-pause-write operations that transfer information between the CPU and the memory's storage array. The pulse widths and repetition rates of these signals are chosen so that the memory circuits operate error-free under worst-case conditions. Once chosen, the circuit parameters of most memory systems are fixed and can only be modified by extensive hardware wiring and circuit changes.

This is not the case in the MF20. A control RAM called the timing RAM is used to establish the signal sequences for accessing the MOS storage array. The actual order and pulse widths of the signals generated become a function of the bit patterns loaded into this RAM. When the MF20 is initially powered on, software resident in the PDP-11 must load the timing RAM to establish the proper control and timing sequences. To modify the timing to compensate for variations between different MOS chip manufacturers, the software simply changes the contents of the timing RAM.

1.1.4.2 Configuration and Patching - Earlier DEC memories established the actual relationship between program address and physical memory modules by setting certain front panel switches. The initial process of setting up these switches was called configuration and required manual operator intervention. If certain memory modules failed, they could be patched out by reconfiguring the switches to place the bad boxes off-line until they could be repaired. If a single bit failed in a particular module, that whole module had to be patched out and the system operated with a considerable loss in its available primary storage resource.

In the MF20 the processes of configuration and patching are automated and programmable. A control RAM called the address response RAM is used to establish the relationship between program address and physical memory modules. Effectively, the software sets the switches by loading the RAM with specific bit patterns. During real-time operations the software can reconfigure the memory to patch out failed areas by simply changing the contents of the RAM.

The chance of a single-bit error (SBE) causing considerable loss in the available MOS resource is minimized by use of a hardware

spare bit mechanism. This consists of spare MOS RAM chips coupled with a control RAM that decides which bit positions are to be substituted for which addresses. This spare bit substitution RAM is loaded by the program to establish the relationship between address and bad-bit position. System software uses the error correction code (ECC) single-bit error detection mechanism coupled with special diagnostic function instructions to determine and log the failing bit position. Using this information the program can set up the spare-bit substitution RAM to substitute the spare bit for the failing bit. The philosophy is based on the fact that single-bit errors are correctable and transparent to the program. If an SBE is substituted for before a second bit fails, the chance that a double-bit error (DBE) will cause a fatal error is minimized. Like the other control RAMs, the spare bit substitution RAM must be initialized by the program during system startup.

1.1.4.3 Error Detection and Correction - The MF20 contains additional hardware for detecting and correcting errors. It uses a standard ECC code to provide the mechanism for detection and correction of SBEs, and detection only of DBEs. This feature coupled with the spare-bit logic provides the MF20 with extremely high reliability.

1.1.4.4 Fault Isolation - Eleven diagnostic functions have been included in the design of the MF20. These diagnostic functions are special instructions that permit the program to load the control RAMs during initialization, configuration, and patching, and to read them during error analysis and recovery. A comprehensive set of diagnostic instructions permits the program to read the states of internal logic signals when analyzing errors. With the ability to gather detailed information about errors, the diagnostic software provides error reports and statistics that enable the service engineer to rapidly locate a failing module and make the repair.

1.1.4.5 Summary - It is important to recognize the inter-dependence of hardware and software when working with the MF20. Since the control RAMs are volatile, they must be set up properly by running certain programs. In effect, the MF20 is not an MF20 until the correct information is loaded into the RAMs. Chapter 4 contains a more detailed discussion of this hardware-software interdependency.

1.2 MF20 SYSTEM SPECIFICATIONS

Memory Type	Metal Oxide Semiconductor (MOS)
Capacity	Minimum 262,144 words Maximum 3,145,728 words
Word Length	44 bits (36 data bits; 6 ECC bits; 1 ECC parity bit; 1 spare bit)
Interleave	Always 4-way

Refresh Interval	All cells within the MOS array must be refreshed once every 2 ms
Read/Write Cycle and Access Times	See Table 1-2
Power Supply	
Type	Switching regulator
Outputs, dc	+5 V ± 1.5% @ 35 A max. +12 V ± 1.5% @ 11 A max. -5.2 V ± 1.5% @ 60 A max. -2 V ± 1.5% @ 25 A max.
Input, ac	
Voltage	90-132 Vac 180-264 Vac
Current	12 A @ 120 Vac
Frequency	49-51 Hz 59-61 Hz
Power Factor	70%
Size	
Supply	66.04 cm X 22.86 cm X 35.56 cm (26 in X 9 in X 14 in)
Battery	20.32 cm X 30.48 cm X 25.4 cm (8-5/8 in X 9-3/4 in X 10 in)
Weight	
Supply	30.84 kg (68 lb)
Battery	12.6 kg (28 lb)
Battery Pack	
Voltage	96 Vdc nom. in two 48 V segments
Protection	One 35 A circuit breaker for each 48 V segment
Charge Current	4 mA trickle charge
Charge Voltage	145 ± 8 Vdc open-circulated
Crossover Voltage	90 ± 2 Vac rms

Min. Battery Voltage	85 Vdc during discharge
Recharge Time	18 hr for each 30 s discharge time

Environment

Temperature	15° to 32° C (59° to 90° F) (operating) -40° to 66° C (-40° to 151° F) (storage)
Humidity	20% to 80% (operating) 0% to 90% (storage)
Rate of Change	7° C/hr (12° F/hr) 2%/hr
Air Volume Inlet	600 ft^3/min

KW20 Oscillator

Number of outputs	5
Frequencies	25 MHz (min.) 30 MHz (nom.) 31 MHz (max.) External Source
Control	Switchable under program control

1.3 REFERENCES

1.3.1 Drawings

MF20 Field Maintenance Print Set (MP00662)
KL10-E Field Maintenance Print Set (MP00300)
KL10-PV Field Maintenance Print Set (MP00301)
MF20F Field Maintenance Print Set (MP01711)
KL10-R Field Maintenance Print Set (MP01708)

1.3.2 Hardware Documentation

The following manuals are available in hard copy from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532
Attention: Printing and Circulating Services (NR03/W3)
Customer Services Section

MBox Unit Description Manual (EK-0MBOX-UD)
KL10-Based DECSYSTEM-20 Installation Manual (EK-0KL20-IN)
DECSYSTEM-20 Site Preparation Guide (EK-DEC20-SP)
KL10-Based Site Preparation, Power System,
Installation Manual (EK-0KL10-SP)

1.3.3 Software Documentation

TOPS-20 Operator's Guide (AA-41760-TM)
TOPS-20 Operator's Guide Addendum (AA-41760-T2)
SPEAR Manual (AA-J833A-TK)

Table 1-2 Read and Write Cycle and Access Times

Action	At Memory	At CPU
Read cycle time		
Successive reads of 1 word	733 ns	733 ns
Read 1 word followed by a write	733 ns	1000 ns
For each additional word of a 2-, 3-, or 4-word read, add:	200 ns	200 ns
Write cycle time		
1-word write*	867-1133 ns	867-1133 ns
4-word write	1267 ns	1267 ns
Read access time		
Read first word	533 ns	800 ns
For each additional word of 2-, 3-, or 4-word read	200 ns	200 ns
Write access time		
1-word write	133 ns	267 ns
For each additional word of 2-, 3-, or 4-word write	133 ns	133 ns

*The time varies depending on the position of the word in a quadword group.

NOTE
These times are based on a nominal clock frequency of 30 MHz.

CHAPTER 2
SITE PREPARATION AND PLANNING

2.1 INTRODUCTION

This chapter provides information necessary to prepare the site prior to installing the MF20, and to schedule and plan the subsequent installation. Since installation of the MF20 option involves a major change to the DECSYSTEM-20's mainframe, it is imperative that the customer and the field service engineer work together to develop an effective installation plan and schedule prior to the arrival of the MF20 at the site. Proper planning will minimize the amount of time the system will be unavailable for customer use.

2.2 SITE PREPARATION

Since the MF20 is an addition to an existing DECSYSTEM-20, the site preparation activity is minimal. Most of the site preparation activity has already been done at the time of the DECSYSTEM-20 installation itself. Any detailed site preparation information required and not found in this document is available in the DECSYSTEM-20 Site Preparation Guide (EK-DEC20-SP). Tables 2-1 and 2-2 provide the detailed mechanical, electrical and environmental data required to plan the installation.

2.2.1 Internal MF20 Options

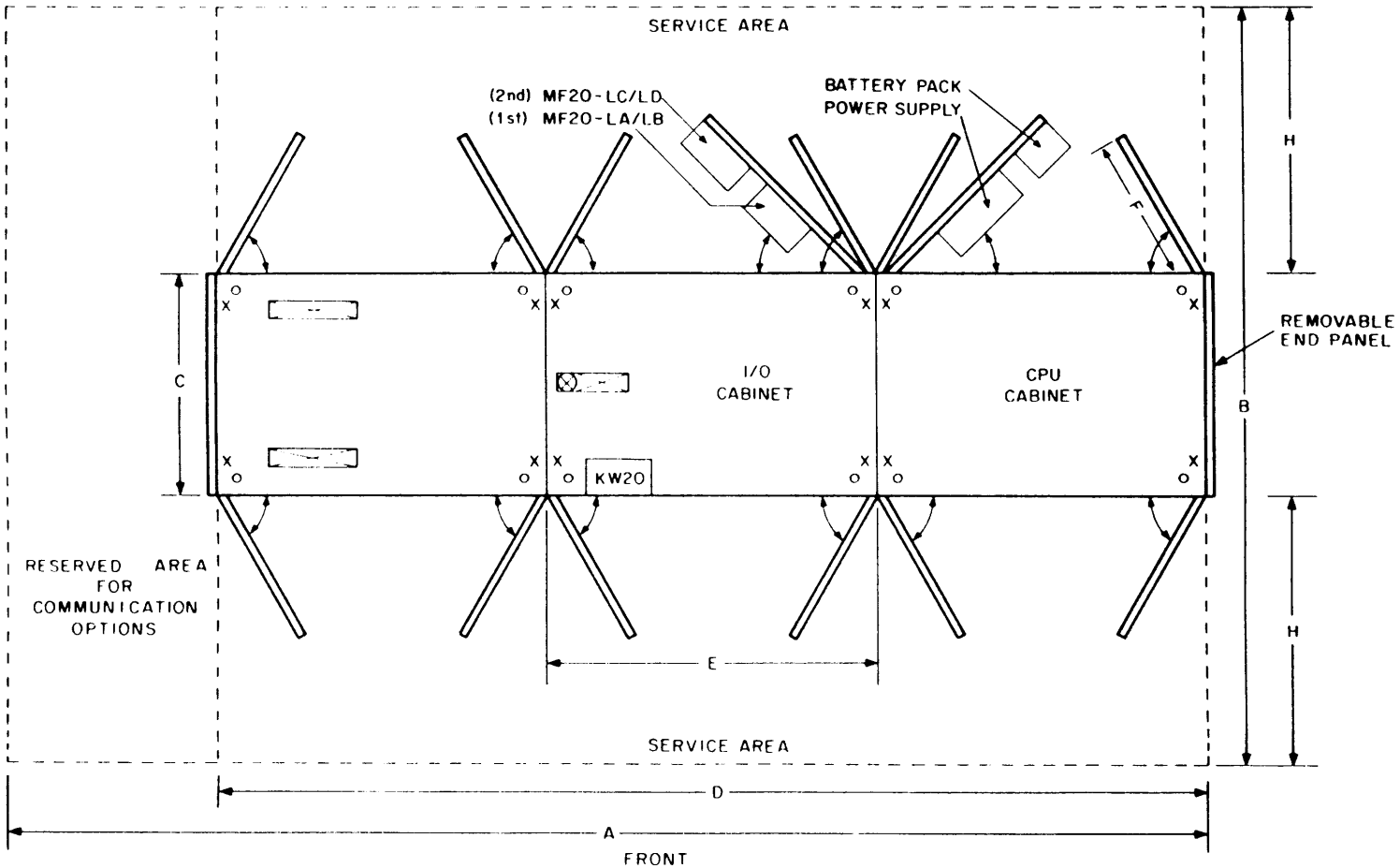
The MF20-LA/LB, LC/LD options are installed on the rear equipment mounting doors that already exist in the DECSYSTEM-20 cabinets. Refer to Figure 2-1 for the physical details. Since the DECSYSTEM-20 cabinets will already be fixed in place, the floor space requirements have already been defined. The primary consideration prior to receiving the MF20 at the site will be the transportation from the point of delivery at the site to the actual computer room where the DECSYSTEM-20 is located. The MF20 (LA/LB or LC/LD options) will be shipped on a specially fabricated shipping skid with brackets used to mount the following hardware units.

1. MF20 logic assembly
2. H7131 power supply
3. Battery pack
4. KW20 master oscillator (MF20-LA/LB only)

The same skid will be reused to remount the MA20 or MB20 if it is to be removed and returned for credit. Refer to Figure 2-2 for a description of the shipping skid.

2.2.2 External MF20 Options

The MF20-LH/LJ, LK/LL options mount in a free-standing cabinet that bolts onto the end of the DECSYSTEM-20 to the right of the CPU cabinet (viewed from the front). Refer to Figure 2-3 for the physical details. It is important here to consider the present location of the DECSYSTEM-20 in the computer room. There may not be sufficient space to the right of the CPU cabinet to attach the



X = CASTERS
O = LEVELERS

DIMENSIONS	A	B	C	D	E	F	G	H
METERS	4.11	2.59	0.76	3.43	1.14	0.56	0.76	0.91
INCHES	162	102	30	135	45	22	30	36

*TWO BATTERY PACKS AND TWO POWER SUPPLIES MOUNTED ONE ABOVE THE OTHER. BOTTOM UNITS ARE FOR FIRST MF20.

MR-2266

Figure 2-1 KL10-C/E CPU Unit Floor space Requirements

Table 2-1 MF20-LA/LB, LC/LD Data Sheet
(Internal Memory)

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
RM		N/A	N/A	N/A	N/A	9905990-00

POWER (AC)

AC Voltage			Frequency Tolerance	Phase(s)	Steady State Current (CRMS)	Surge Current	Surge Duration
Low	Nom	High					
100 191	115 220	122 233	60 Hz 1 50 Hz 1	1 1	7.6 A 3.8 A	38 A 19 A	less than 200 ms

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
30 s Note 1	2390 Btu/hr 602 kg-cal/hr	700	874	N/A Note 2	N/A Note 2	.5 mA

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume Inlet
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	
15 to 32 C 59 to 90 F	-40 to 66 C -40 to 151 F	20–80%	0–90%	7° C/hr 12° F/hr	2%/hr	600 ft^3/min

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
N A	N A	N A	N A	N A	N A

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
XBus Note 3	N A	N A	N/A	N A

Table 2-2 MF20-LH/LJ, LK/LL Data Sheet
(External Cabinet)

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
FS		152 cm 60 in	116 cm 45 in	76 cm 30 in	3-H9500	N/A

POWER (AC)

AC Voltage			Frequency	Phase(s)	Steady State	Surge	Surge
Low	Nom	High	Tolerance		Current (CRMS)	Current	Duration
100 191	115 220	122 233	60 Hz 1 50 Hz 1	1 1	7.6 A 3.8 A	38 A 19 A	less than 200 ms

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
30 s Note 1	2390 Btu/hr 602 kg-cal/hr	700	874	N/A Note 2	N/A Note 2	.5 ma

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	Inlet
15° to 32° C 59° to 90° F	-40° to 66° C -40° to 151° F	20–80%	0–90%	7° C/hr 12° F/hr	2%/hr	600 ft^3/min

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
N/A	N A	N/A	N/A	N/A	N/A

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
XBus Note 3	N/A	N/A	N/A	N/A

NOTES: (applicable to Tables 2-1 and 2-2)

1. Battery back-up supply

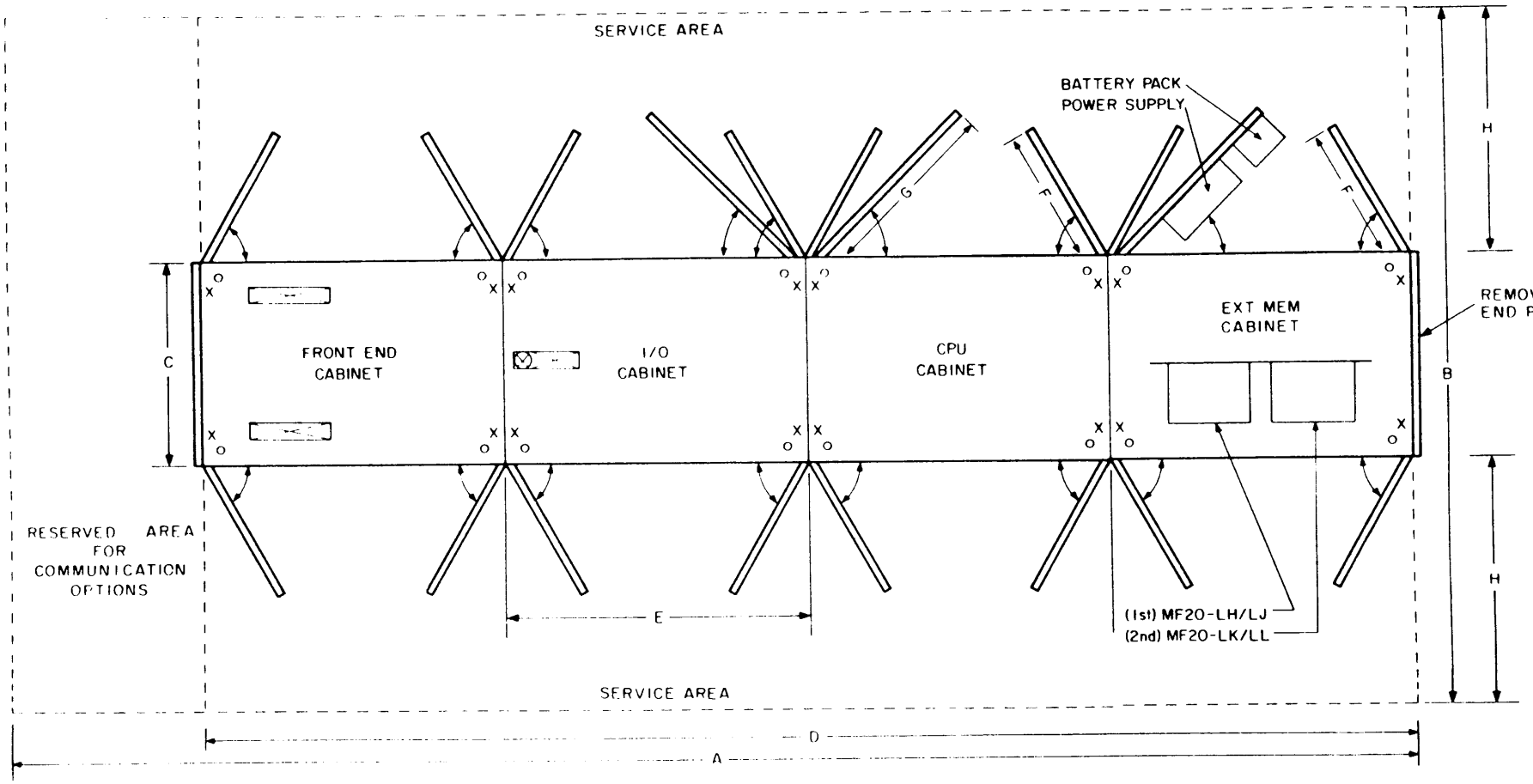
2. Connects to the 863 power control unit via cabling and connectors supplied with the option

3. External memory: One 13 ft XBus cable connects SBus 1 in the CPU cabinet to the MF20 no. 2 in the external cabinet.

   Four XBus jumper cables (1 ft) connect MF20 no. 1 to MF20 no. 2 in the external cabinet.

   Internal memory: One 3 ft 8 in XBus cable connects SBus 0 in the CPU cabinet to the MF20 no. 1 in the I/O cabinet.

   Four XBus jumper cables (1 ft) connect MF20 no. 1 to MF20 no. 2 in the I/O cabinet.



X = CASTERS
O = LEVELERS

DIMENSIONS	A	B	C	D	E	F	G	H
METERS	4.11	2.59	0.76	4.57	1.14	0.56	0.76	0.91
INCHES	162	102	30	180	45	22	30	36

*TWO BATTERY PACKS AND TWO POWER SUPPLIES MOUNTED ONE ABOVE THE OTHER. BOTTOM UNITS ARE FOR FIRST MF20.

MR-226

Figure 2-2 KL10-C/E CPU-MF20 LH/LJ Unit Floor Space Requirements

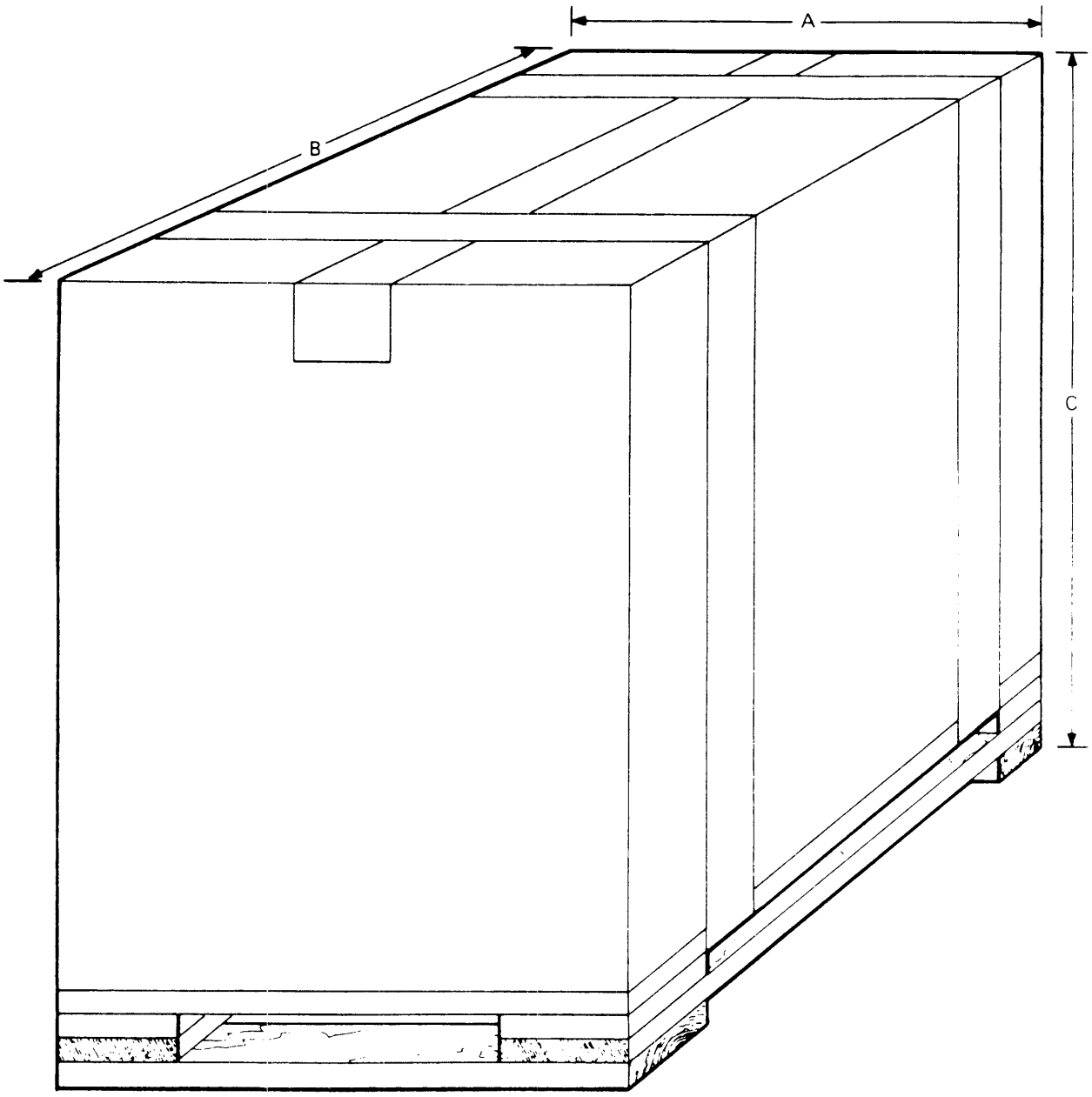


Figure 2-3 MF20 LA/LB, LC/LD, LK/LL Packaging

external memory. In this case the entire DECSYSTEM-20 will have to be relocated to a position in the computer room that permits installing the MF20. This could involve considerable changes and must be accomplished prior to receiving the MF20 cabinet on site. As with the internal MF20, the prime consideration prior to receiving the MF20 at the site will be the transportation from the point of delivery at the site to the actual computer room where the DECSYSTEM-20 is located. The MF20 (LH/LJ option) will be shipped on a specially fabricated skid. Refer to Figure 2-4 for a description of the package shipped. Refer to the DECSYSTEM-20 Site Preparation Guide, pages 1-38 for detailed equipment handling information.

2.3 INSTALLATION PLANNING

At least 30 days prior to the scheduled arrival of the MF20 at the computer site, the customer and the field service engineer should meet to plan and schedule the installation. To minimize the amount of time the system must be down during any one day, a 3-phase approach is suggested. The 3-phase approach will also minimize the impact of any unexpected problems that may arise. Without a careful plan these problems could seriously affect the customer's normal work schedules.

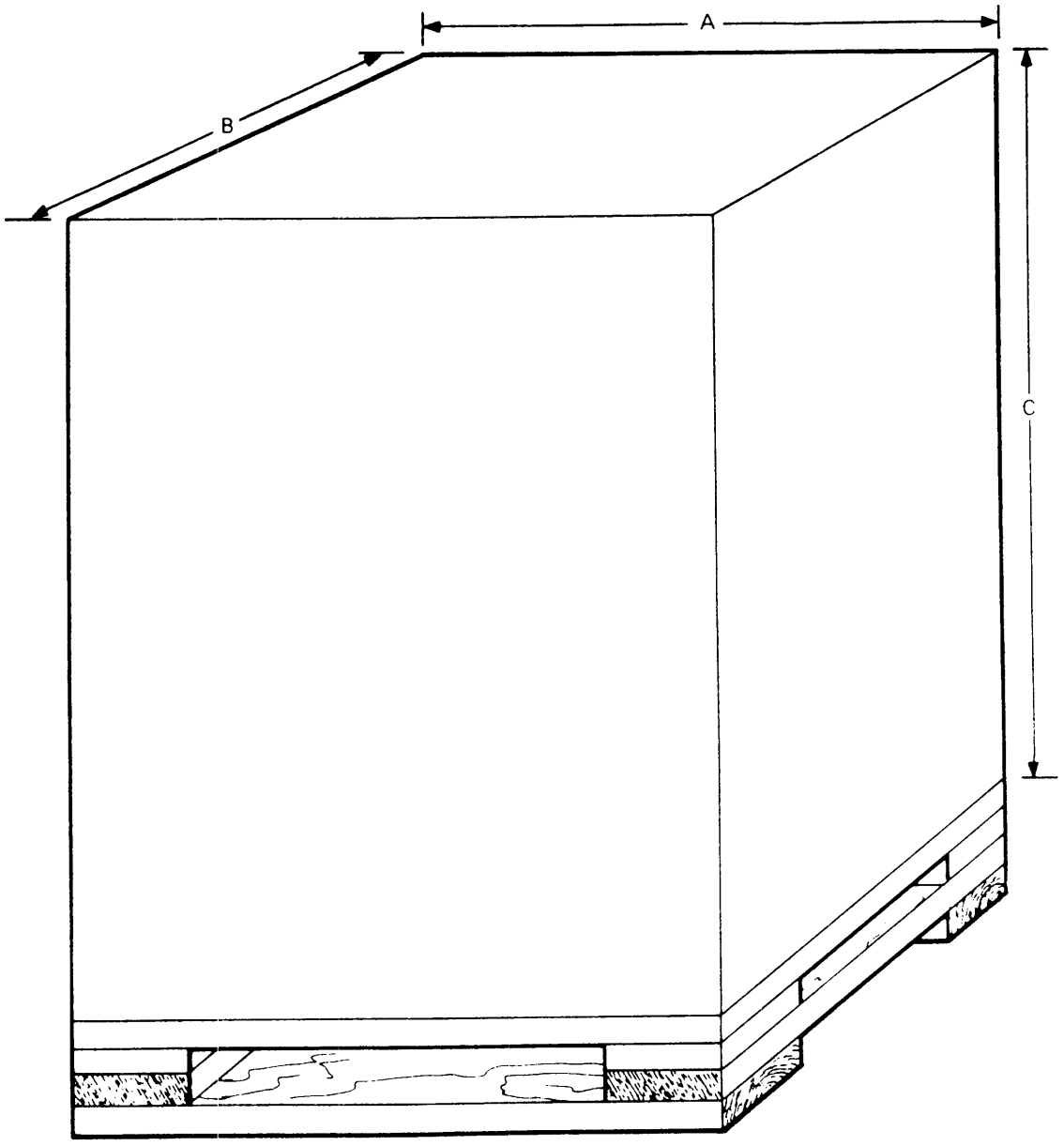
2.3.1 Phase 1 - Prearrival Activity

2.3.1.2 Mounting Hole Check

1. Power Supply - Check the CPU cabinet rear equipment mounting door for holes and rivnuts in the short member of the door frame as shown in Figure 2-5.

> NOTE
>
> If the system presently has H7420 power supplies mounted in this position for the MA20/MB20 add-on, these holes are already present. If not present, holes must be drilled and rivnuts mounted before the H7131 power supplies can be mounted. Contact the LCG Product Support group in Marlboro, Mass. at (617)481-9511 to obtain the special template required. Only a very limited number of DECSYSTEM-20 cabinets will need the holes installed.

2. Battery Box - Check CPU cabinet rear equipment mounting door for holes and rivnuts in the members to accept the battery box as shown in Figure 2-5.



A = 45 IN.
B = 27 IN.
C = 57 IN.

MP 2269

Figure 2-4 MF20 LH/LJ Packaging

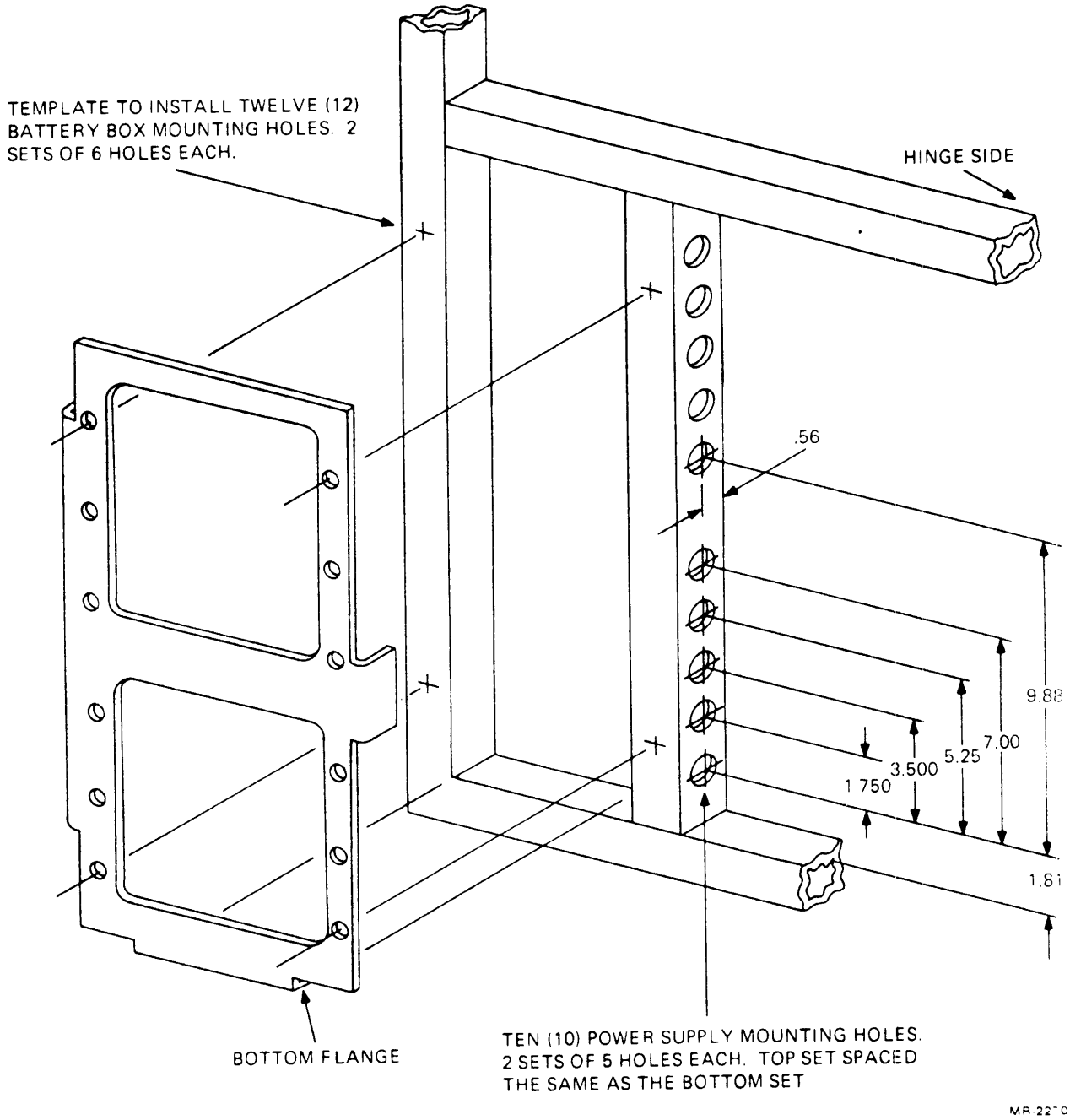


Figure 2-5 CPU Cabinet Rear Equipment Mounting Door

NOTE

The template used to install these holes will be shipped as part of the MF20 option. The rivnut tool shown in Figure 2-6 will be supplied as part of an MF20 controlled-distribution kit.

3. Check to verify that the CPU backplane is at REV 4 level or higher.

   If the wiring changes have not been made, the ECOs required to bring the KL10 up to REV 4 must be installed prior to scheduling the MF20 installation.

NOTE

The system should be operated on-line for at least two weeks after installing REV 4 before scheduling the MF20 installation.

2.3.2 Phase 2 - Preinstallation Checkout

After the MF20 arrives at the site the second phase must be planned and scheduled. This involves the following two major operations.

1. Unpacking and inspection (Refer to Paragraph 3.3.)

2. Preinstallation checkout (skidded checkout) (Refer to the Engineering Installation Procedure ASP-MF20-0-2.)

The purpose of the skidded checkout is to verify that the MF20 is working properly before mounting it in the customer's system and possibly taking away his core memory system. If the MF20 was damaged during shipment and requires major repairs, we want to be in a position to restore the customer's system to its normal condition as rapidly as possible.

2.3.3 Phase 3 - Final Installation and Checkout

After the skidded checkout has verified that the MF20 is ready to be installed, the units are mounted permanently into the DECSYSTEM-20. The cables are properly dressed and tied, and final checkout and acceptance tests run.

2.4 INSTALLATION SCHEDULING

The actual scheduling of the installation should be worked out in advance. See Table 2-3. The actual times listed in Table 2-3 will vary dependent on the level of experience of the engineer performing the installation and the particular configuration installed.

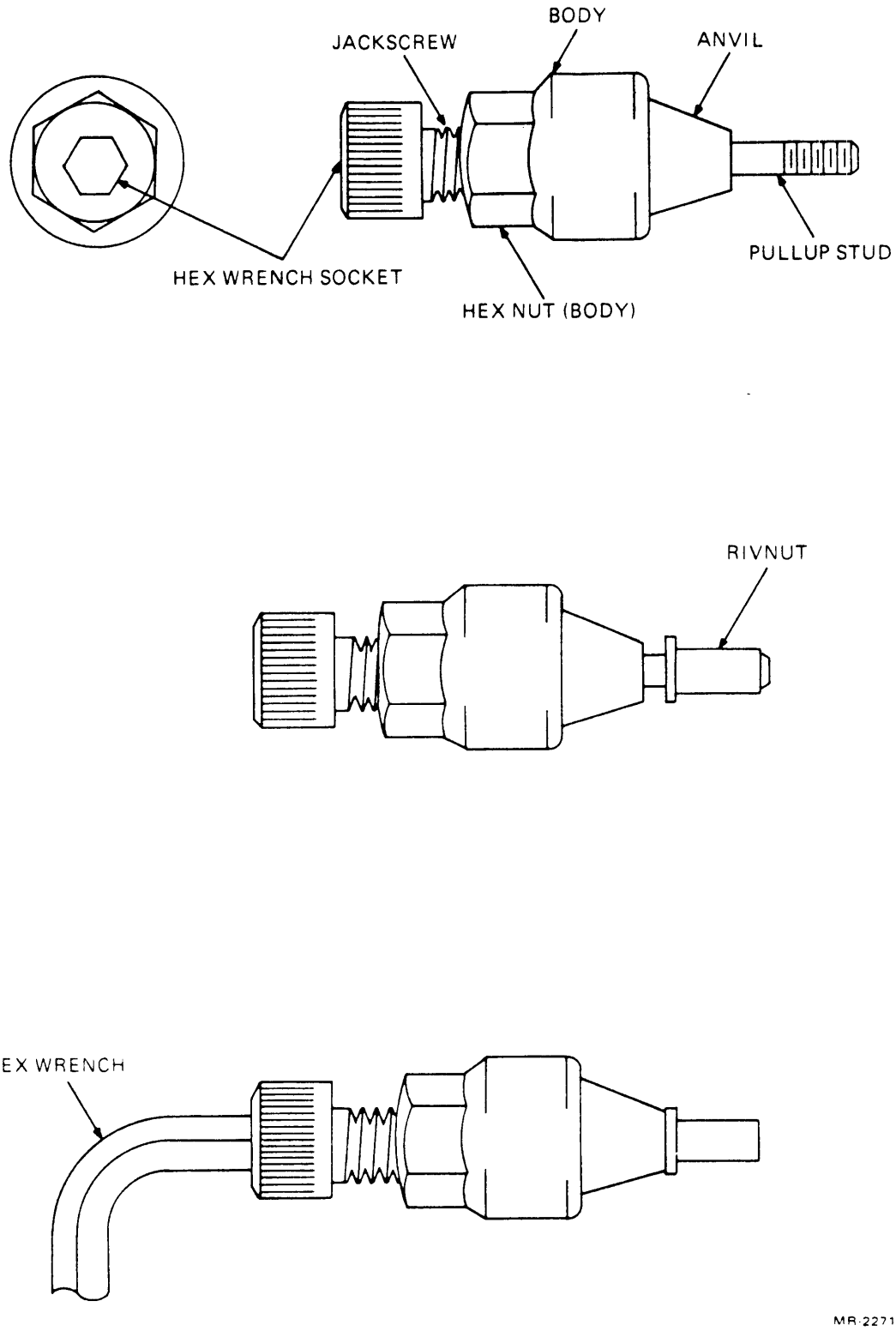


Figure 2-6 Rivnut Tool

Table 2-3 Installation Scheduling

Phase	Activity	Time Required
1	Check KL10 REV Check for mounting holes	15 min to 6 hr depending on whether ECO needs to be installed.
2	Unpacking and inspection Drill holes - insert rivnuts Skidded checkout	8 hr
3	Permanent installation Checkout and acceptance	8 to 12 hr

NOTE

Because of the bulk and weight of the units to be installed, two people must be sheduled to perform the job.

CHAPTER 3
INSTALLATION

3.1 INTRODUCTION

This chapter provides the information necessary to unpack, install, and check out the MF20 MOS Storage Subsystem in any DECsystem-10/DECSYSTEM-20 computer system. The level of description assumes that the service engineer in charge of the installation (at least two people are required in most cases) is an experienced DECsystem-10/DECSYSTEM-20 service engineer who has completed the formal MF20 maintenance course. Prior to performing the installation, the engineer should read all of Chapters 2 and 3 in this manual and become familiar with the content of the other documents referenced by the procedures. Since the installation procedures are complicated and will vary depending on the specific circumstances, the information is organized to permit application to the following general situations.

1. Adding a single MF20 to the rear door of the system I/O cabinet. This may or may not involve removing an existing MA20 or MB20 core memory which may be mounted there already.

2. Adding a second MF20 to the system I/O cabinet to expand the capacity of an existing MOS memory subsystem.

3. Adding an external memory cabinet that houses a single MF20 susbsystem.

4. Adding a second MF20 to the external cabinet to expand the capacity of the MOS memory subsystem.

5. Simply plugging in a set of four storage modules to increase the capacity (256K per set) of an existing MF20.

3.2 REFERENCE DOCUMENTATION

The following reference documentation should be readily available at the time and location of the installation.

1. MF20 MOS Storage Subsystem Technical Manual (EK-0MF20-TM)

2. MF20 Field Maintenance Print Set (MP00622)

3. KL10 Field Maintenance Print Set (MP00300)

4. KL10 Microfiche Library (manuals and diagnostic listings)

5. MF20 Installation Procedure (ASP-MF20-0-2)

3.3 UNPACKING AND INSPECTION (Figure 3-1)

3.3.1 Unpacking

Digital has implemented strict shipping regulations to ensure that all hardware and software is received as stated in the customer's order. All boxes and skids should be left intact to avoid damaging or losing equipment, accessories, parts, or manuals. Before unpacking the MF20 from its shipping container the following checks should be made to verify the integrity of the shipment.

1. Ensure that all containers are sealed as evidenced by labels placed across the tape/band.

2. Verify that the box and skid housing the MF20 do not have any visible damage.

3. Verify that the correct number of boxes and skids was received. (These numbers will appear on one of the gummed labels affixed to the shipment.)

If you see any damage or if there are any missing items, report this to the shipper and the local DIGITAL field service branch supervisor responsible for the installation.

When ready to begin the actual installation, move the shipping skid to a convenient work area close to the DECsystem-10/ DECSYSTEM-20. (Refer to Figure 3-1.) Cut the plastic strappings that secure the cardboard carton to the shipping skid. Remove the cardboard carton and polyethylene bag used to protect the MF20 during shipment.

NOTE

When unpacking the MF20 shipment, save all the shipping containers and mounting hardware. These may be needed to return a damaged MF20 or an existing MA20 or MB20 that is being removed to make room for the MF20.

3.3.2 Inspection

Once the covers have been removed, carefully inspect the MF20 hardware. During the inspection, take careful notes of any discrepancies found and report them to the shipper and the local DIGITAL branch supervisor. Any discrepancies should also be reported on the Marlboro Manufacturing Feedback Form so that shipping problems may be corrected at the source. Inspection should include the following steps.

1. A complete inventory of all items as shown on the shipping list. The shipping list itself is part of the system.

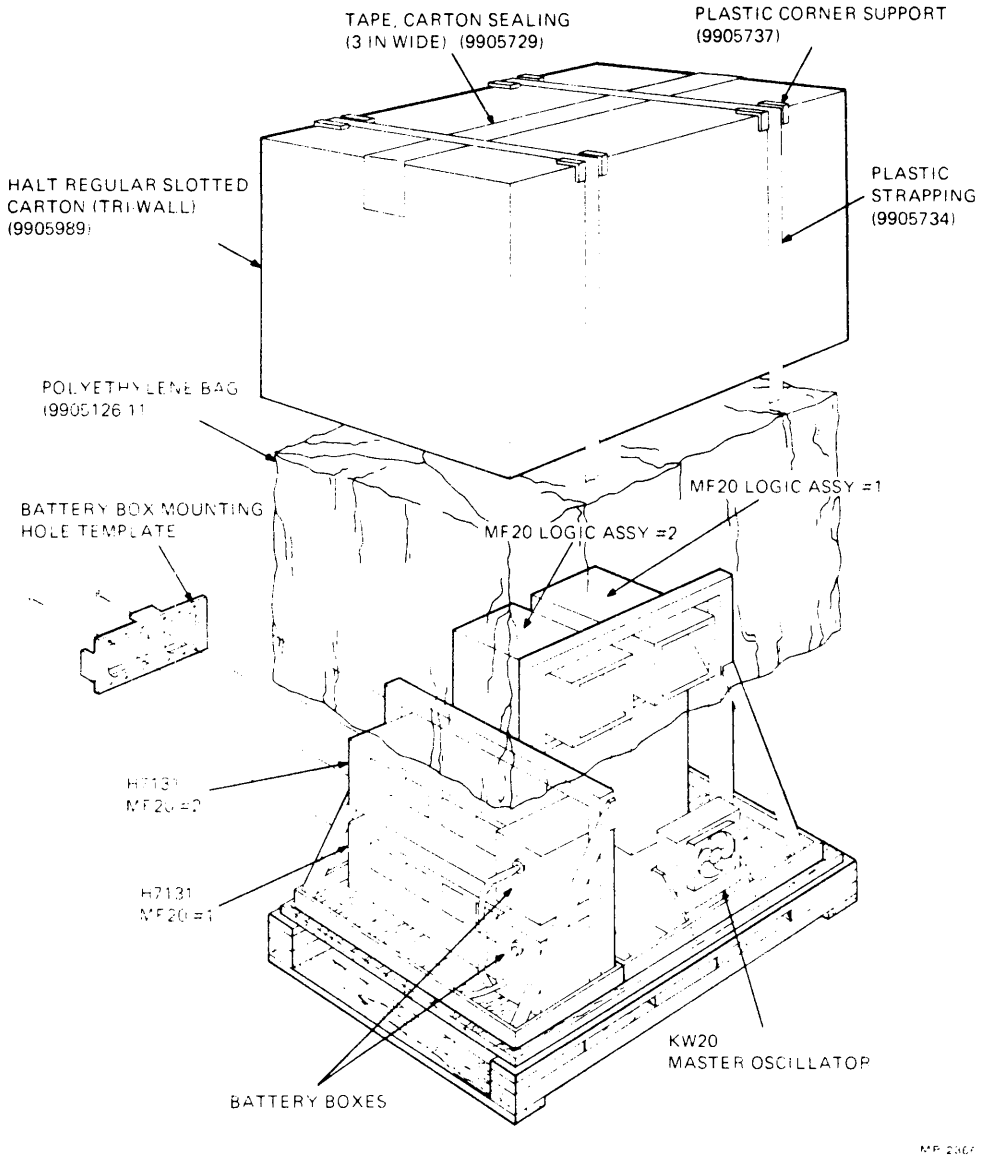


Figure 3-1 MF20 Shipping Skid

NOTE

If any items are missing (i.e., cables, modules, etc.) do not proceed with the installation. Taking the customer's system down to install the MF20 and latch and then finding you are unable to complete the job on schedule because of a missing part could cause a serious customer relations problem.

2. Inspect the logic assembly, oscillator, power supply and battery box for any physical damage (dents, bent pins, broken wires, missing screws, etc.).

3. Verify that all the logic modules are plugged into the correct slots as shown in the module utilization list (MUL).

Once it has been verified that all the components needed for the installation are available, the actual MF20 installation may be performed as outlined in the following paragraphs.

3.4 CABLING

This paragraph describes the MF20 cabling scheme along with the engineering documentation that supports it. Before proceeding with any installation the service engineer should read this paragraph and review the referenced documents. Paragraph 3.4.1 lists all the MF20 cables and incudes part numbers, names, and functional descriptions. Engineering drawing UA-MF20-0-0 (sheet 4 of 8) shows six views of the MF20 subsystem. Each view shows the approximate routing of one or more of the cables to be installed. Drawing IC-MF20-0-3 (sheets 1 and 2) shows a detailed schematic representation which supplies the information not shown in the assembly drawings.

3.4.1 MF20 Cables Listing

Part No.	Name	Description
7015190-0-0	Voltage margin cable	Connects the margin sense and control signals between the MF20 backplane and the H7131 power supply
7015453-0-0	Door switch harness	Connects the MF20 logic assembly door lock microswitch to the DECsystem-10/ DECSYSTEM-20 door lock sensing logic

Part No.	Name	Description
7015477-0-0	Temperature sense harness	Connects the 2 overtemperature sensors mounted on the side of the MF20 logic assembly to the DECsystem-10/DECSYSTEM-20 overtemperature sensing logic
7015671-0-0	MF20-LA/LB dc power harness	Connects the dc voltages from the H7131 power supply to the MF20 logic assembly
7015524-0-0	Clock select cable	Connects the clock select signals from the M8572 module in the CPU to the KW20 master oscillator
7015471-0-0	KW20 power harness	Connects the dc power to the KW20 from the MF20s (+12 V) and from the H7420 (+15 V) in the I/O cabinet
17-001000-0-0	Coaxial cable	Connects the clock signals from the KW20 to the CPU and each MF20 backplane - a separate cable is required for each clock signal to be routed
7015448-0-0	KW20 ac fan harness	Connects 115 Vac from the H7131 power supply to the KW20 master oscillator fan
7015222-0-0	MF20 logic assy. fan harness	Connects the 115 Vac from the H7131 power supply to the MF20 logic assembly fans
7015450-0-0	AC power cord	Connects the H7131 power supply to the 115 Vac, 60 Hz line via the 863 power control
7015449-0-0	AC power cord	Connects the H7131 power supply to the 115 Vac, 50 Hz line via the 863 power control
70-08288-8F	Remote turn-on cable	Eight foot cable that connects the H7131 power supply to the DECsystem-10/DECSYSTEM-20 main power control panel (70-11639 assembly J4) to permit remote turn-on

Part No.	Name	Description
70-08288-3F	Remote turn-on cable	Connects two H7131 power supplies together to permit daisy-chaining the remote turn-on cable
7015189-0-0	MF20-LC/LD dc power harness	Connects the dc voltages from the H7131 power supply to the second MF20 logic assembly
5412855	XBus terminator	Resistor terminator module that plugs into the backplane of the last MF20 on the XBus - four terminators are used
17-00101-0-0	XBus jumper cable	One-foot section of XBus cable used to connect MF20 no. 1 to MF20 no. 2 - four are required
D-UA-M8572-0-0	XBus cable assy. and module	Connects the CPU to the first MF20 via the XBus
B-UA-915-3-0	Patch cords	Backplane jumper wires used to establish the controller address

3.4.2 Cabling Summaries for Specific Options

3.4.2.1 MF20-LA/LB - Each of the steps listed below is referenced to one of six views shown in UA-MF20-0-0 (sheet 4 of 8) by the notation top-left, top-right, middle-left, etc. The sequence listed does not necessarily agree with the sequence in which the cables have to be installed.

1. DC power harness from the H7131 to the MF20 logic assembly (top-right view)
2. Margin-sense cable from the H7131 to the MF20 logic assembly (top-right view)
3. KW20 master oscillator power harness from the KW20 to the H7420 (+15 V) and the MF20 no. 1 backplane (+12 V) (top-right view)
4. AC power harness for the fans from the H7131 to the MF20 logic assembly (middle-left view)
5. Clock coaxial cable from the KW20 to the CPU backplane (middle-left view)
6. Clock coaxial cable from the KW20 to the MF20 no. 1 backplane (top-left view)

7. Clock frequency select cable from the M8572 module in the CPU to the KW20 (top-left view)

8. XBus cable assembly from the M8572 module in the CPU to the MF20 no. 1 backplane (top-left view)

9. Overtemperature switch harness from MF20 no. 1 to the system sensing control logic (bottom-right view)

10. Remote turn-on cable from the H7131 power supply to the 70-11639 main power switch assembly (J4) on the system front-end cabinet

NOTE

Cable may be shown in the middle-left view of UA-MF20-0-0 (sheet 4). This is a mistake.

11. AC power harness for the KW20 fan from the H7131 to the KW20 (middle-left view)

12. Main ac power cord from the H7131 to the 863 power control (middle-left view)

13. Door switch interlock cable from MF20 no. 1 to the system door lock sensing logic (bottom-right view)

14. Connect the jumper wires to the backplane to establish the proper controller number (refer to Table 3-1).

Table 3-1 Controller No. Jumper Installation

No.	Pin to Ground Connections A4L1	A4M1	A4N1
10	NC	NC	NC
11	NC	NC	GND
12	NC	GND	NC
13	NC	GND	GND
14	GND	NC	NC
15	GND	NC	GND
16	GND	GND	NC
17	GND	GND	GND

Note: NC = No connection
GND = Jumper to ground

3.4.2.2 MF20-LC/LD - Each of the steps listed below describes the additional cabling connections made when installing the second MF20 to the rear door of the system I/O cabinet. The references - top-right, middle-left, etc. - refer to the six views shown in UA-MF20-0-0 (sheet 4 of 8). The sequence listed is not necessarily the sequence in which the cables must be installed.

1. DC power harness from the H7131 to MF20 no. 2 backplane (bottom-left view)

2. Margin/sense cable from the H7131 to MF20 no. 2 backplane (top-left view)

3. Clock coaxial cable from the KW20 to MF20 no. 2 backplane (top-left view)

4. AC power harness to MF20 no. 2 logic assembly fans from the H7131 (middle-right view)

5. Four 1 ft XBus jumper cables from MF20 no. 1 to MF20 no. 2 (middle-right view)

6. Main ac power cord from the H7131 to the 863 power control (middle-right view)

7. (Refer to step 3 in Paragraph 3.4.2.1.) Connect the dc power harness (+12 V) to the KW20 from the MF20 no. 2 power supply (middle-right view)

8. Door switch interlock cable from MF20 no. 2 to the system door interlock sensing logic (bottom-right view)

9. Overtemperature switch harness from MF20 no. 2 to the system overtemperature sensing control logic (bottom-right view)

10. Remote turn-on cable from the H7131 for MF20 no. 1 to the H7131 for MF20 no. 2

11. Connect the jumper wires to the backplane to establish the proper controller number (refer to Table 3-1).

3.4.2.3 MF20-LH/LJ - Each of the steps listed below describes the cabling connections that must be made when adding an external MF20 cabinet. The references - top-right, middle-left, etc. - refer to the six views shown in UA-MF20-0-0 (sheet 4 of 8). The sequence listed is not necessarily the sequence in which the cables must be installed.

1. Remote turn-on cable from the H7131 for MF20 no. 1 in the external cabinet to the H7131 for MF20 no. 2 in the CPU cabinet (middle-right view)

2. Main power ac cord from the H7131 for MF20 no. 1 to an existing ac power connector in the CPU cabinet (top-left view)

3. Overtemperature switch harness from MF20 no. 1 in the external cabinet to an existing overtemperature sensing connector in the CPU cabinet (top-right view)

4. Door switch interlock cable from MF20 no. 1 in the external cabinet to an existing door interlock sensing connector in the CPU cabinet (upper-right view)

5. DC harness (+12 V) from the MF20 no. 1 logic assembly to the KW20 master oscillator (bottom-left view)

6. Clock coaxial cable from the KW20 to MF20 no. 1 in the external cabinet (bottom-right view)

7. AC power harness for the cabinet flushing fans to an existing ac power harness in the CPU cabinet (bottom-left view)

8. XBus cable assembly from the M8572-YA module in the CPU to MF20 no. 1 backplane (middle-left view)

9. Connect the jumper wires to the backplane to establish the proper controller number address (refer to Table 3-1).

3.4.2.4 MF20-LK/LL - Each of the steps listed below describes the additional cabling connections made when installing the second MF20 in the external cabinet. The references - top-right, middle-left, etc. - refer to the six views shown in UA-MF20-0-0 (sheet 4 of 8). The sequence listed is not necessarily the sequence in which the cables must be installed.

1. DC power harness from the H7131 to MF20 no. 2 backplane (top-left view)

2. Margin/sense cable from the H7131 to the MF20 no. 2 backplane (top-left view)

3. Clock coaxial cable from the KW20 to the MF20 no. 2 backplane (bottom-right view)

4. AC power harness to the MF20 no. 2 logic assembly fans from the H7131 (middle-left view)

5. Four 1 ft XBus jumper cables from MF20 no. 1 to MF20 no. 2 (middle-left view)

6. Main ac power cord from the H7131 to the 863 power control (top-left view)

7. Door switch interlock cable from MF20 no. 2 to the system door interlock sensing logic (top-right view)

8. Overtemperature switch harness from MF20 no. 2 to the system overtemperature sensing control logic (top-right view)

9. Remote turn-on cable from the H7131 for MF20 no. 1 to the H7131 for MF20 no. 2

10. Connect the jumper wires to the backplane to establish the proper controller number address (refer to Table 3-1).

3.5 INSTALLATION OVERVIEW

The following paragraphs provide a general description of the installation procedures for every possible case. They are not intended to provide detailed step-by-step procedures for the engineer who is installing an MF20 for the first time. These detailed procedures are described in a separate document, the MF20 Installation Procedure (ASP-MF20-0-2).

> CAUTION
>
> Prior to any installation, run the BB.CMD string to verify proper operation of the mainframe. If any faults are detected, they should be resolved prior to beginning the MF20 installation; otherwise, it will be impossible to determine whether the fault preexisted in the CPU or was caused by installation of the MF20 subsystem.

3.5.1 Installing an MF20-LA/LB

This procedure involves installing a single MF20 in an existing DECsystem-10/DECSYSTEM-20 computer system. This procedure may entail removal of an existing MA20 or MB20. The following steps outline the general procedure. References to sections are to the MF20 Installation Procedure (ASP-MF20-0-2).

1. After unpacking and inspection (Paragraph 3.3), position the MF20-LA/LB shipping skid at the rear of the DECsystem-10/DECSYSTEM-20 as shown in Figure 3-2.

2. Perform the skidded checkout procedure described in Section 6.6.

3. Based on the results of the skidded checkout and the specific installation schedule, one of two alternatives are possible at this point. Either disconnect the MF20 and restore the system to its original state, or complete the installation by continuing with step 4.

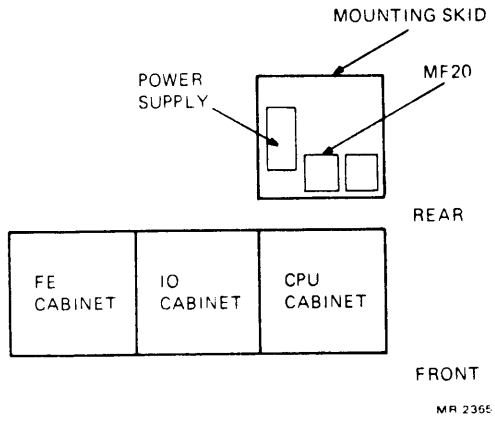


Figure 3-2 Skidded Checkout

4. If the system has an MA20 or MB20 installed on the I/O cabinet rear equipment mounting door, remove it and its associated power supplies using the procedure described in Section 7.

5. Mount the power supply, battery pack, and logic assembly for MF20 no. 1 using the procedure described in Section 8.

6. Interconnect the oscillator, power supply, logic assembly, and CPU using the cabling procedure described in Section 10.

7. Perform the preliminary electrical checkout described in Section 12.

8. Conduct the complete checkout and acceptance procedure described in Section 13.

9. At this point the installation is complete and the system is ready to be turned over to the user for on-line operation with the additional MF20 memory resource.

3.5.2 Installing an MF20-LC/LD

This procedure involves installation of a second MF20 option on the rear door of the DECsystem-10/DECSYSTEM-20 I/O cabinet. The following steps outline the general procedure. The sections referred to are in the MF20 Installation Procedure.

1. After unpacking and inspection (Paragraph 3.3), position the MF20-LC/LD shipping skid near the rear of the system I/O cabinet (Figure 3-1).

2. Mount the MF20 power supply, battery pack, and logic assembly as described in Section 9.

3. Interconnect MF20 no. 1, MF20 no. 2, the KW20, and the power supply using the cabling procedure described in Section 10 (part 2).

4. Perform the preliminary electrical checkout described in Section 12.

5. Conduct the complete checkout and acceptance procedure described in Section 13.

6. At this point the installation is complete and the system is ready to be turned over to the user for on-line operation with the additional MF20 memory resource.

3.5.3 Installing an MF20-LH/LJ

This procedure involves installing an external memory cabinet housing the MF20 MOS storage subsystem. The cabinet is bolted onto the right of the CPU cabinet and the external MF20 connected into the CPU via XBus 1 (slot 3 in the CPU backplane). The MA20 or MB20 and associated power supplies that are mounted in the CPU cabinet and connected to SBus 1 must be removed during the installation. The following steps outline the general procedure. The sections referred to are in the MF20 Installation Procedure.

1. After unpacking and inspection (refer to Paragraph 3.3) roll the external cabinet up to the system adjacent to the CPU cabinet.

2. Power down the system and perform the skidded preinstallation checkout described in Section 11, steps 1.1.1.1 through 1.1.1.8.

3. If the results of the skidded checkout and the specific installation plan indicate that the installation is to proceed, continue with step 4 below. If not, disconnect the MF20 and reconnect the internal core memory system and return the system to the user after checking it out using the BB.CMD string.

4. Remove the MA20 or MB20 that is mounted in the CPU cabinet using the procedure described in Section 11, steps 11.1.1 through 11.1.10.16.

5. Install the external MF20 cabinet permanently using the procedure described in Section 11, steps 11.2.1 through 11.2.7.

6. Perform the final checkout and acceptance procedures described in Section 13.

7. At this point the installation is complete and the system is ready to turned over to the user for on-line operation using the additional MF20 memory resource.

3.5.4 Installing an MF20-LK/LL

This procedure is similar to the MF20-LC/LD except that the additional MF20 is mounted in the external memory cabinet. The following steps outline the general procedure. Sections referred to are in the MF20 Installation Procedure.

1. After unpacking and inspecting (Paragraph 3.3), position the MF20-LK/LL mounting skid near the external memory cabinet.

2. Install the second MF20 logic assembly and power supply in the external cabinet as described in Section 11, steps 11.3.1 through 11.3.3.

3. Perform the preliminary electrical checkout as described in Section 12.

4. Conduct the checkout and acceptance procedures described in Section 13.

5. At this point the installation is complete and the system is ready to be turned over to the user for on-line operation using the additional MF20 resource.

3.5.5 Installing an MF20-E

The simplest installation involves adding one or more groups of M8579 storage modules to increase the capacity of an existing MF20 subsystem. Each set of four modules increases the capacity by 256K words. The following steps outline the general procedure.

1. Power down the system.

2. Insert the four M8579 modules into the MF20 logic assemblies as follows.

Group No.	Plug into Slots
0 (0 - 256K)	10, 13, 16, 19
1 (256 - 512K)	9, 12, 15, 18
2 (512 - 768K)	8, 11, 14, 17

3. Power on the system and conduct the checkout and acceptance procedures described in Section 13 (MF20 Installation Procedures).

4. At this point the installation is complete and the system is ready to be turned over to the user for on-line operation using the additional MF20 resource.

CHAPTER 4
OPERATION/PROGRAMMING

This chapter describes the operational and programming procedures required to place the MF20 in a state where it can be used by the system software for information storage and retrieval. The operating controls and indicators are classified into the following two categories.

1. External switches and lights
2. Internal control logic and status flip-flops

Most of the controls and indicators fall into the second category and must be activated or displayed by running specific system programs.

4.1 OPERATOR CONTROLS AND INDICATORS

4.1.1 Controls

Battery Pack Circuit Breakers - Each battery pack contains two circuit breakers to protect the battery from excessive charging current from the power supply. During normal operation both circuit breakers on each battery pack should be in the ON position. When removing or replacing the battery pack, the circuit breakers should be placed in the OFF position before disconnecting or connecting the power supply cable to the battery.

Each MF20 power supply has a separate main circuit breaker that must be in the ON position to apply primary ac power to the supply. Placing the main breaker in the OFF position removes all power from the MF20.

4.1.2 Indicators

Power Supply - Each power supply has a red POWER ON indicator that indicates when primary power is applied to the MF20 supply. This light should come on whenever the main circuit breaker is placed in the ON position.

KW20 - The master oscillator contains a LED that indicates -5.2 Vdc is present at the output of the oscillator power supply regulator. It should always be on during normal operation.

MF20 Backplane - A single LED on the MF20 backplane illuminates when the program clears the POWER BAD flip-flop. This flip-flop is set by a signal that the MF20 power supply furnishes. (Refer to Paragraph 4.3.9.)

4.2 OPERATING PROCEDURES

4.2.1 Manual Operations

1. Place the MF20 main circuit breaker in the ON position.

2. Check that the POWER ON indicator on the MF20 power supply is ON and the fans turn ON.

3. Check that the MF20 logic assembly fans turn on.

4. The LED on the MF20 backplane should be OFF.

5. The KW20 LED should be ON.

NOTE

With an MF20 subsystem installed, the KW20 must supply the clock signals to operate the CPU and memory. Since the KW20 acts as an external clock source, the following command sequence must be executed to ensure proper operation.

```
>.FW72/3    ;select 30 MHz in KW20
>.CS2       ;select external EBox clock
```

Generally this is automated by the software and should present no problems to the operator.

When the KW20 is installed, an ECO to the CPU logic must also be installed to permit the software to test for the presence of an external oscillator. This change causes an APRID instruction to return bit 21 = 1 to indicator that the KW20 is present.

4.2.2 Program Operations

After powering on the system and verifying that power is present at all MF20s and the master oscillator, the memory system must be configured. Configuration involves running an 11-based program called MEMCON under control of KLDCP or RSX20. (Refer to Figures 4-1 and 4-2.) The following procedure describes the minimum dialogue that must occur between the operator and MEMCON to properly configure the memory system.

1. Stand-alone mode using KLDCP - The following dialogue illustrates how to configure the MF20 using MEMCON. The text underlined indicates what is typed in while the remainder is typed out by the program.

```
CMD:
>. P MEMCON.A11
MEMCON.A11   VER 1.6   08-OCT-78
>. SED
DIAGNOSTIC MEMORY BOOT VER 1.6
>       DP

MEMORY RESOURCES:
CONTROLLER ADDRESS   TYPE   MODULES/GROUPS
                            7 6 5 4 3 2 1 0

        10           MF20

>       CM

LOGICAL MEMORY CONFIGURATION.
  ADDRESS  SIZE   INT   TYPE CONTROLLER
   000000  768K    4    MF20  10

>
```

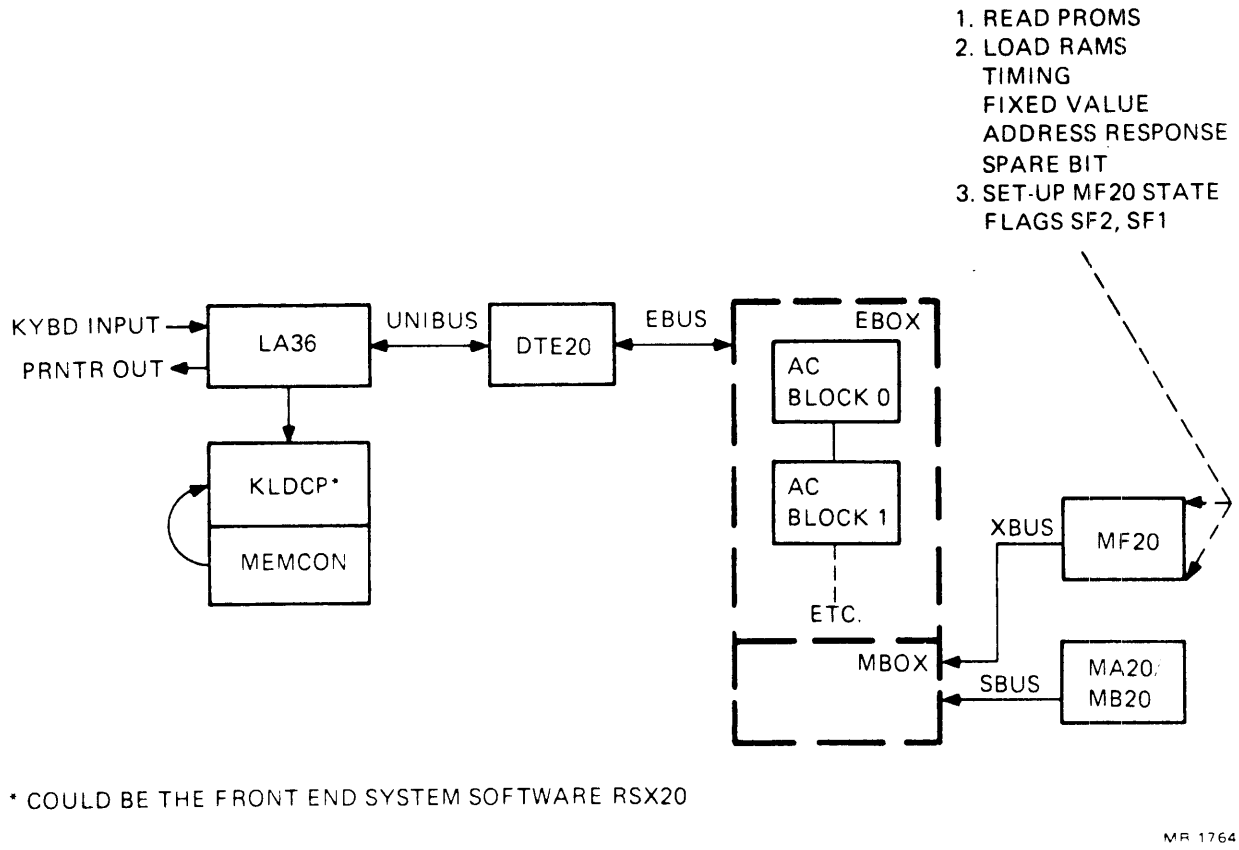


Figure 4-1 Memory Configuration Overview Phase 1

MF20 STATES

0. CONTROLLER WAS JUST POWERED UP.
1. ALL RAMS BUT ADDRESS RESPONSE RAM ARE LOADED AND CRUDE PATCHING IS DONE.
2. SAME AS 1 BUT ADDRESS RESPONSE RAMS ARE LOADED.
3. SAME AS 2 BUT TGHA HAS FINISHED ITS INITIAL TASKS.

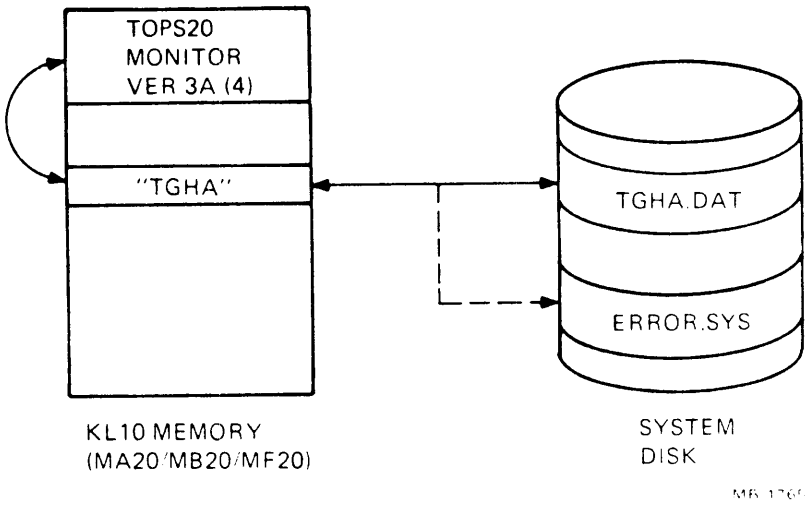


Figure 4-2 Memory Configuration Overview Phase 2

2. On-line using RSX-20F - Refer to Appendix B in the DECSYSTEM-20 Operator's Guide (DEC-20-0TPGA-A-D). Be sure to use the revision that describes the MOS dialogue, i.e., TOPS-20 REV 3A(4) or higher.

Once MEMCON has configured the MF20s, they are in state 2 (refer to Table 4-2, bits 26-27), ready to be used by system software. When system software [TOPS-20 Monitor - Version 3.., (4)] is loaded and started, it will load and start TGHA, the MF20 MOS memory RAM analyzer program. Once started, TGHA will complete the configuration process by updating the history file TGHA.DAT. At this point the MF20 is available for use by the system as a primary memory resource.

During normal system operation under control of TOPS-20, any MF20 error will cause the monitor to start up TGHA to analyze the error. TGHA will gather all the MF20 error information, process it, and then update the history file, TGHA.DAT, which is stored on the system disk. TGHA can also write into the ERROR.SYS file to convey MF20 error history information to SYSERR. Chapters 6 and 7 describe the use of TGHA, SYSERR, MEMCON, and the diagnostic programs for maintenance of the MF20.

4.3 MF20 PROGRAMMING CONSIDERATIONS

Paragraphs 4.3.1 through 4.3.11 describe the eleven diagnostic functions implemented in the MF20 hardware. Diagnostic functions are a set of special instructions that permit the program to access the MF20 logic in a nonstandard manner for one or more of the following purposes.

1. Loading a control RAM to configure the memory after power-up

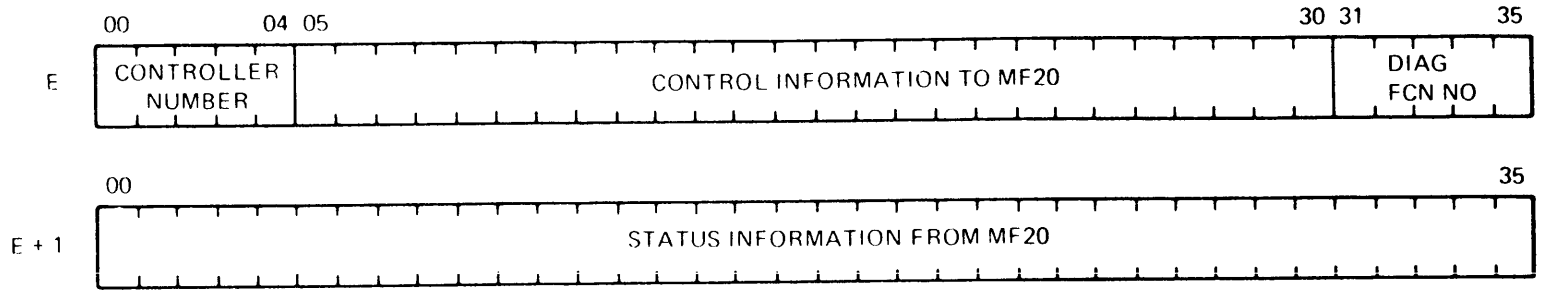
2. Reading error history flip-flops to log and analyze memory errors

3. Reading the state of key logic signals to diagnose memory faults

4. Setting up and activating the spare bit mechanism

Each diagnostic function is described in general with MACRO 10 examples included to show how it could be used by the programmer. Figure 4-3 describes the general format of the diagnostic function instructions.

Tables 4-1 through 4-11 provide a detailed description of the bit encoding for each of the eleven functions. Each table is separated into two groupings. The first group describes the bit encoding for [E], the control information sent to the MF20; and the second group describes the [E + 1] bit encoding, the status information returned from the MF20. Column 1 lists the bit position number, column 2 lists the logic print numbers where the function is implemented, and column 3 gives a brief synopsis of



SBDIAG E

WHERE:

- THE CONTENTS OF LOCATION E IS SET UP BY THE PROGRAM TO CONTAIN CONTROL INFORMATION TO BE SENT TO THE MF20.
  BITS 00-05 = 01bbb WHERE bbb IS THE MF20 UNIT NUMBER.
  BITS 31-35 = AN OCTAL NUMBER 00-12 THAT SPECIFIED THE DIAGNOSTIC FUNCTION TO BE EXECUTED.
- LOCATION E + 1 RECEIVES STATUS INFORMATION FROM THE MF20.

NOTES:

1. REFER TO TABLES 4-1 THROUGH 4-11 FOR THE DEFINITIONS OF THE CONTROL AND STATUS BITS.
2. UNUSED BITS IN E GET LOADED INTO THE MF20S PORT BUFFER, BUT HAVE NO EFFECT ON MF20 OPERATION.
3. UNUSED BITS IN E + 1 READ BACK AS 0S FROM THE MF20.

MR-1766

Figure 4-3 Diagnostic Function Format

the purpose of the bit. Since the diagnostic programs detect most hardware faults using diagnostic functions, the information in column 2 will be especially valuable to the service engineer to key the software to the physical hardware function.

4.3.1 Diagnostic Function 0 (Table 4-1)

This function permits activating an error clear signal that clears all MF20 error flags, and retrieving 36 bits of error/status information. The following example illustrates its use.

NOTE
All examples assume MF20 no. 0.

Example

```
HRLZI 0,210000      ;AC0 = 210000,,0
SETZM   1           ;AC1 = 0
SBDIAG  0           ;execute the function 0
TLNE  1,770000      ;skip if all flags cleared
PUSHJ  P,ERROR      ;go service error
continue .....
```

4.3.2 Diagnostic Function 1 (Table 4-2)

Function 1 is used to enable looping back data via the M8579 data multiplexer chips for diagnostic testing of the MF20 data path (refer to Figure 4-4). It also permits controlling a pair of status flip-flops that log the current state of the MF20. Ten bits of status are returned that specify controller type, loopback group enabled, and MF20 state. The following example illustrates how a program might use a function 1.

Example

```
HRLZI   0,200000
ADDI    0,1         ;AC0 = 200000,,1
SETZM     1         ;AC1 = 0
SBDIAG    0         ;execute the function 1
MOVS    2,1         ;set up ac2 to test it
ANDI    2,500
CAIE    2,500       ;skip if it is an MF20
JRST    NOMF20      ;jump if not an MF20
continue .....
```

Table 4-1 Diagnostic Function 0

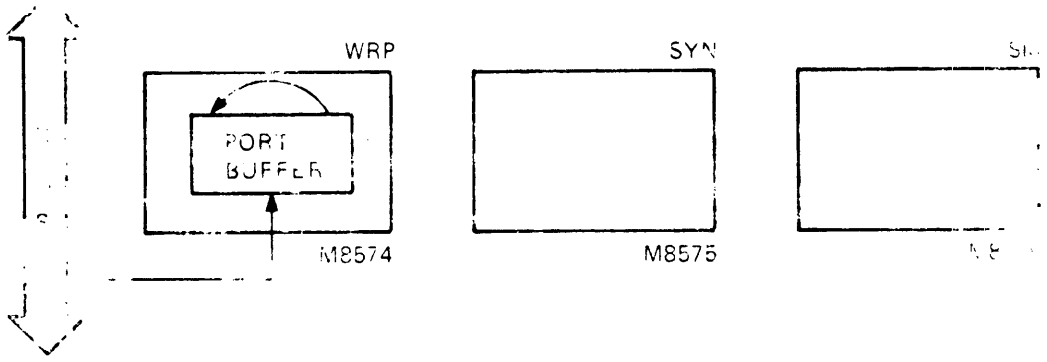
Bit No.	Print(s)	Description
[E] Control Information to MF20		
05	ADT9	Generates ADT9 CLR ERR that is used to clear MF20 error flags.
[E+1] Status Information from MF20		
00	ADT7,9	Asserted when one of the following RAM control errors occurs. BLK RAM ERR - Parity error when accessing the address response RAM TIM RAM ERR - Parity error when accessing the timing RAM SUB RAM ERR - Parity error when accessing the spare bit RAM A diagnostic function 2 is used to retrieve the individual error flag.
01	ADT7,9 SYN5	Asserted to indicate a RD PAR ERR which is a correctable error detected during a read cycle - this flag may be inhibited by setting the ICE bit in the spare bit RAM.
02	ADT7,9	INC RQ ERR - Asserted to indicate a memory request failed to complete its normal cycle sequence within 128 ticks of the basic clock.
03-05	ADT7,9	A 3-bit field that indicates various types of data-related errors. 03 RD PAR ERR - Either a correctable read error or a double-bit error (DBE) 04 WR PAR ERR - A data parity error was detected during a write 05 ADR ERR - Overall XBus address parity error - includes RQ 0-3 RD RQ, WR RQ
06-07	CTL7	A 2-bit field hardwired to always read back as 11_2 to indicate 4-way interleave mode.
08-11	CTL4,7	ERR RQ 0-3 - A 4-bit field that indicates which words were being requested when an error was detected.

Table 4-1 Diagnostic Function 0 (Cont)

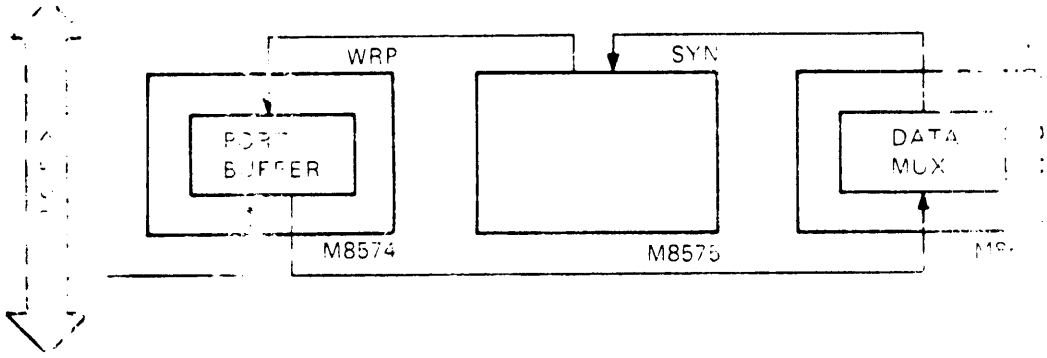
Bit No.	Print(s)	Description
12-/13	CTL4,7	A 2-bit field that indicates the type of memory request in progress when an error was detected. <12-13> = 01 Write cycle 10 Read cycle 11 Read-pause-write cycle
14-35	SYN7,9 CTL4 ADT1,9	These 22 bits indicate the XBus address being accessed when an error was detected.

Table 4-2 Diagnostic Function 1

Bit No.	Print(s)	Description
[E] Control Information to MF20		
12-14	WRP4	Used to activate storage module loopback mode. <12> = 1 Enable Loopback <13-14> = n n = 0, 1, 2 - The group number to loop back
25-28	ADT7,9	<28> = 1 Enables bits 26-28 to set conditions within the MF20 <26-27> = 00 MF20 has just powered up 01 All RAMS except the address response RAM have been loaded 10 Same as 01 but address response RAMs have been loaded to configure the MF20 11 Same as 10 except THGA has been run and completed its initialization tasks <25> - Used to permit software to place the MF20 off-line <25> = 0 Enable MF20 <25> = 1 Disable MF20
[E+1] Status Information from MF20		
08-11	WRP7	Hardwired to respond with a code of 05_8 to indicate the memory type is an MF20.
12-14	WRP7,4	Indicates state of the loopback control signals set in first half of diagnostic cycle.



(A) PORT LOOPBACK



(B) GROUP LOOPBACK

Figure 4-4 Data Loopback

4.3.3 Diagnostic Function 2 (Table 4-3)

Function 2 permits reading the personality PROM and also provides special diagnostic control for hardware fault analysis. The example shown below illustrates its use to access the PROM in MF20 no. 0, group 0, field 0, to test MOS chip size. (Refer to Figure 5-41.)

Example

```
        HRLZI   0,200030      ;AC0 = 200030,,2
        ADDI    0,2
        SETZM     1           ;AC1 = 0
        SBDIAG    0           ;execute the function 2
        MOVS    2,1           ;mask chip size bits
        ANDI    2,300
        TRC     2,300         ;complement chip size bits
        TRNN    2,300         ;4K chips
        PUSHJ   P,X4K         ;yes
        TRNN    2,200         ;16K chips?
        PUSHJ   P,X16K        ;yes
        TRNN    2,100         ;32K chips?
        PUSHJ   P,X32K        ;yes
X64K:   .....                 ;must be 64K
```

4.3.4 Diagnostic Function 3 (Table 4-4)

Function 3 is used to load the required bit patterns into the fixed value RAMs. The example below illustrates how a diagnostic might write and read back a RAM location for testing 1s.

Example

```
        HRLZI   0,200374      ;AC0 = 200374,,3
        ADDI    0,3
        SBDIAG    0           ;execute the function 3
        HRLZI   0,200000      ;AC0 = 200000,,3
        ADDI    0,3
        SETZM     1           ;AC1 = 0
        SBDIAG    0           ;read back loc. 000
        TLC     1,360         ;complement RAM data
        TLNE    1,360         ;skip if data all 1s
        PUSHJ P,RAMERR        ;go service error
        continue .....
```

Table 4-3 Diagnostic Function 2

Bit No.	Print(s)	Description
[E] Control Information to MF20		
09-12	CTL9 SM14	PROM select field used to select one of 12 possible PROM chips to be read. <9-10> Group select 0, 1, or 2 <11-12> Field select 00, 01, 10, or 11
13-14	CTL9 SM14	PROM word select - A 2-bit field used to select one of the first four locations in the PROM chip selected by <9-12>.
23-27	CTL4,7	When all 0s, this field indicates that the contents of bits D07-14 during the second half of the diagnostic cycle will be PROM data. When not all 0s, the bits are used to select which MF20 control signals will appear in D07-14 during the second half of the diagnostic cycle (diagnostic data).
[E+1] Status Information from MF20		
05-06	CTL4,7	ERR WD 2, 1 - This 2-bit field specifies which word in the quadword group caused an error.
07-14	CTL4,7	This 8-bit field displays either PROM data or diagnostic data dependent on the setting of D23-27 during the first half of the diagnostic cycle. (Refer to Paragraph 5.8.7)
20-27	ADT2,9	This 8-bit field displays the contents of the MOS ADR 0-7 register during single-step diagnostic operations.
28-30	ADT7,9	This 3-bit field displays the type of control RAM error flagged by bit 00 in a diagnostic function 0. 28 = 1 Timing RAM error 29 = 1 Spare bit RAM error 30 = 1 Address response RAM error

Table 4-4 Diagnostic Function 3

Bit No.	Print(s)	Description
[E] Control Information to MF20		
10-15	CTL4,5	Used by the program to load or access the fixed value RAMs as follows. 14 = 1 Enable loading ACKN RAM (bit 10) 15 = 1 Enable loading DATA VALID RAM (11-13) 10 P ACKN EN bit 11 Set DATA VALID bit 12-13 P RD ADR 34-35 bits
20-27	CTL4,5,7	Used by the program to establish the desired RAM addresses when loading or accessing the fixed value RAMs.
[E+1] Status Information from MF20		
10-13	CTL4,5,7	This 4-bit field displays the contents of the fixed value RAM address specified by D20-27 during first half of the diagnostic cycle. 10 CTL4 PACK EN L 11 CTL5 SET DATA VALID RAM H 12-13 CTL5 P RD ADR 34-35 H

4.3.5 Diagnostic Function 4 (Table 4-5)

Function 4 permits testing the MF20 control logic using single-step operations. It also allows the program to set up and monitor the refresh control logic. This is the function used by the diagnostic program to activate port loopback mode. (Refer to Figure 4-4.) The example shown below would be used to set up the refresh interval. (Refer to Paragraph 5.8.9.)

Example

```
HRLI    0,200000    ;AC0 = 200000,,021644
HRRI    0,021644
SBDIAG    0         ;execute function 4
                    ;set interval = 29
```

4.3.6 Diagnostic Function 5 (Table 4-6)

A function 5 would be used in conjunction with a function 4 to allow the diagnostic program to single-step the MF20 control logic. After stepping to the desired test point the function 5 returns the state of the row address strobe (RAS) signals. The example below tests to verify that all RAS signals are inactive.

Example

```
HRLZI  0,200000     ;AC0 = 200000,,5
ADDI   0,5
SETZM    1          ;AC1 = 0
SBDIAG   0          ;execute function 5
TLNE   1,7400       ;skip if all RAS = 0
PUSHJ  P,RASERR     ;jump if any are on
continue .....
```

4.3.7 Diagnostic Function 6 (Table 4-7)

Like functions 4 and 5 this function is used strictly for hardware diagnosis. It provides seven diagnostic subfunctions that permit extensive diagnostic operations to be executed under program control. The example below illustrates how the diagnostic might verify the ECC complement register.

Example

```
HRLI   0,207774     ;AC0 = 203774,,2006
HRRI   0,2006
SETZM    1          ;AC1 = 0
SBDIAG   0          ;execute function 6
TLC    1,3770       ;complement ECC data
TLNE   1,377        ;skip if ECC = 1s
PUSHJ  P,ECCERR     ;jump if not all 1s
continue .....
```

Table 4-5 Diagnostic Function 4

Bit No.	Print(s)	Description
[E] Control Information to MF20		
05	WRP0	Enables port data loopback mode of operation in the selected MF20.
09-12	CTL2,3 ADT5	Provides program control of MF20 clocking during single-step diagnostic operations. 09 - CTL2 SIM A PHS COM 10 - CTL2 SIM B PHS COM 11 - CTL3 P DATA VALID IN 12 - ADT5 REFRESH NOW
14-15	ADT6 CTL2	Used to enable single-step mode and gated clocks. 14 ADT6 CLK GO 15 CTL2 SINGLE STEP
20	ADT5	Enables activating the refresh logic timing chain by asserting the signal ADT5 REFRESH NOW L.
21	ADT3	Enables refresh logic.
22	ADT3	Enables loading refresh interval latches.
24-30	ADT3	This 7-bit field permits the program to establish the refresh interval. It is loaded into the refresh interval latches if bit 22 = 1.
[E+1] Status Information from MF20		
09-10	CTL7	Used to display the states of the simulated A and B phase coming clocks set up during first half of the diagnostic cycle.
15	ADT9	Used to display the state of the single-step mode control signal. 1 = ON 0 = OFF
21	ADT9	Used to display the state of the refresh control logic. 1 = ON 0 = OFF
23-30	ADT9	This 8-bit field displays the current contents of the refresh interval latches.

Table 4-6 Diagnostic Function 5

Bit No.	Print(s)	Description
[E] Control Information to MF20		
06-13	CTL1	An 8-bit field used during diagnostic single-step operations to activate the MF20 control and timing logic. 06 CT01 DIAG START A 07 CT01 DIAG START B 08-11 CT01 P RQ 0-3 12 CT01 P RD RQ 13 CT01 P WR R0
15	CTL7	Enables generating CTL7 DIAG CYC RQ HLD to hold control information during single-step diagnostic mode.
20-27	SYN7	This 8-bit field is used to provide the address to the address response RAM during single-step diagnostic control operation.
29-30	CTL1	This 2-bit field selects which word in a quadword group is being referenced during the current step of a diagnostic single-step operation.
[E+1] Status Information from MF20		
24-27	ADT6,9	This 4-bit field is used to display the state of the row address strobe (RAS) signals ADT6 MOS RAS 0-3.

Table 4-7 Diagnostic Function 6

Bit No.	Print(s)	Description
[E] Control Information to MF20		
07-14	WRP7 SYN2	An 8-bit field used to set up the diagnostic data used to control SYN2 M TO CHK ECC 32, 16, 8, 4, 2, 1, PAR to be used as specified by the diagnostic subfunction specified in D25-27
25-27	WRP7	Specifies one of eight subfunctions performed by a diagnostic function 6 25-27 = 0 Read the ECC register on WRP7. 25-27 = 1 Read the syndrome buffer register on the SYN board. 25-27 = 2 Select diagnostic bits 07-13 in place of MOS bits 36-42, force 0s on 00-35, run a correction pass, and return 00-35. 25-27 = 3 XNU 25-27 = 4 Write the ECC complement register if D15 = 1, then read it back. 25-27 = 5 Write the ECC complement register, then enable it to be sent to memory in place of D36-42 on the next write cycle. 25-27 = 6 Read the output of the D36-42 mixer. 25-27 = 7 Enable latching of D36-42 mixer after next write.
[E+1] Status Information from MF20		
07-14	SYN9	This 8-bit field displays diagnostic data as a function of the subfunction specified by D25-27 in the first half of the diagnostic cycle.
24-27	ADT5,9	A 4-bit field that displays the state of the column address strobe (CAS) signals ADT5 MOS CAS 0-3.

4.3.8 Diagnostic Function 7 (Table 4-8)

This function permits controlling the bit substitution RAM. It also returns the state of the MOS write enable signals during single-step diagnostic operations. The example below loads location 021_8 in the bit substitution RAM to specify bit position 15_8 and inhibit reporting correctable errors.

Example

```
HRLI    0,200674        ;AC0 = 200674,,2407
HRRI    0,002407
SBDIAG    0             ;execute function 7
continue .....
```

4.3.9 Diagnostic Function 10 (Table 4-9)

Function 10 is primarily concerned with setting up and monitoring the power supply margin control logic. It is also used to clear the dc bad flip-flop on initial power-up, which illuminates a LED on the MF20 backplane. The example shown below tests the state of the power supply flag and clears it if it is set.

Example

```
HRLZI   0,200000        ;AC0 = 200000,,10
ADDI    0,10
SETZM     1             ;AC1 = 0
SBDIAG    0             ;read flag
TLNE    1,400           ;skip if off
PUSHJ   P,OK
IORI    0,400           ;AC0 = 200000,,410
SBDIAG    0             ;clear it - light LED
```

4.3.10 Diagnostic Function 11 (Table 4-10)

This function is used to set up and monitor the timing RAM. The example shown illustrates how the diagnostic might use it to test the RAM.

Example

```
HRLI    0,200000        ;AC0 = 200000,,777411
HRRI    0,777411
SBDIAG    0             ;execute function 11
HRRI    0,600011        ;AC0 = 200000,,600011
SETZM     1             ;AC1 = 0
SBDIAG    0             ;read back loc. 003
TLC     1,077400        ;complement RAM data
TLNE    1,77400         ;skip if data was 1s
PUSHJ   P,RAMERR        ;jump if not
continue .....
```

Table 4-8 Diagnostic Function 7

Bit No.	Print(s)	Description
[E] Control Information to MF20		
07-14	WRP8	This 8-bit field specifies the information to be written into the spare bit RAM. 07-12 Bit position number 13 Flag to inhibit reporting correctable errors (ICE bit) 14 Parity bit (odd)
15		When set, enables write to the spare bit RAM.
21-27		An 8-bit field that holds the address of the spare bit RAM location to be accessed.
[E+1] Status Information from MF20		
07-14	WRP8	An 8-bit field that displays the contents of the addressed spare bit RAM location.
24-27	ADT5,9	A 4-bit field that displays the current state of the write enable signals to the MOS chips ADT5 MOS WE 0-3.

Table 4-9 Diagnostic Function 10

Bit No.	Print(s)	Description
[E] Control Information to MF20		
07-10	SYNA	A 4-bit field that specifies the direction of voltage margining. 1 = margin high 0 = margin low 07 +12 V MARG 08 +5 V MARG 09 -2 V MARG 10 -5.2 V MARG
11-14	SYNA	A 4-bit field that enables voltage margining. 11 +12 MARG EN 12 +5 MARG EN 13 -2 MARG EN 14 -5.2 MARG EN
15	SYNA	Enables loading D07-14.
26	SYN9	Inhibits ECC correction when set.
27	SYNA	Enables clearing the DC CHK PWR BAD flip-flop which also lights a LED on the backplane.
[E+1] Status Information from MF20		
07-10	SYNA	Displays the state of the margin control direction flip-flops set up during the first half.
11-14	SYNA	Displays the state of the margin control enable flip-flops set up during the first half.
26	SYN9	Displays the state of the ECC correction disable flip-flop.
	SYNA	Displays the state of the DC CHK PWR BAD flip-flop.

Table 4-10 Diagnostic Function 11

Bit No.	Print(s)	Description
[E] Control Information to MF20		
13-19	ADT4	A 7-bit field used to specify a timing RAM address to be accessed.
20	ADT4	Enables writing D21-27 into the timing RAM location addressed by D13-19.
21-27	ADT4	A 7-bit field that contains the data to be written into the timing RAM. 21 RAS 22 CAS 23 RAM PAR 24 WE 25 ADR 2nd HALF 26 DATA RDY 27 RAM BUSY CLR The bit patterns loaded into the timing RAM control the MOS read/write cycle timing when accessing the MF20.
[E+1] Status Information from MF20		
21-27	ADT4	A 7-bit field that displays the contents of the addressed timing RAM location.
29	ADT2,9	This bit is activated by a backplane jumper to specify relative MOS chip size.
30	ADT5,9	This bit displays the state of the signal DESELECT CYC EN.

4.3.11 Diagnostic Function 12 (Table 4-11)

This function is used to set up and monitor the address response RAM. The following example illustrates how the program would set the deselect bit in RAM location 5.

Example

```
HRLI    0,201014        ;AC0 = 201014,,2412
HRRI    0,2412
SBDIAG    0             ;execute function 12
```

Table 4-11 Diagnostic Function 12

Bit No.	Print(s)	Description
[E] Control Information to MF20		
08-14	SYN7	An 8-bit field that contains the data to be loaded into the address response RAM. 8 BLK ADR PAR set to establish odd parity on D8-14. 9 TYPE SELECT used to specify one of two possible MOS chip sizes (16K or 4K, 64K or 16K etc.) 10-11 Group no. 0-2 12-13 Block no. 0-3 14 -BOX SELECT set to inhibit response to specific XBus addresses
20-27	SYN7	An 8-bit field that contains the address of the address response RAM location being accessed.
[E+1] Status Information from MF20		
08-14	SYN7	An 8-bit field that displays the contents of the location being accessed in the address response RAM.

CHAPTER 5
TECHNICAL DESCRIPTION

5.1 INTRODUCTION

This chapter describes how an MF20 works. It is written using a "top down" approach that explains how the MF20 functions at varying levels of complexity. Emphasis is placed on describing the unit as an organized set of functional logic areas interacting to perform well defined system functions required for information storage and retrieval. Physical logic partitioning and key signal generation are stressed through the use of simplified illustrations, tables, flowcharts, and timing diagrams. When reading this chapter, a copy of the MF20 Engineering Drawings should be kept available for reference to detailed logic operations.

5.1.1 MF20 Conceptual Block Diagram (Figure 5-1)

The basic functions of a memory system are generally simple and well defined. It follows that the hardware designed to support these functions should exhibit a logical and well organized structure. This paragraph defines and discusses the structure of the MF20 from a conceptual viewpoint. This conceptual approach is used throughout the technical description.

The MF20 can be divided into the following functional areas.

1. MOS storage array
2. Control section
3. Data path
4. Master oscillator
5. Power supply
6. XBus

Each area performs a specific function involved with the task of information transfer between a MOS storage array and a KL10 CPU (EBox/MBox). Understanding these functions and how they interact to perform the overall system function is essential to comprehension of MF20 operation.

5.1.1.1 MOS Storage Array - A single MF20 can store up to 768K (786,432) PDP-10 words in 256K (262,144) increments called groups. The storage array provides the storage medium to achieve this function using a matrix of 16K (16,384) X 1 MOS chips. Each location within the array is uniquely identified and may be written (storage) or read (retrieval) under control of programmed instructions executed by the EBox.

5.1.1.2 Control Section - The control section generates the required sequence of timing and control signals that performs the following functions.

1. Selects the locations within the MOS storage array to be read or written

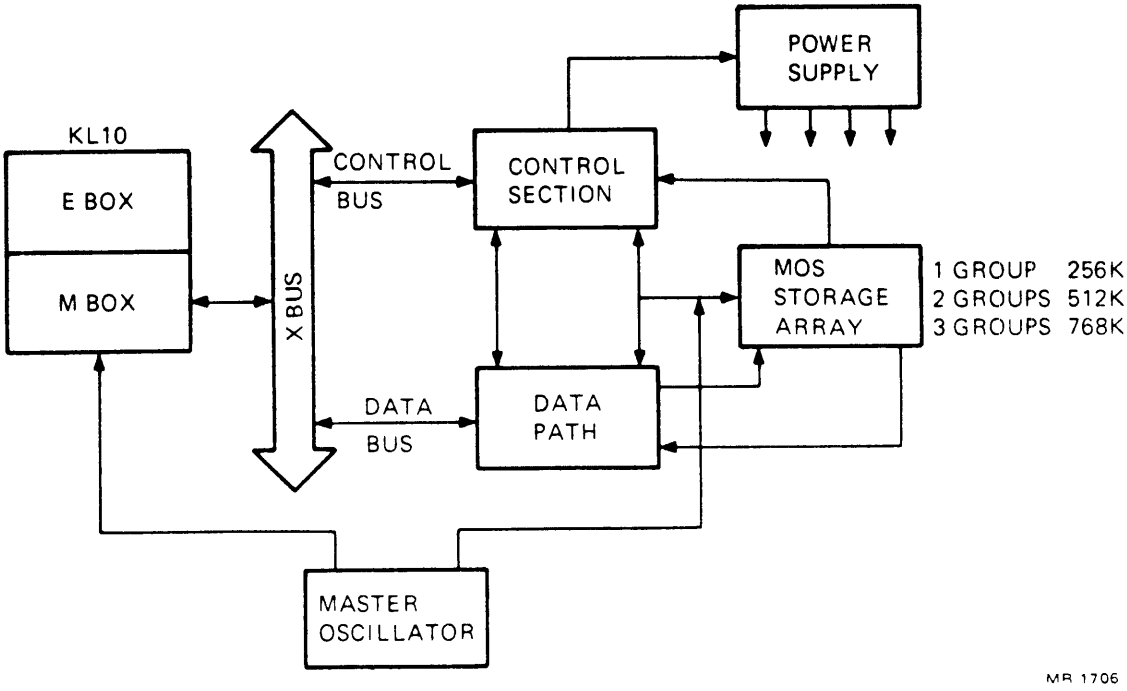


Figure 5-1 MF20 Conceptual Block Diagram

2. Results in an orderly movement of data between the XBus and MOS storage array via the data paths

3. Can margin the power supply voltages under program control.

The control section consists of a network of both combinational logic and control RAMs that are activated by specific control signals received from the MBox via the XBus (control bus).

5.1.1.3 Data Paths - Acting as a buffer between the XBus (data bus) and the MOS storage array, the data paths provide the following functions.

1. Temporary storage of all data entering or leaving the MF20

2. Generation of ECC check bits prior to storing data in the MOS storage array (write)

3. Checking of ECC check bits when retrieving data from the MOS storage array (read)

4. Automatic data correction of single-bit errors during a read

5. Spare bit substitution

Consisting of a network of interconnected receivers, drivers, registers, and multiplexers, the data paths are functionally static and depend on the control section to move data from the XBus to the MOS storage array or from the MOS storage array to the XBus.

5.1.1.4 Master Oscillator - To synchronize operations between the MBox in the CPU and the MF20, a master oscillator is required to provide a fixed timing reference. The oscillator provides crystal-controlled clocks (30 MHz nominal) that synchronize both the CPU and memory.

5.1.1.5 Power Supply - Each MF20 requires a source of dc operating voltages for the circuits that comprise the functional logic areas. The power supply includes a mechanism to allow the program to margin the dc voltage output to the MF20 logic so as to isolate marginal component failures. It also contains a battery backup supply that can provide power to keep the MF20 operating for approximately 30 seconds following power failures.

5.1.1.6 XBus - The XBus provides the physical connection between the CPU (MBox) and the MF20. It transports a set of control and data signals between the two units. The dialogue that must occur between the CPU and MF20 is rigidly defined for the following functions.

1. Memory read cycle
2. Memory write cycle
3. Memory read-modify-write cycle
4. Diagnostic cycle

5.1.2 Summary

Conceptually, the MF20 consists of just three major areas: storage array, control section, and data paths. Understanding how these areas interact is the key to explaining how the MF20 works. The remainder of Chapter 5 will build on this logical structure to explain the operation of the MF20 MOS Memory Subsystem.

5.2 MF20 MODULE BLOCK DIAGRAM (Figure 5-2)

MOS storage array, control section, and data paths are contained on from eight to sixteen extended-width, hex-height modules in the MF20. The MOS storage array consists of up to three sets of M8579 storage modules (4 modules per set). The control section is primarily contained on the M8577 address and timing module and the M8576 control and timing module. The M8574 write path module and the M8575 syndrome module form the data paths.

5.2.1 Storage Module - M8579

Each storage board within a selected group (a physical partition of 256K words) stores an 11-bit slice of the data path called a field. Four storage modules are accessed simultaneously to provide a total data path width of 44 bits as described below.

D00-35	36 Data bits
D36-42	7 Check bits
D43	1 Spare bit

The total storage capacity of a single group of four M8579 boards is variable and a function of the MOS chip size.

Chip Size	Capacity Per Group
4K	64K (65,536)
16K*	256K (262,144)
32K	512K (524,288)
64K	1024K (1,048,576)

*When capacity is referenced throughout the remainder of this section it will assume the 16K chip size.

Physically, the storage modules for each field are identical except for the bit positions each one supports. Bit position is established by the backplane slot the modules plug into. Since the logic elements on each board are identical, the storage boards are described by a single set of 15 engineering drawings labeled SM00-SM14. The logic partitioning in the prints is as follows.

SM00-10 11 identical prints that describe each bit in the 11-bit field.

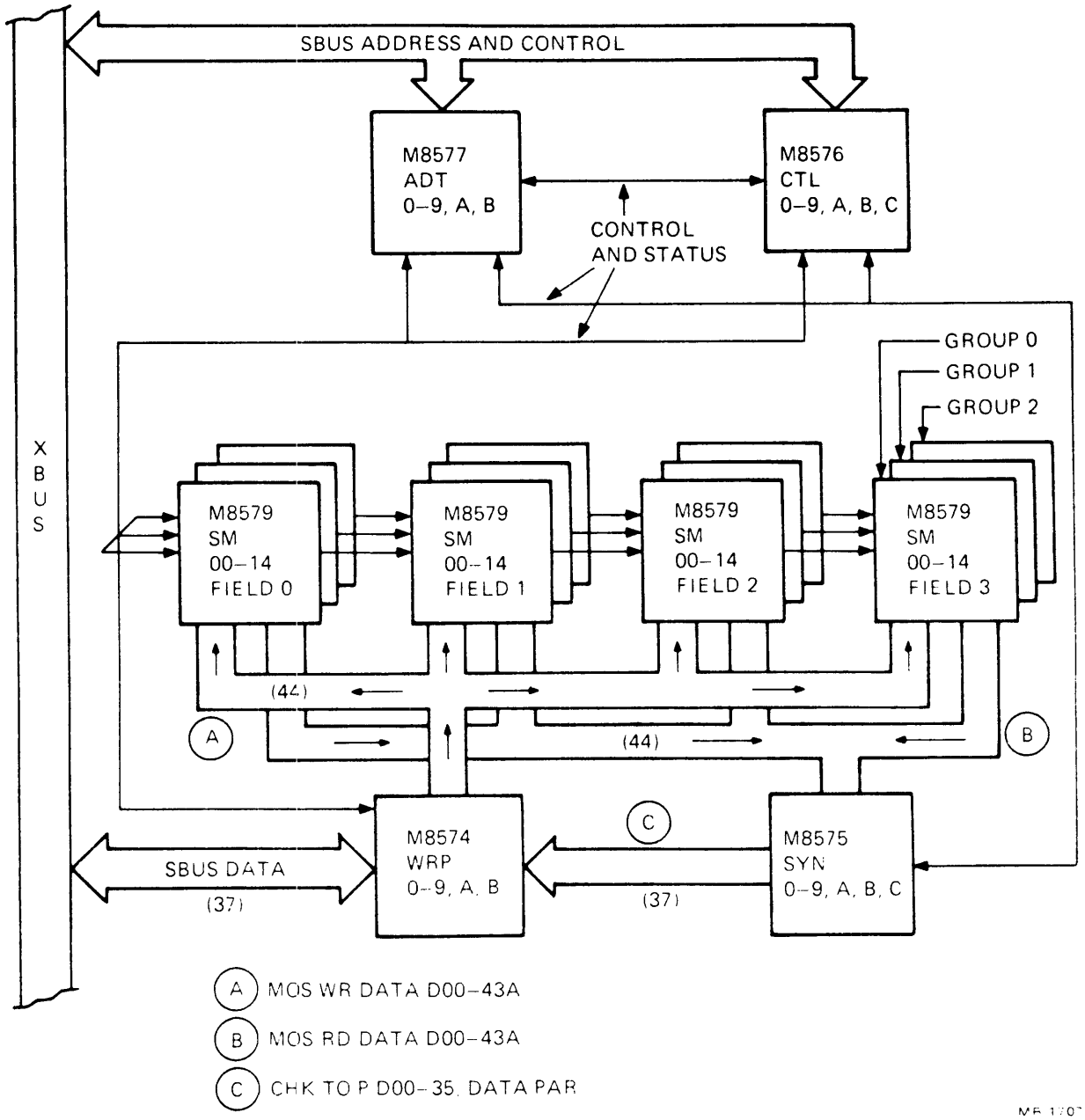


Figure 5-2 Module Block Diagram

SM11 MOS Storage Write Pulse Logic
SM12 MOS Storage Row Address Strobe Drivers
SM13 MOS Storage Column Address Strobe Drivers
SM14 MOS Address and Data Control Logic plus Personality PROM

During a write cycle the storage modules receive 44 bits from the M8574 over an internal data bus, MOS WR DATA 00-43A, and store this information in the MOS chips under control of the M8577 and M8576 modules. For a read the storage modules send 44 bits to the M8575 module over a separate internal data bus, MOS RD DATA 00-43A, under strict control of the M8576 and M8577 control modules.

5.2.2 Write Path - M8574

All data entering or leaving the MF20 passes through the M8574 module. Print prefixes WRP0 through WRP9 identify each of the drawings in the set of ten that describes the write path. The major logic within the M8574 is partitioned as follows.

WRP0-1 XBus Data Bus Drivers and Receivers
WRP2-4 Port Data Buffer and Input-Output Mixers
WRP5 Data Drivers to Storage Modules
WRP6 ECC Check Bit Generator
WRP7 ECC Bit Mixers and Diagnostic Registers
WRP8 Spare Bit Substitution RAM
WRP9 Spare Bit Mixer

During a write cycle the WRP module accepts 37 bits from the XBus data bus, (36 data bits and 1 parity), generates the appropriate ECC check bits, and selects the correct spare bit (if active). It then sends 44 bits to the selected group of M8579 modules over the internal data bus, MOS WR DATA 00-43A, to be written into the proper location within the MOS storage array. If a read cycle is being executed, the M8574 module accepts 37 bits of checked and corrected data (36 data bits and 1 parity) from the M8575 module over a separate internal data bus, CHK TO P D00-35, DATA PAR. The M8574 clocks the data into the port buffer register and transmits it back to the MBox via the XBus data lines. For diagnostic operations, the WRP module simply accepts and distributes control information from the XBus, or collects and transmits error/status information to the XBus.

5.2.3 Syndrome - M8575

This module contains most of the logic that supports the read data path. Print prefixes SYN0 through SYNA identify each of the eleven prints that describe the syndrome module. The major logic areas are partitioned as shown below.

SYN0 Spare Bit Selection Decoder, XBus Address Transceivers (14-21), and the Spare Bit Receiver

SYN1 Data Spare Bit Mixers

SYN2 ECC Spare Bit Mixers, ECC Register

SYN3 Syndrome Generator

SYN4 Syndrome Calculator and Buffer

SYN5 Syndrome Correction Decoders

SYN6 Data Correct and Distribution

SYN7 Address Response RAM, Error Register

SYN8-9,A Diagnostic Function Control

During a read the M8575 module receives a 44-bit word from the storage modules via an internal data bus, MOS RD DATA 00-43A. After substituting the spare bit the data is checked, corrected (if possible), and distributed to the WRP module over a separate 37-bit internal bus, CHK TO PD00-35, DATA PAR. From there the data is transmitted back to the MBox via the XBus.

If a single-bit error (SBE) is encountered, the syndrome module corrects it and saves the information (called the syndrome) necessary to indicate the bit position in error. A flag is set in the M8577 that causes the control section to activate an XBus error flag to notify the MBox. Double-bit errors (DBEs) are detectable but uncorrectable. In these cases the MBox is flagged that the data just received is suspect. If more than two bits are in error the results may be undefined.

5.2.4 Address and Timing - M8577

The M8577, the first in a pair of control boards, contains most of the logic involved with addressing the MOS storage array. Ten prints, prefixed ADT0 through ADT9, describe the functions performed within the M8577.

ADT0 XBus Address Receivers
ADT1 Port Address Registers
ADT2 Port Address Mixers
ADT3 Refresh Cycle Timer
ADT4 Timing RAM
ADT5 MOS Storage Array Timing Drivers and Refresh Control
ADT6 Clock Control
ADT7 Error Handling Logic
ADT8-9 Diagnostic Function Control

The primary functions performed by the ADT module are as follows.

1. Accept an address from the XBus and distribute it to the storage modules to specify one of 16,384 locations within a set of MOS chips

2. Generate the required sequence of timing pulses to properly operate the MOS chips

3. Collect and store all error conditions detected within the MF20

5.2.5 Control and Timing - M8576

This module contains all the timing and control logic not covered by the M8577. Eleven prints with prefixes CTL0 through CTLA describe the following functional logic areas.

CTL0	XBus Control Signal Receivers
CTL1	Start Logic
CTL2	Read/Write Cycle Control and Timing
CTL3	Write Data Mover
CTL4	Error Register and ACKN RAM
CTL5	Read Data Mover
CTL6	Data Control Logic
CTL7	Diagnostic Mixer
CTL8	Diagnostic Control
CTL9	SM PROM Control
CTLA	SM Selection Logic

Where the ADT module was concerned primarily with controlling the MOS address, the CTL module is primarily concerned with controlling the orderly movement of data into and out of the MF20. It accepts specific XBus control signals and initiates the proper control and timing sequences as outlined below.

1. Determines whether to execute a read, write, or diagnostic cycle sequence

2. Determines the number of words to be read or written

3. Times the proper loading of data from the XBus into the WRP module or gating data out of the WRP module onto the XBus

4. Controls the movement of up to 4 words (quadword), one at a time, from the XBus to the storage modules during a write or from the storage modules to the XBus during a read

5.2.6 Summary

A single MF20 can support up to three sets of M8579 modules to provide a total storage capacity of 768K (786,432) PDP-10 words. The data path between the storage array and the XBus data bus is contained on two modules, the M8574 write path and M8575 syndrome. Two control modules, the M8577 address and timing module and the M8576 control and timing module, contain all the timing and control logic required to move data between the XBus and the storage modules via the WRP and SYN modules under control of the MBox.

5.3 STORAGE ARRAY ORGANIZATION

Before continuing with more detailed descriptions of the MF20, several terms need to be understood. These definitions concern how the MOS storage array is organized; therefore, they are used in discussing any part of the MF20 that relates to accessing data within the MOS array.

5.3.1 Storage Units

As discussed earlier, a single MF20 can store up to 768K (786,432) 44-bit words, a grand total of 34,603,008 bits. Each one of these bits must be assigned to a single storage element, and this assignment must uniquely identify each bit in the storage array. Once the assignments have been made, the control section can be designed to permit the program to access any one of these millions of storage elements. To aid the design, these storage elements are grouped into more manageable units. The following definitions describe the groupings chosen.

Group - A set of four M8579 storage modules. A single MF20 can control up to three groups, a total of twelve storage modules.

Block - A set of 176 MOS storage elements (chips). A group consists of 4 blocks.

Sub-block - A set of 44 MOS storage elements. A block consists of 4 sub-blocks.

Field - An 11-bit slice of the MF20's internal data path. Each M8579 storage module delivers 11 bits (1 field) when accessed. Four M8579 modules are required to provide storage for a 44-bit word.

5.3.2 Storage Array Access

When the MBox wishes to retrieve data from the MF20, it places a 22-bit address on the XBus and initiates a memory cycle. The MF20 stores the address in its control sections and utilizes it as described in Figure 5-3. It is important to recognize that software has the ability to load the address response RAM, which makes the relationship between storage unit and physical MOS chip a program-controlled variable. This process is more completely described in Paragraph 5.8.4.

5.3.3 Summary

Several new terms have been defined to describe how the MF20 storage modules are organized. Understanding what is meant by the terms group, block, sub-block, and field is a prerequisite for understanding the detailed descriptions in subsequent paragraphs.

5.4 FUNCTIONAL BLOCK DIAGRAMS

This paragraph examines more closely the functional areas of logic within each module. It emphasizes how the logic within the boundaries of each module is partitioned into specific functional areas and how these areas interact to contribute to the overall

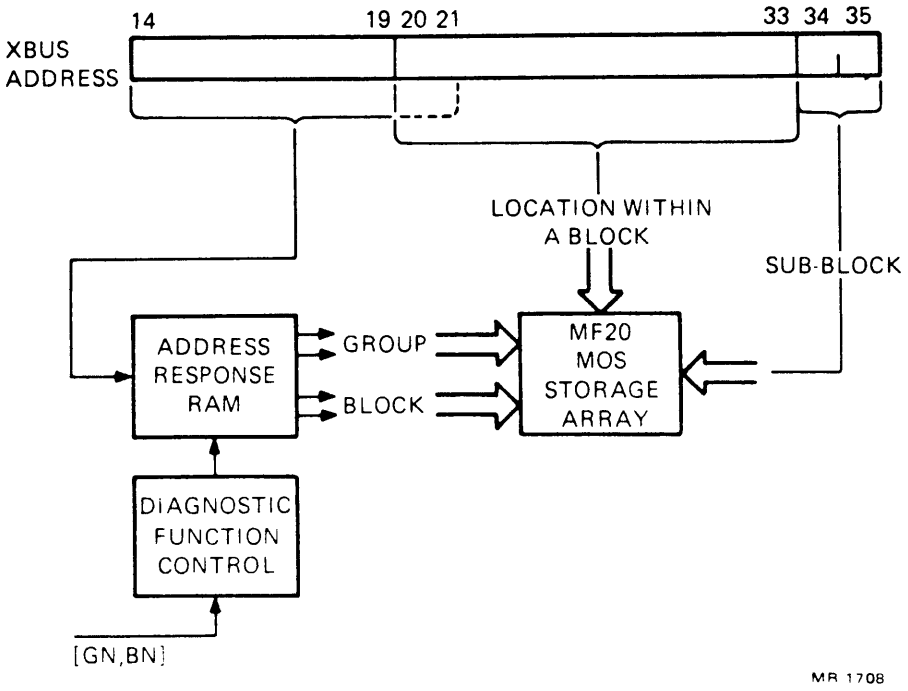


Figure 5-3 Storage Array Access

MF20 operation. The location and function of key signals entering and leaving each module are stressed.

5.4.1 M8579 Storage Module (Figure 5-4)

The M8579 can be partitioned into the following functional areas.

1. MOS RAM array
2. Data multiplexers
3. Address, data, and timing control
4. Personality PROM

The MOS array consists of 176 16K MOS chips arranged in a rectangular 16 X 11 matrix. There are eleven identical prints, SM00-SM10, each of which describes a 1-bit slice of the data path. All data entering or leaving the MOS array is buffered by a set of DC008 data multiplexer chips. Forty-four multiplexer chips are required (11 per M8579) to perform the dual functions of:

1. Quadword buffering during read and write

2. ECL to TTL translation for a write and TTL to ECL translation for a read.

To permit transferring data between the MOS storage array and the data multiplexer chips, the following signals must be activated in a definite sequence.

1. RAM address - 14 bits to specify one of 16K locations to be read or written. (This is applied in two 7-bit units.)

2. Row address strobes (RAS)

3. Column address strobes (CAS)

4. Write enable signals during write only (WE)

These functions are handled by the address and timing logic shown on SM11-14.

Getting data into and out of the storage module is the task of the data multiplexer chip control logic. In addition to normal read/write data control, the MOS WR DATA bus can be looped back via the multiplexer chips to test the integrity of the connections between the WRP and SYN modules. This is achieved via a diagnostic function operation.

The final area is the personality PROM. During manufacture the locations within this PROM are set up to identify the M8579 and its unique characteristics. Software can read this information to track error statistics for each individual storage module manufactured. Refer to Paragraph 5.8.7 for more detail on the PROM content.

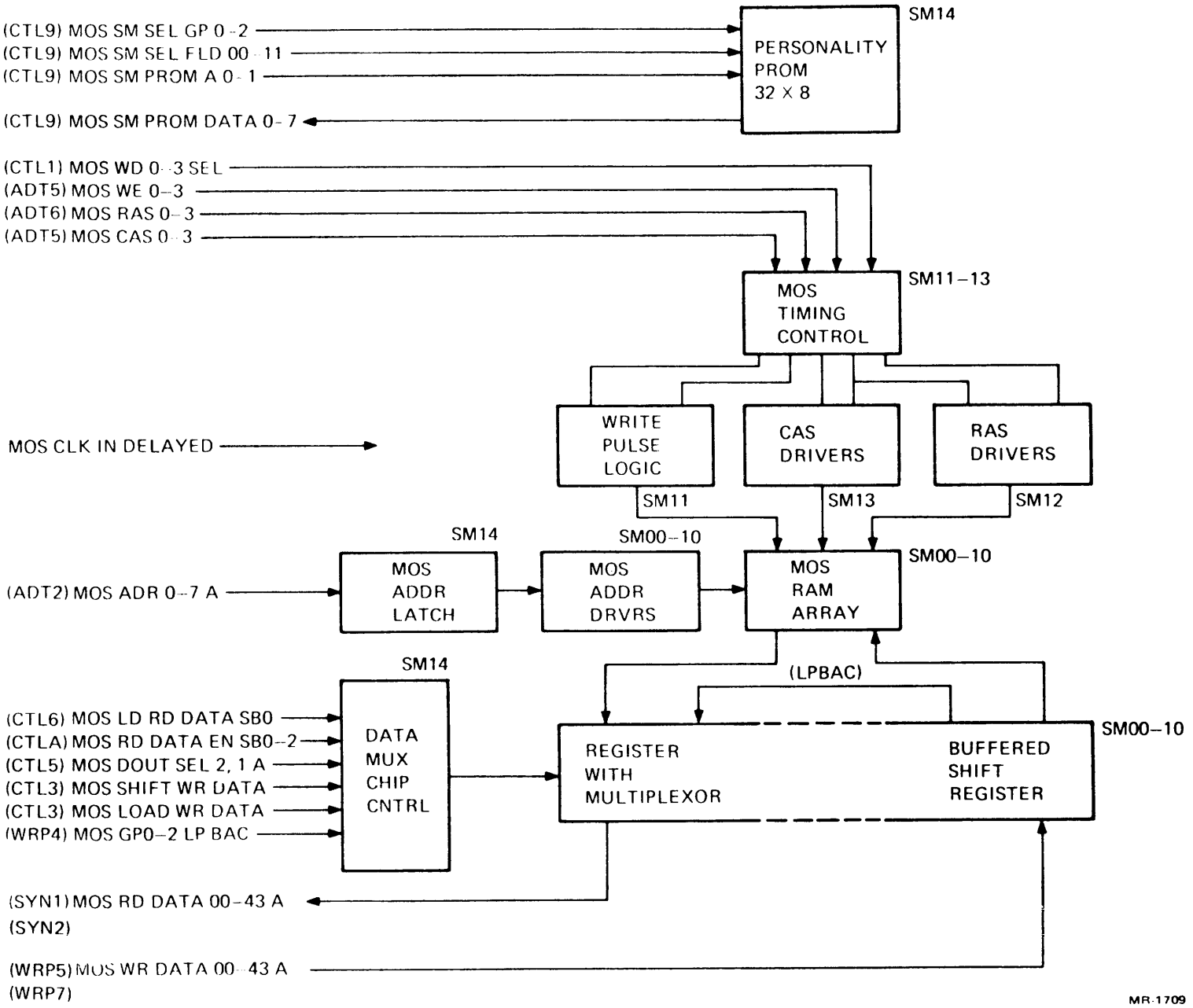


Figure 5-4 M8579 Functional Block Diagram

Table 5-1 describes the functions of the key signals within the intermodule interface. Paragraph 5.8.3 describes in more detail the logic operations within the storage modules.

5.4.2 M8574 Write Path (Figure 5-5)

The WRP module forms the focal point for all data entering or leaving the MF20 via the XBus data bus. It contains the logic to support the following functions.

1. Port data mixer and buffer
2. Check bit generation (write)
3. Spare bit control
4. Diagnostic function control

During a write operation the MBox sends 36 bits of data plus parity over the XBus to the XBus transceivers where it is steered into the port buffer via a data mixer. At this point seven check bits are generated and combined with the 36 data bits to become MOS WR DATA D00-42A. If directed by the spare bit substitution RAM, any single data or ECC bit can be written into the spare bit chip as MOS WR DATA D43A.

For read operations the port data mixer and buffer is switched to receive the data from the SYN module over the internal data bus CHK TO P D00-35, DATA PAR. Once stored in the port buffer it can be gated back to the MBox via the XBus transceivers.

The remaining function is to load the port buffer with control information during the first half of a diagnostic function and distribute this to the correct points in the MF20. This uses the same path as for a write, except that MOS WR DATA D00-35 reflects control information and is not stored in the MOS. During the second half of a diagnostic function, the port data mixer/buffer collects the diagnostic error/status information and transmits it back to the MBox over the XBus.

Table 5-2 describes the function of the key signals within the intermodule interface. Paragraphs 5.8.1 and 5.8.2 give detailed descriptions of the ECC generation and spare bit substitution mechanisms.

Table 5-1 M8579 Storage Module

Source/Signal Name	Destination Print	Function
(SM00-SM10) MOS RD DATA 00-43A	SYN0 SYN1 SYN2	A 44-bit wide internal data bus that transports the information at the output of the data multiplexer chips to the M8575 module.
(SM14) MOS SM PROM DATA 0-7	CTL9	An 8-bit wide internal bus that transports the information stored in the personality PROM to the M8576 module when the PROM is accessed during execution of a diagnostic function 2.

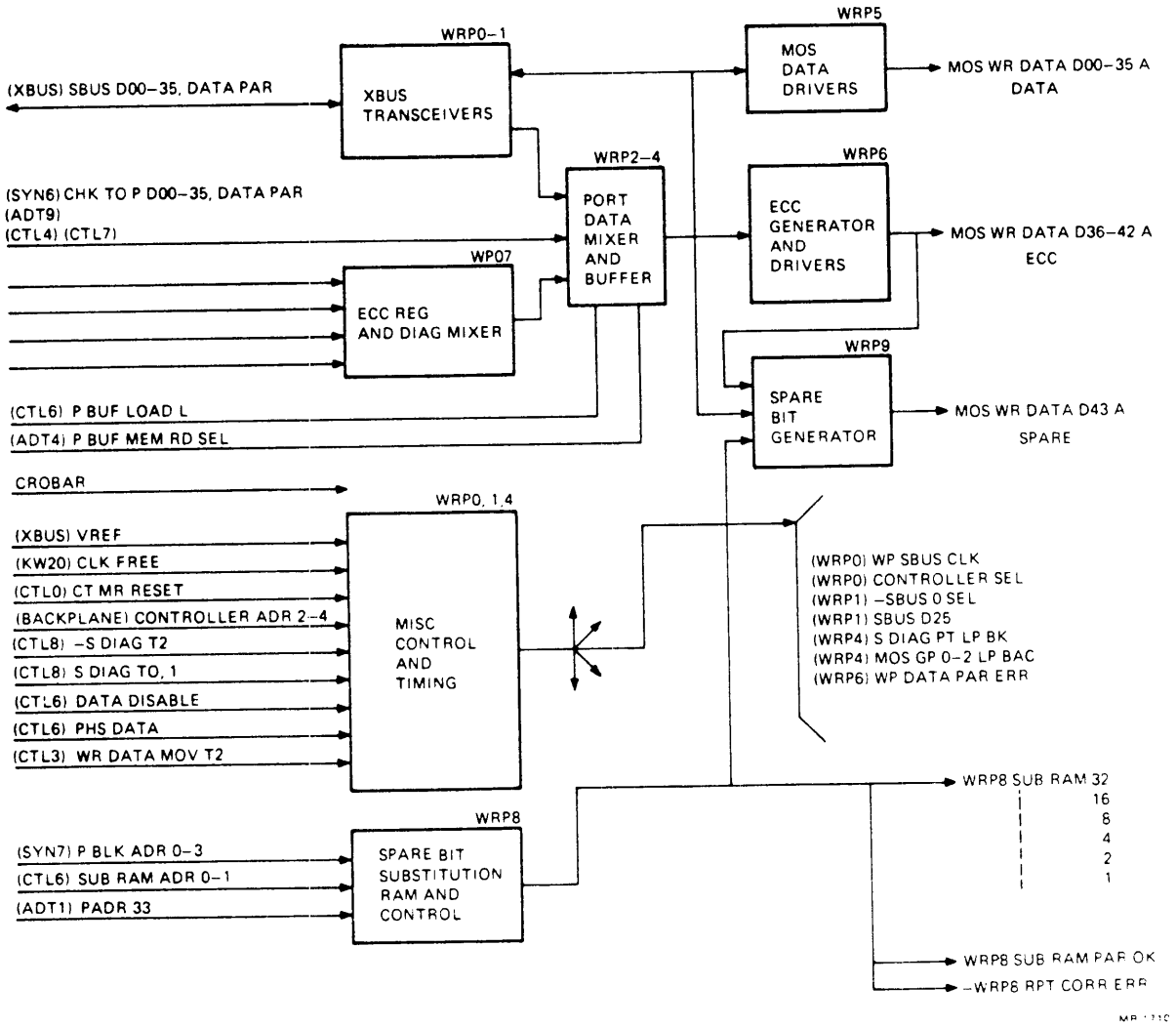


Figure 5-5 M8574 Functional Block Diagram

Table 5-2 M8574 Write Path Output Signal Glossary

Source/Signal Name	Destination Print	Function
(WRP0) SBUS TO P CROBAR	CTL1	A single control signal used during MF20 power sequencing to initialize logic or inhibit activating certain signals.
(WRP0) CONTROLLER SEL	CTL6	A single signal used during a diagnostic function execution to indicate that D00-04 matched the MF20 unit number established by hardwired backplane connections.
(WRP1) SBUS 0 SEL	XBUS	This signal is asserted to indicate that the signal DATA DISABLE D is active, which indicates the SBus data lines are disabled.
(WRP4) S DIAG PT LPBK	CTL6	This signal is set by executing a diagnostic function 4 with D05 = 1. It is used to modify operation of the MF20 to perform the data loopback function.
(WRP4) MOS GP<0-2> LP BAC	SM14	Three signals that are used to enable data loopback through the data multiplexer chips on the M8579 storage modules. They are activated via a diagnostic function 1. MOS GP0 LP BAC - Loopback Group 0 MOS GP1 LP BAC - Loopback Group 1 MOS GP2 LP BAC - Loopback Group 2

Table 5-2 M8574 Write Path Output Signal Glossary (Cont)

Source/Signal Name	Destination Print	Function
(WRP5) MOS WR DATA D00-35 A	(SM00-10) SYN7 SYN8 SYNA ADT3 ADT8 ADT9 CTL1 CTL7 CTL8	This is a 36-bit wide internal data path that transports the information from the write path to the M8579 storage module to be written into MOS. It also distributes control information during execution of diagnostic functions.
(WRP7) MOS WR DATA D36-42 A	SM n	Seven signals that transport the ECC bits from the M8574 to the M8579 modules to be written into MOS.
(WRP9) MOS WR DATA D43 A	SM	This is a single signal that represents the output of the spare bit selector during a write. It is written into the spare bit MOS chip.
(WRP6) WP DATA PAR ERR	ADT7	This signal is activated during a write to indicate bad parity was detected in the M8574 module. It is sent to the ADT module where it sets an error flag. It also jams a DBE into the data transferred to MOS.
(WRP8) ECC SUB RAM 32 16 8 4 2 1	SYN0	These six signals represent the bit position to be substituted when the spare bit is used. They represent the contents of a RAM location and are used to switch the spare bit multiplexers on both the SYN and WRP modules.

Table 5-2 M8574 Write Path Output Signal Glossary (Cont)

Source/Signal Name	Destination Print	Function
(WRP8) -ECC RPT CORR ERR	ADT7	This bit in the spare bit substitution RAM is set by the program. When active it inhibits generating XBUS ERROR when a single-bit error (SBE) is detected.
(WRP8) ECC SUB RAM PAR OK	CTL4	A single signal that is used to flag parity errors when the spare bit substitution RAM is accessed.
(WRP0) (WRP1) SBUS D00-15 SBUS D16-35, DATA PAR	XBUS WRP0 WRP1	A 37-bit data bus that transmits the MF20 data plus parity back to the MBox via the XBus.

5.4.3 M8575 Syndrome (Figure 5-6)

The read half of the data paths is primarily supported by logic on the M8575 module. Major functional areas supported by the SYN module are:

1. Spare bit selection
2. Error checking
3. Error correction
4. Address response RAM
5. Power supply margin register
6. Miscellaneous diagnostic control/status.

During a read operation the storage modules supply 44 bits to the M8575 over the internal data bus, MOS RD DATA D00-43. After substituting the spare bits, if selected, the data and ECC are checked and used to generate a "syndrome." Both the syndrome and ECC are latched into registers for subsequent examination by the program if an error was detected. For single-bit errors the syndrome is decoded and used to correct the bit in error prior to leaving the SYN module. From here 37 bits are sent to the WRP module over the check bus, CHK TO P D00-35, DATA PAR, to be sent back to the MBox via the XBus. Refer to Paragraphs 5.8.1 and 5.8.2 for a more detailed description of spare bit substitution and error checking and correction.

The address response RAM is loaded by the program during initialization to configure the MF20. Its task is to translate physical XBus address to MF20 group and block numbers under control of the software. Refer to Paragraph 5.8.4 for a complete description of the address response RAMs.

An 8-bit register is included to permit the program to control a set of margin signals to the dc power supply. The +12, +5, -2, and -5.2 supplies can be margined above and below their nominal outputs for the purpose of isolating marginal logic circuits that cause intermittent errors.

A diagnostic function mixer is included on the M8575, making it possible for the program to read the state of:

1. The P.S. margin register
2. Any location in the address response RAM
3. The ECC register
4. The syndrome buffer.

Table 5-3 describes the function of all the key signals within the intermodule interface.

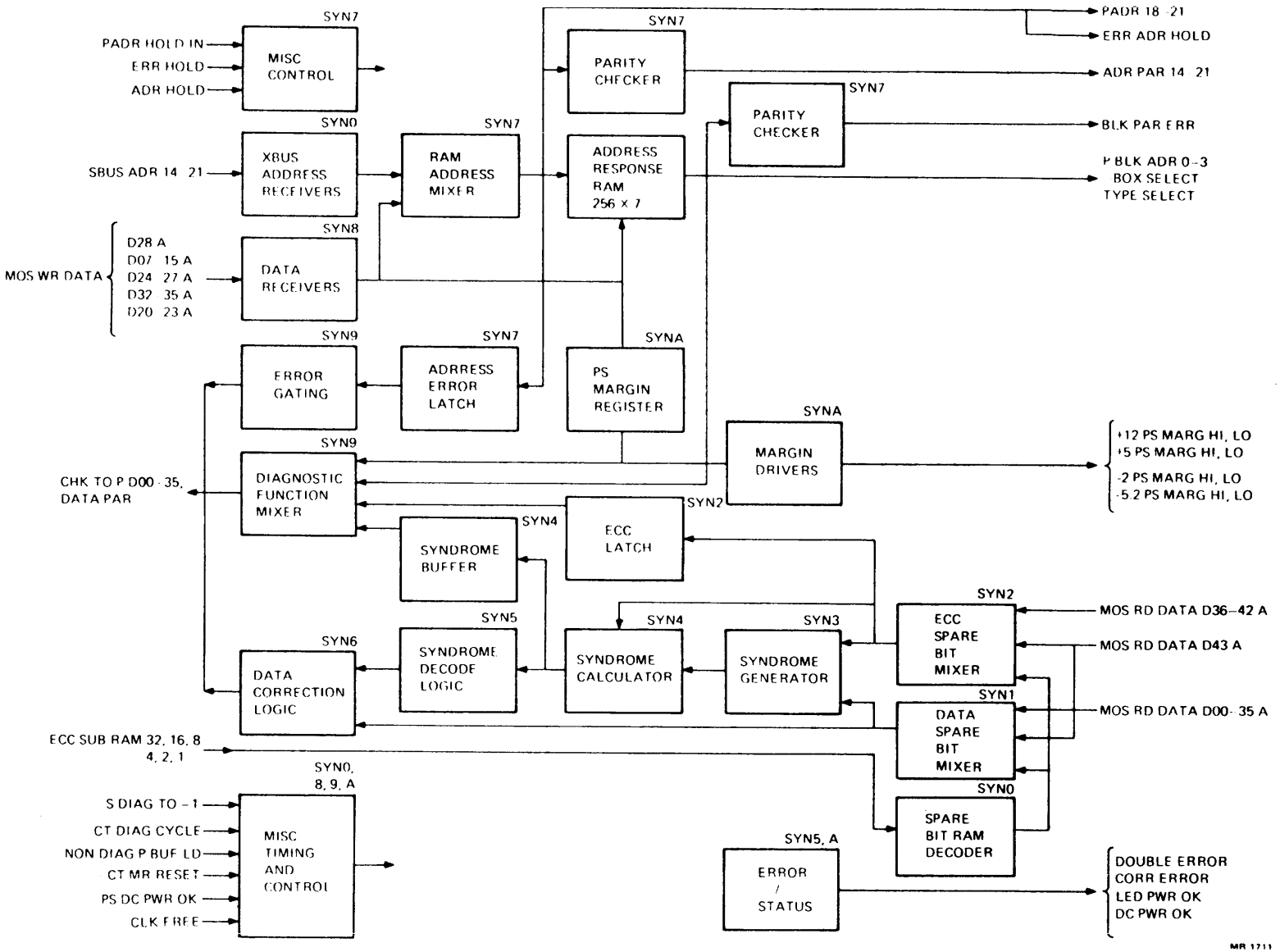


Figure 5-6 M8575 Functional Block Diagram

Table 5-3 M8575 Syndrome Output Signal Glossary

Source/Signal Name	Destination Print	Function
(SYN5) DOUBLE ERROR	ADT7	This signal is asserted when a double-bit error is detected by the M8575 during a read.
(SYN5) CORR ERROR	ADT7	This signal is asserted when a single-bit error is detected to flag it as correctable.
(SYN6) CHK TO P D00-35, DATA PAR	WRP2 WRP3 WRP4	This is a 37-bit internal data path that transports the data plus parity from the SYN to the WRP module during a read. It also conveys status information from other modules during diagnostic functions.
(SYN7) ERR ADR HOLD	ADT1 CTL4	This signal is asserted whenever an MF20 error is detected and used to latch address and word request information.
(SYN7) ADR PAR 14-21	ADT2	This signal is used to indicate overall parity of XBUS ADR bits 14 through 21. It is used on the ADT module for checking all ADR PAR bits.
(SYN7) P ADR 18-21	ADT2	Four lines that transmit the state of XBUS ADR bits 18 through 21 from the SYN to the ADT module.
(SYN7) BLK PAR ERR	ADT7	A single signal that is asserted if a parityas detected when accessing the address response RAM.

Table 5-3 M8575 Syndrome Output Signal Glossary (Cont)

Source/Signal Name	Destination Print	Function
(SYN7) TYPE SELECT	ADT2	A single bit within the address response RAM that is set by the program - used to indicate one of two possible chip sizes, i.e.: 4K or 16K 16K or 64K, etc.
(SYN7) P BLK ADR 0-3	ADT5 CTLA WRP8	Four signals generated by the output of the address response RAM, used to specify the physical group and block number P BLK ADR 0-1 Group no. P BLK ADR 2-3 Block no.
(SYN7) -BOX SELECT	CTL1	A single signal controlled by a bit in each address response RAM location - set by the program to prevent response to particular addresses.
(SYNA) LED POWER OK	Backplane	A single signal that activates a LED indicator on the MF20 backplane to indicate dc power is normal.
(SYNA) DC POWER OK	ADT3	A single signal that is asserted to indicate dc power is normal.
(SYNA) +12 V PS MARG HI +12 V PS MARG LO +5 V PS MARG HI +5 V PS MARG LO -2 V PS MARG HI -2 V PS MARG LO -5.2 V PS MARG HI -5.2 V PS MARG LO	Power Supply	Eight signals that may be asserted under program control to margin the specified dc voltages above (HI) or below (LO) their nominal values.

5.4.4 M8577 Address and Timing (Figure 5-7)

This module supports the following logic functions:

1. MOS array address generation
2. MOS array timing
3. Refresh logic
4. Error handling logic.

During any read or write operation, the M8577 must supply the storage modules with the proper address to specify which location within the 16K MOS chip is to be accessed. A 14-bit address is supplied in two 7-bit slices over the address lines, MOS ADR 0-7A, along with either a row or column address strobe (RAS or CAS). The address is derived from the XBus address lines via the receivers and port address latch.

An alternate source for the MOS address is supplied from the refresh logic along with a RAS strobe. This path is periodically switched in to permit refreshing each row once every 2 ms. See Paragraph 5.8.8 for a complete description of the refresh operation.

The detailed timing sequence of the MOS storage array access is controlled by the contents of a timing RAM. The order and width of the RAS, CAS, and write enable (WE) signals is controlled by the bit patterns loaded into the timing RAM. Refer to Paragraph 5.8.5 for a more detailed description of the RAM.

In addition to the address and timing logic, the M8577 contains most of the error handling logic and the means to assert the XBus error signals to flag the MBox. Table 5-4 describes the function of all the intermodule signals generated or received by the M8577 module.

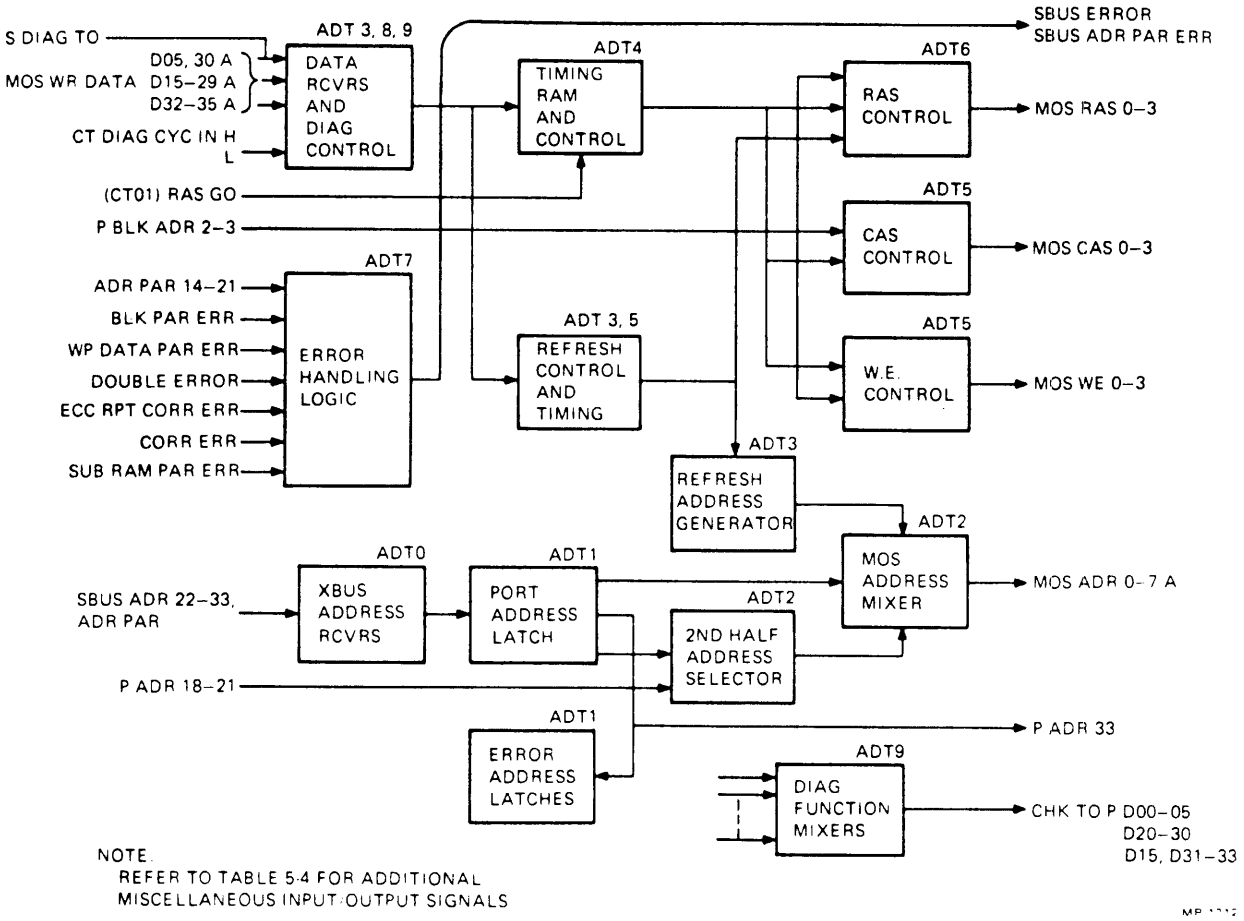


Figure 5-7 M8577 Functional Block Diagram

Table 5-4 M8577 Address and Timing Output Signal Glossary

Source/Signal Name	Destination Print	Function
(ADT0) SBUS TO P DIAG	CTL6	A single control signal that is asserted during execution on a diagnostic function.
(ADT0) SBUS ERROR	XBUS	A single signal asserted by the M8577 to flag MF20 errors - sent back to the MBox via the XBus.
(ADT0) SBUS ADR PAR ERR	XBUS	A single signal asserted by the M8577 to flag bad overall address parity.
(ADT1) P ADR 33	WRP8	A single address signal sent to the WRP module to be used to form the address to the spare bit substitution RAM.
(ADT2) MOS ADR 0-7 A	SM14	An 8-bit internal address bus used to transport the MOS address from the M8577 to the M8579 storage modules.
(ADT3) TIM RAM BUSY CLR A	CTL2	A single control signal generated by the timing RAM - used to indicate termination of a MOS RAM timing sequence.
(ADT3) INC RQ ERR + 1	CTL2	A single signal that is asserted one clock tick after the MF20 detected an incomplete request error.
(ADT4) TIM DATA RDY	CTL2	A signal generated by the timing RAM to indicate when data is ready during a MOS RAM read.

Table 5-4 M8577 Address and Timing Output Signal Glossary (Cont)

Source/Signal Name	Destination Print	Function
(ADT4) P BUF MEM RD SEL	WRP3	A single control signal used to switch the port buffer multiplexer during a read or diagnostic function (second half).
(ADT5) REFRESH NOW REFRESH GO REFRESH GO + 3	CTL1	Three signals generated by the refresh control logic during a refresh cycle to indicate a refresh operation is in progress.
(ADT5) MOS CAS 0-3	SM13	Four control signals used to transmit the column address strobes to the M8579 modules.
(ADT5) MOS WE 0-3	SM11	Four control signals used to transmit the write enable signals to the M8579 module write pulse logic.
(ADT6) CLK FREE 00-02	WRP0 SYN0 CTL0	A source of free-running 30 MHz clocks to the WRP, SYN, and CTL modules.
(ADT6) CLK GATED 00-13	CTL0	A source of 14 gated 30 MHz clock pulses.
(ADT6) MOS RAS 0-3 A	SM12	A source of four row address strobes to the M8579 storage modules.
(ADT7) BOX SELECT EN IN	CTL2	A single control signal that is set by the program via a diagnostic function 1 to enable selecting the MF20.

Table 5-4 M8577 Address and Timing Output Signal Glossary (Cont)

Source/Signal Name	Destination Print	Function
(ADT7) ERR HOLD	SYN7	A single control signal that is asserted by the error handling logic when the MF20 detects an error - used to latch error status.
(ADT9) CHK TO P D00-05, CHK TO P D 15, CHK TO P D 20-33	WRP2 WRP3 WRP4	Twenty-one signals used to transmit error/status information during second half of diagnostic functions.

5.4.5 M8576 Control and Timing (Figure 5-8)

This module contains all the timing and control logic necessary to support the following MF20 functions.

1. Read cycle
2. Write cycle
3. Read-Modify-Write
4. Diagnostic functions

All the XBus control signals enter this module, are stored, and are used to initiate the required timing and control signals to:

1. Initiate the ADT module to generate the required address and timing to access the MOS storage array, and

2. Request, process, and move up to four words from the XBus to the storage module data multiplexer chips if writing or

3. Select, move, process, and transfer up to four words from the storage module data multiplexer chips to the XBus during a read.

The control sequences generated are controlled in part by the contents of the fixed value RAM. This RAM is loaded by the program via diagnostic functions. Its primary functions are to:

1. Control XBus ACKN
2. Control XBus DATA VALID
3. Control the read data mover.

Refer to Paragraph 5.8.6 for a more complete description of the fixed value RAMs.

Besides the cycle timing and data control, the CTL module contains logic to support some of the diagnostic functions. Table 5-5 describes most of the key signals in the intermodule interface.

5.4.6 Summary

Paragraphs 5.4.1 through 5.4.5 presented a functional overview of the logic partitioning within the MF20 logic modules. The block diagrams supplemented with key signal glossaries provide a complete description of the internal operation of the MF20 from a functional viewpoint. Detailed logic descriptions are contained in Paragraph 5.8.

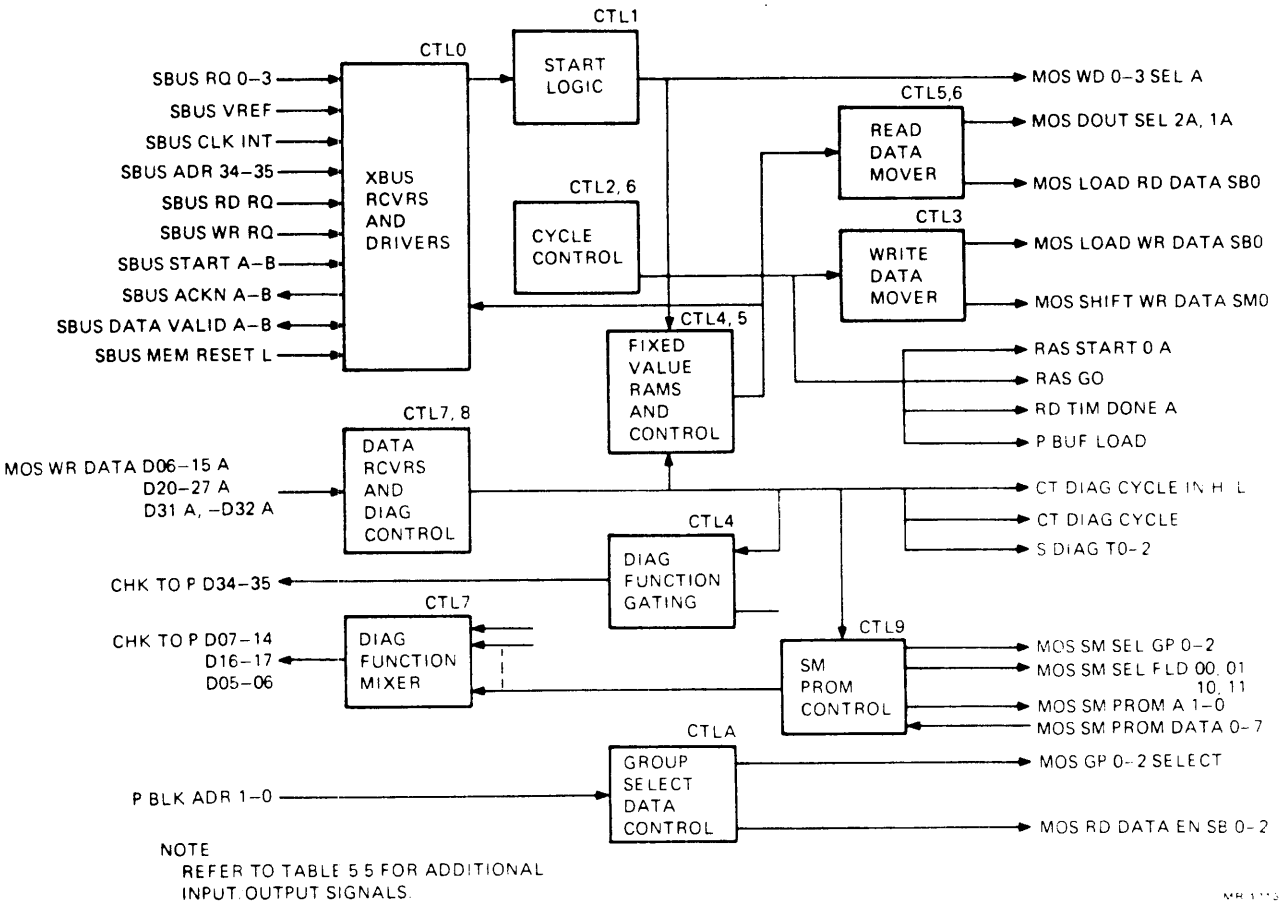


Figure 5-8 M8576 Functional Block Diagram

Table 5-5 M8576 Control and Timing Output Signal Glossary

Source/Signal Name	Destination Print	Function
(CTL0) CT MR RESET	SYN9 WRP4 ADT7	A single master reset signal derived from the XBUS MR RESET.
(CTL0) VREF/#19/		XBus voltage reference to the MF20 backplane.
(CTL0) SBUS ACKN A/B	XBUS	Two XBus acknowledge signals (phase A or B) generated by the MF20 to acknowledge receipt of a request from the MBox.
(CTL0) DATA VALID A/B	XBUS	Two XBus signals asserted by the MF20 during a read to strobe data back to the MBox.
(CTL1) RPW 2ND HALF	ADT7	A single control signal asserted during the second half of a read-pause-write cycle.
(CTL1) P RQ PAR	ADT2	A single signal used to indicate the parity detected on the RD RQ, WR RQ, and RQ 0-3 XBus control lines.
(CTL1) P RD RQ	ADT4	A single control signal that is asserted during a read or read-pause-write cycle.
(CTL1) MOS WD 0-3 SEL A	SM11	Four signals used to specify which words in a quadword group are to be written into MOS.
(CTL1) ADR HOLD	SYN7 ADT1	A single control signal used to latch the address received from the XBus into the MF20.

Table 5-5 M8576 Control and Timing Output Signal Glossary (Cont)

Source/Signal Name	Destination Print	Function
(CTL1) RAS GO	ADT4	A single control signal from the CTL to the ADT module that initiates a MOS timing RAM sequence.
(CTL1) RAS START 0 A	ADT7	A single control signal that is asserted whenever a MOS memory cycle is started.
(CTL2) SINGLE STEP	ADT6	A single control signal that is asserted to indicate the MF20 is in single-step diagnostic mode.
(CTL2) A PHS COM FREE B PHS COM FREE PHS COMING	ADT6 ADT7	Three basic timing signals derived from the XBus synchronizing clock.
(CTL2) RD TIM DONE	ADT5	A single control signal used to indicate that the read is done during a timing RAM sequence.
(CTL2) P ADR HOLD IN	SYN7	A single signal used to time latching of the error address registers.
(CTL2) P ADR HOLD	ADT7	A single control signal used to time error handling logic and latching port address.
(CTL3) WR DATA MOV T2	WRP4	A single write timing signal generated during a write operation.
(CTL3) MOS LOAD WR DATA SM0 MOS SHIFT WR DATA SM0	SM14	Two control signals generated by the write mover logic to control loading and shifting of data into the data multiplexer or the M8579 modules.

Table 5-5 M8576 Control and Timing Output Signal Glossary (Cont)

Source/Signal Name	Destination Print	Function
(CTL4) SUB RAM PAR ERR	ADT7	A single control signal that is asserted when a parity error is detected while accessing the spare bit substitution RAM.
(CTL4) CHK TO P D34-35	WRP4	Two status signals that are gated onto the check bus during a diagnostic function to read ADR 34-35.
(CTL5) MOS DOUT SEL 2, 1 A	SM14	Two control signals used to switch the data multiplexer chips on the M8579 modules during a read.
(CTL6) DATA DISABLE	WRP4	A single control signal used to control the gating signals that enable the XBus data drivers.
(CTL6) PHS DATA	WRP4	A timing signal used to control gating of data onto the XBus.
(CTL6) P BUF LOAD	WRP3	A single control signal that enables loading the port data buffer register.
(CTL6) NON DIAG P BUF 0	ADT7 SYN9	A single control signal used to time the error handling logic during normal read/write operations.
(CTL6) SUB RAM ADR 0-1	WRP8	Two signals that permit modifying the low-order bits of the spare bit substitution RAM address.
(CTL6) MOS LOAD RD DATA SB0	SM14	A single control signal that activates the data multiplexer chips on the M8579 module during a read.

Table 5-5 M8576 Control and Timing Output Signal Glossary (Cont)

Source/Signal Name	Destination Print	Function
(CTL7) CHK TO P D05-06 D07-14 D16-17	 WRP2 WRP3 WRP4	Twelve signals that permit gating back error/status information during a diagnostic function.
(CTL8) CT DIAG CYCLE IN H CT DIAG CYCLE IN L CT DIAG CYCLE S DIAG T0-2	 SYN8 WRP4	Six timing and control signals used to activate logic to activate logic to execute diagnostic functions.
(CTL9) MOS SM SEL FLD 00 MOS SM SEL FLD 01 MOS SM SEL FLD 10 MOS SM SEL FLD 11 MOS SM SEL GP 0-2 MOS SM PROM A 1,0	SM14	Nine signals used to select a group, field, and location when accessing the personality PROM.
(CTLA) MOS GP0-2 SELECT	SM14	Three signals used to select one of three groups of storage modules during a read or write.
(CTLA) MOS RD DATA EN SB0-2	SM14	Three signals used to control the data multiplexer chips on the M8579 storage modules during a read.

5.5 THE XBUS

All communication between the MBox and the MF20 MOS Memory Subsystem occurs over a system bus called the XBus. This paragraph describes the physical and functional characteristics of the XBus including the communication dialogues that occur between the MBox and the MF20 during typical memory operations. Since the XBus signals form the most convenient test points for checking internal MF20 operations, it is important to understand the operation of the XBus prior to looking at more detailed MF20 logic operations.

5.5.1 Physical Description (Figure 5-9)

The communication link between the MBox and up to two MF20s consists of the following hardware.

1. Two memory translator modules that plug into slots 7 and 8 in the MBox backplane

2. A set of twisted-pair wires that connect the active bus signals from slots 7 and 8 to slots 2 and 3 in the MBox backplane

3. A 5 ft XBus cable assembly that consists of a quad M8572 cable board on the MBox end (plugs into slot 2), a set of four multiconductor coaxial cables (20 X 2 positions per cable), and four Berg connectors at the MF20 end

4. A set of four Berg connectors mounted directly on the MF20 backplane

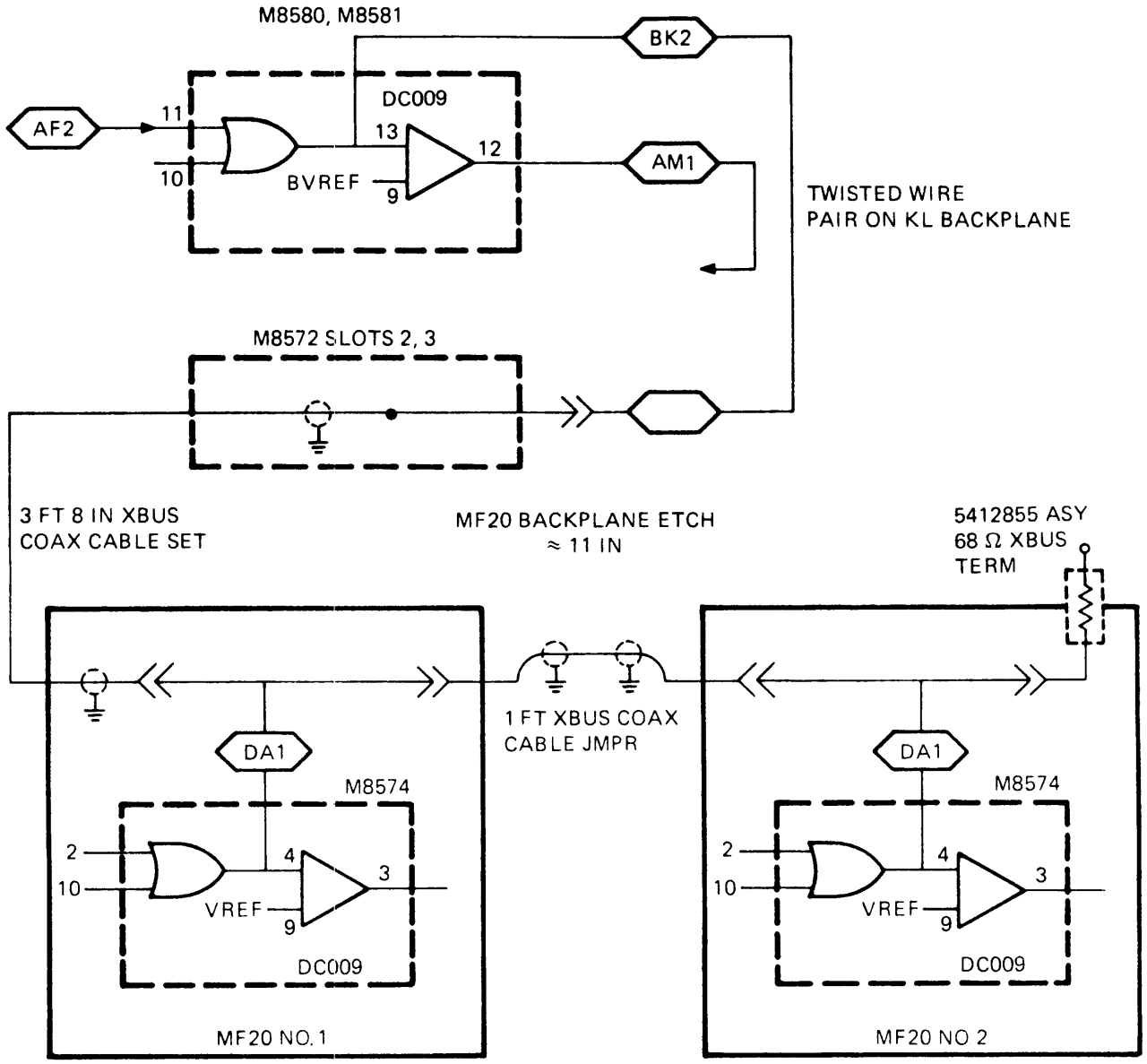
5. A set of four more multiconductor coaxial cables (1 ft long) with Berg connectors at both ends to connect the XBus to a second MF20

6. A set of four 68 ohm terminator assemblies that plug into the output connectors on the last MF20 backplane

A second XBus can be plugged into slot 3 (XBus 0) to connect additional MF20s to the MBox. In this case the M8572 cable assembly contains 13 ft coaxial cable assemblies.

5.5.2 Functional Description

Table 5-6 lists all the signals in the XBus interface with brief functional descriptions. Paragraphs 5.5.2.1 through 5.5.2.4 describe how these signals interact to perform the four basic MF20 operations. Table 5-7 lists the XBus signal test points.



- MF20 SIGNALS (WRP0)
  M8574
  PIN 4 SBUS D04 H
  PIN 3 SBUS TO P D04 H
  PIN 2 P BUF D04 H
  PIN 10 DATA DISABLE A H
- KL10 CPU
  M8580
  PIN 11 MB 04 H
  PIN 10 DATA TO MEM DISABLE B H
  PIN 13 SBUS 0 D04
  PIN 12 MEM DATA IN 04 H

MR-1714

Figure 5-9 XBus Overview (Data Bit 04)

Table 5-6 XBus Signal Summary

Signal Name	Print	Function
ADR 14-35 14-21 22-33 34-35	 SYN0 ADT0 CTL0	These are 22 lines that transport the physical address from the MBox to the MF20 MOS Memory Subsystem. Used to uniquely identify a specific block within a specific group in a specific MF20.
ADR PAR	ADT0	A single line that is controlled by the MBox whenever a memory reference is started. It is used to establish overall odd parity for the following lines. ADR 14-35 RD RQ WR RQ RQ 0-3 The MF20 checks the parity and asserts the XBus signal, ADR PAR ERR if an error is detected.
RQ 0-3	CTL0	Four lines that are asserted by the MBox to specify which words in a quadword group are to be accessed. RQ0 Request word 0 RQ1 Request word 1 RQ2 Request word 2 RQ3 Request word 3 If more than one word is requested, ADR 34-35 will specify the order in which the words will be processed.
RD RQ	CTL0	The MBox asserts this line to specify that the request is a read request.
WR RQ	CTL0	The MBox asserts this line to specify that the request is a write request. Note During a read-modify-write both RD RQ and WR RQ are asserted by the MBox.
START A-B	CTL0	The MBox asserts one of these two signals to signal the MF20 to begin executing the read/write function specified by the state of the RD RQ, WR RQ, and RQ 0-3 lines.

Table 5-6 XBus Signal Summary (Cont)

Signal Name	Print	Function
ACKN A-B	CTL0	The selected MF20 responds to the MBOX START A-B signal by asserting the corresponding ACKN A or ACKN B signal to acknowledge the request. If no memory unit asserts ACKN within a specified time limit, the MBox will abort the request and set a nonexistent memory (NXM) flag.
D00-35	WRP0, 1	These 36 lines are used to transport a single KL10 data word either from the MF20 to the MBox during a read, or from the MBox to the MF20 during a write.
DATA PAR	WRP1	This line is used to establish odd parity on the data bus, D00-35, during any read or write operation. Read - The MF20 generates the parity, and the MBox checks it. Write - The MBox generates the parity, and the MF20 checks it.
DATA VALID A-B	CTL0	Two lines asserted by either the MF20 (read) or the MBox (write portion of read-modify-write) to signal when data is valid on the data bus
DIAG	ADT0	The MBox asserts this line to signal the MF20 selected by D00-04 that it is to perform the diagnostic function specified by D31-35.
ERROR	ADT0	This line is asserted by the selected MF20 to signal the MBox that one of the following types of errors has been detected. RD ERR WR PAR ERR ADR ERR TIM RAM ERR SUB RAM ERR BLK RAM ERR INC ERR The MBox can execute a subsequent diagnostic function 1 or 2 to determine which of the above errors has occurred.

Table 5-6 XBus Signal Summary (Cont)

Signal Name	Print	Function
ADR PAR ERR	ADT0	This line is asserted by the MF20 to signal that bad address parity was detected during a read-write operation.
MEM RESET	CTL0	The MBox can assert this signal to place the MF20 in a known initial state.
CROBAR	WRP0	This signal clears the MF20 control logic to its initial state during power-up or power-down.
CLK INT	CTL0	XBus clock used to synchronize all operations within the MF20 to the MBox timing
VREF	CTL0	A single line that establishes a common voltage reference for the XBus transceivers at both ends of the XBus.

Table 5-7 XBus Signal Test Points

Signal Name		Backplane	Berg Connector
ADR 14	(H)	6CE1	J1-8
ADR 15	(H)	6CF1	J1-12
ADR 16	(H)	6CJ1	J1-16
ADR 17	(H)	6CK2	J1-20
ADR 18	(H)	6CL2	J1-24
ADR 19	(H)	6CN1	J1-28
ADR 20	(H)	6CR1	J1-32
ADR 21	(H)	6CT2	J1-36
ADR 22	(H)	7CU2	J1-38
ADR 23	(H)	7CV2	J1-40
ADR 24	(H)	7DE1	J2-8
ADR 25	(H)	7DF1	J2-12
ADR 26	(H)	7DJ1	J2-16
ADR 27	(H)	7DK2	J2-20
ADR 28	(H)	7DL2	J2-24
ADR 29	(H)	7DN1	J2-28
ADR 30	(H)	7DR1	J2-32
ADR 31	(H)	7DS1	J2-36
ADR 32	(H)	7DU2	J2-38
ADR 33	(H)	7DV2	J2-40
ADR 34	(H)	5EE1	J3-8
ADR 35	(H)	5EF1	J4-34
ADR PAR	(H)	7EJ1	J3-16
RD RQ	(H)	5EL2	J3-24
WR RQ	(H)	5EK2	J3-20
RQ 0	(H)	5FF1	J4-12
RQ 1	(H)	5FE1	J4-8
RQ 2	(H)	5EV2	J3-38
RQ 3	(H)	5EU2	J3-36
START A	(H)	5FL2	J4-24
START B	(H)	5FJ2	J4-28
ACKN A	(L)	5FV2	J4-40
ACKN B	(L)	5FU2	J4-38
DATA VALID A	(L)	5FR1	J4-32
DATA VALID B	(L)	5FT2	J4-36
D00	(L)	4CA1	J1-2
D01	(L)	4CD1	J1-6
D02	(L)	4CH2	J1-14
D03	(L)	4CL1	J1-22
D04	(L)	4DA1	J2-2
D05	(L)	4DD1	J2-6
D06	(L)	4DH2	J2-14
D07	(L)	4DL1	J2-22
D08	(L)	4DP2	J2-30
D09	(L)	4ED1	J3-6
D10	(L)	4EH2	J3-14
D11	(L)	4EL1	J3-22
D12	(L)	4EP2	J3-30
D13	(L)	4FA1	J4-2
D14	(L)	4FD1	J4-6

Table 5-7 XBus Signal Test Points (Cont)

Signal Name		Backplane	Berg Connector
D15	(L)	4FH2	J4-14
D16	(L)	4FL1	J4-22
D17	(L)	4FP2	J4-30
D18	(L)	4CC1	J1-4
D19	(L)	4CE2	J1-10
D20	(L)	4CK1	J1-18
D21	(L)	4CM2	J1-26
D22	(L)	4DC1	J2-4
D23	(L)	4DE2	J2-10
D24	(L)	4DK1	J2-18
D25	(L)	4DM2	J2-26
D26	(L)	4DS2	J2-34
D27	(L)	4EE2	J3-10
D28	(L)	4EK1	J3-18
D29	(L)	4EM2	J3-26
D30	(L)	4EC1	J3-4
D31	(L)	4FC1	J4-4
D32	(L)	4FE2	J4-10
D33	(L)	4FK1	J4-18
D34	(L)	4FM2	J4-26
D35	(L)	4FS2	J4-34
DATA PAR	(L)	4EN1	J3-28
DIAG	(L)	7FK2	J4-20
MEM RESET	(L)	5EA1	J3-2
CROBAR	(L)	4ER1	J3-34
CLK INT	(H)	5ET2	J3-32
VREF		5FA1	J3-40
ADR PAR ERR	(L)	7CP2	J1-30
ERROR	(L)	7FJ1	J4-16

Note: (H) = +2.19 V; (L) = +1.6 V
First digit in backplane column is module slot number in backplane.

5.5.2.1 Read Dialogue - When the MBox prepares to retrieve information from the MF20 it initiates the following XBus sequence.

1. It asserts the address lines, XBUS ADR 14-35, to specify the location of the first word it wants to retrieve.

2. It asserts the XBUS RD RQ line to instruct the MF20 to initiate a read cycle.

3. It asserts up to four request lines, XBUS RQ 0-3, to specify which words within a quadword group it wishes to retrieve.

4. It calculates odd parity based on the state of all address and request lines and either asserts or negates the XBUS ADR PAR signal to reflect this parity.

5. It then asserts either XBUS START A or XBUS START B to command the MF20 to execute the read request.

6. The selected MF20 uses XBUS START A/B to store the address and request information and initiates a MOS memory timing sequence to access and retrieve the information from the MOS storage array.

7. The MF20 responds immediately to the MBox START by asserting the signal XBUS ACKN A/B and keeps it asserted for N number of XBUS INT CLK periods, where N is the number of words requested. The MBox drops START and negates all other control lines after the last ACKN period.

8. As the words from the MOS array become available the MF20 gates the information back on the XBUS D00-35, DATA PAR lines along with the control signal XBUS DATA VALID A,B.

9. The MBox uses DATA VALID to strobe the data lines into the appropriate memory buffer register where it is subsequently checked for parity and distributed to its ultimate destination (EBox, MBox, channel buffer, cache, etc.)

10. After all the words have been processed the MF20 returns to its initial state (if a refresh request is not pending) and is ready to accept a new command.

5.5.2.2 Write Dialogue - When the MBox prepares to store information in the MF20, it initiates the following XBus sequence.

1. It asserts the address lines, XBUS ADR 14-35 to specify the location of the first word to be written.

2. It asserts the XBUS WR RQ line to instruct the MF20 to initiate a write cycle.

3. It asserts the XBUS RQ 0-3 lines to specify which words within the referenced quadword group are to be written.

4. It asserts XBUS ADR PAR to generate odd parity based on the state of all address and control lines (ADR 14-35, WR RQ, RQ 0-3).

5. It asserts the XBUS D00-35, DATA PARITY lines to reflect the first word to be written.

6. It asserts XBUS START A/B to command the MF20 to execute the write request.

7. Upon receiving START the MF20 responds by asserting XBUS ACKN and maintains it asserted for N number of XBUS CLK INT periods, where N is equal to the number of request lines asserted. It also gates the XBUS D00-35, DATA PAR lines to store the first word to be written in the MF20's port data buffer.

8. During subsequent XBUS INT CLK periods the MBox gates the remaining words to be written onto the XBUS DATA lines where they are taken and stored by the MF20.

9. After receiving all the words (up to 4 max.) the MF20 negates XBUS ACKN and initiates a MOS write timing sequence to store the words in the MOS storage array.

10. After writing the MOS the MF20 returns to its initial state and is ready to accept another command.

5.5.2.3 **Read-Pause-Write Dialogue** - When the MBox prepares to retrieve a single word from the MF20, modify it in the EBox, and then store the result back in the MF20, it initiates the following XBus sequence.

1. It asserts the XBUS ADR 14-35 lines to specify the location of the word to be modified.

2. It asserts both the XBUS RD RQ and XBUS WR RQ lines to instruct the MF20 to execute both a read and a write cycle.

3. It asserts one of the XBUS RQ 0-3 lines to specify which word in the quadword group is to be modified.

4. It asserts or negates XBUS ADR PAR to generate odd parity for all the address and control lines.

5. It asserts XBUS START A/B to command the MF20 to execute the read-modify-write operation.

6. Upon receipt of START the selected MF20 responds by asserting XBUS ACKN A/B and initiates a MOS storage array timing sequence to retrieve the word from MOS.

7. When the word is available, it is gated out of the MF20 back to the MBox on XBUS DATA D00-35, DATA PAR along with the control signal XBUS DATA VALID A/B.

8. The MBox uses DATA VALID to gate the data lines into the appropriate memory buffer register (MBR) where it is sent to the EBox for manipulation.

9. The MF20 pauses at this point and waits for the MBox to respond.

10. When the data is ready to be stored back into memory, it is loaded into the appropriate MBR and gated out onto the XBUS D00-35, DATA PAR lines.

11. At this time the MBox asserts XBUS DATA VALID A/B to signal the MF20 to continue.

12. When the MF20 receives DATA VALID it gates in the data and initiates a MOS write sequence to store the word in the MOS storage array.

13. At this time the cycle ends with the MF20 in its initial state ready to execute a new command.

5.5.2.4 Diagnostic Cycle Dialogue - When the EBox decodes an XBus diagnostic function instruction (BLKO PI, E) it initiates the following XBus sequence.

1. It retrieves the contents of the location specified by E and instructs the MBox to gate this 36-bit word onto XBUS D00-35.

2. Then the MBox asserts the XBUS DIAG signal to initiate a diagnostic cycle.

3. Upon receipt of the DIAG signal, the selected MF20 (specified by XBUS D00-04) stores the 36 bits of data in its port data buffer where the bits are used to activate the control logic signals as specified by the particular function to be performed (XBUS D31-35).

4. After executing the specified function, the MF20's port data buffer is loaded with the pertinent status/error information and gated out onto the XBUS D00-35 lines.

5. At the MBox end four XBUS CLK INT periods after asserting XBUS DIAG, the MBox strobes in the data and stores it at the address E+1.

6. At this point the diagnostic cycle ends and the MF20 returns to its initial state ready to accept a new command.

5.5.3 Summary

To communicate with the MF20, the MBox uses a bus called the XBus that consists of a unique set of logic signals that define the communication protocol. A specific bus protocol must be followed to execute any one of four possible memory operations.

5.6 DIAGNOSTIC FUNCTIONS

When power is first applied to an MF20, the RAMs within its control logic contain undefined information. In effect, the MF20 itself is undefined until the software loads the control RAMs to configure the memory subsystem. A set of eleven diagnostic functions provides the mechanism to permit software to load the RAMs. Besides providing configuration capability, these diagnostic functions permit:

1. System software to read error and status flags implemented in the logic

2. System software to track and log faults in the MOS storage array

3. Diagnostic software to test and verify detailed logic operations within the MF20 control section.

This paragraph describes how a typical diagnostic function operates from both a hardware and a software viewpoint.

5.6.1 Programming

The KL10 architecture defines a single instruction, the BLKO PI, E, as the memory diagnostic function instruction. Prior to executing the instruction the programmer sets up two consecutive memory locations (or ACs) to contain the following information.

[E] A 36-bit control word to be sent out to the MF20

[E+1] A 36-bit error/status word sent back from the MF20

As will be shown shortly, the contents of [E] must select a specific MF20 unit and also specify the function to be performed.

Tables 4-1 to 4-11 list all eleven of the diagnostic functions and includes the following information.

1. The encoding scheme for both the [E], control information, and the [E+1], status information

2. Logic print references to link the programming information to the hardware logic implementation

3. A functional description of each bit or group of bits in [E] or [E+1]

The following example illustrates how the diagnostic software enables data loopback through group 1 on MF20 no. 1.

```
HRLI 1, 220060
HRRI 1, 1
SETZM 2
SBDIAG 1
```

Subsequent paragraphs will include more references to diagnostic function programming as each functional logic area is discussed.

5.6.2 Block Diagram (Figure 5-10)

When the EBox executes a diagnostic function, it instructs the MBox to place the [E] on the XBus data lines and, after sufficient skew delay, assert the XBUS DIAG signal. At the MF20 end the diagnostic function control logic in each MF20 decodes the state of XBUS D00-04 (MF20 no.) and XBUS D31-35 (DIAG FCN no.) and activates the required control signal sequence in the selected MF20. During the first half of the XBus diagnostic cycle the 36 bits of control information are present at the inputs of the port data buffer (WRP2-4) after passing through the receivers (WRP0-1) and the port data mixer (WRP2-4). The first occurrence of (CTL6) P BUF LOAD allows the next CLK FREE to load the port buffer with the XBus data [E]. Once loaded the active P BUF D bits activate one or more signals within the MF20 control facility. For the programming example given in Paragraph 5.6.1, P BUF D13 will activate (WRP4) MOS GP1 LP BAC when clocked into the shift register E4.

During the second half of the diagnostic cycle, (WRP3) RD BUF SEL is asserted to switch the port data mixer to steer the selected error/status/control information to the inputs of the port data buffer. Then the second occurrence of P BUF LOAD permits the next CLK FREE to load the port data buffer. These 36 bits are sent back to the MBox via the SBus drivers and the XBus data bus. Approximately 12 clock ticks after negating XBUS DIAG, the MBox will clock in this data (point X) and proceed to store it in [E+1] as directed by the BLKO PI instruction.

The logic represented by the three boxes is as follows.

1. Diagnostic function control logic
2. MF20 control facility
3. Input diagnostic mixers

This logic is spread across two or more of the control modules. It is more fully described in subsequent paragraphs on the functional areas of logic affected by diagnostic functions. To locate and analyze this logic and the signals represented by A and B in the block diagram, refer to the prints called out in Tables 4-1 to 4-11 for the function being examined.

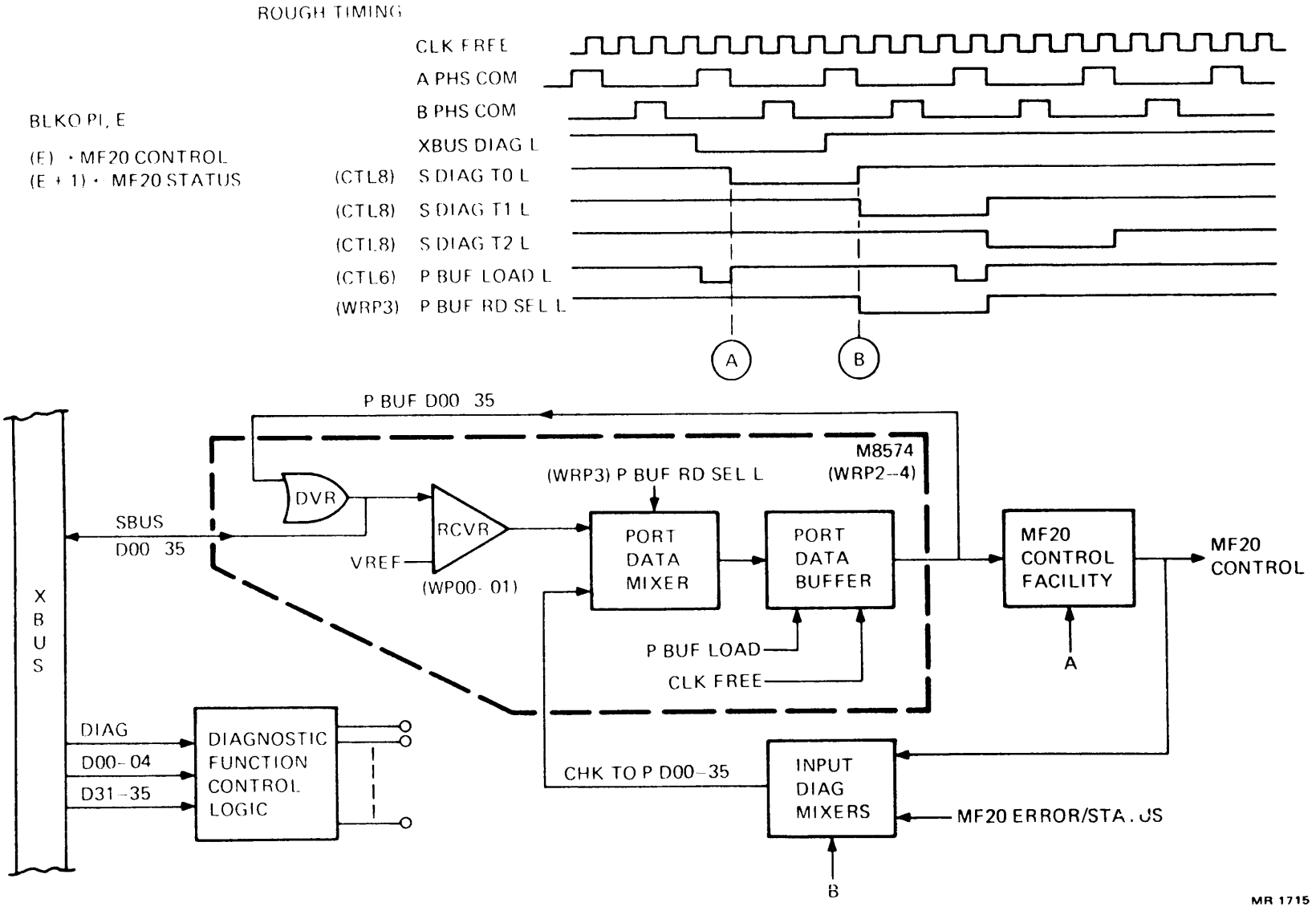


Figure 5-10 MF20 Diagnostic Function Overview

5.6.3 Summary

Eleven diagnostic functions are supported by the MF20 logic. They are activated when system software executes a BLKO PI, E instruction to access the MF20. Besides providing the system software with a method of configuring the MOS memory and monitoring status, they permit diagnostic software to check the MF20 control logic using nonfunctional testing.

5.7 BASIC MF20 OPERATIONS

During normal operation the MF20 performs its system-related tasks by executing four basic operations: read, write, read-pause-write, and diagnostic functions. These operations are cyclic by nature and result in the generation of well defined sequences of logic events within the MF20. This paragraph describes each operation in detail using simplified block diagrams and timing diagrams to explain the information flow and key signal generation.

5.7.1 Read Cycle (Figures 5-11, 5-12, 5-13)

This paragraph describes the major sequence of events that must occur to properly execute a quadword read. It focuses on functional logic areas and intermodule signal generation rather than a complete logic description. This description may be used in conjunction with the detailed timing diagrams in the print sets to analyze the logic operations to any desired level.

When the MBox wishes to initiate a quadword read from the MF20, it asserts the following XBus signals.

1. SBUS ADR 14-35 with the physical bus address to specify the location of the quadword within the address space

2. SBUS RD RQ to specify that the MF20 should execute a read request

3. SBUS RQ 0, 1, 2, and 3 to specify that all four words within the quadword group are to be transferred

4. SBUS ADR PAR to generate overall odd parity on ADR 14-35, RD RQ, and RQ 0-3.

Each MF20 on the XBus uses the signals SBUS ADR 14-21 to access the address response RAM on its M8575 module (SYN7). If the address is mapped in the address response RAM, the signal SYN7 - BOX SELECT will be negated indicating the box is selected. Only the MF20 with this signal negated can "see" the subsequent XBUS START.

After allowing sufficient time to ensure that all MF20s have decoded ADR 14-35, the MBox asserts SBUS START A/B (phase A or phase B) to initiate the MF20 control logic on the M8576 module (CTL1).

If the selected MF20 is not currently busy (refresh cycle or read/write cycle) the SBUS START A/B signal sets a flip-flop and activates the signal CTL1 RAS START 0 A. This signal in turn activates CTL2 P ADR HOLD which turns on CTL1 ADR HOLD, which latches the following information.

1. SYN7 P ADR 14-21 <= SBUS TO P ADR 14-21 which will be used to access the address response RAM and provide ADR 18-21 to form the MOS address

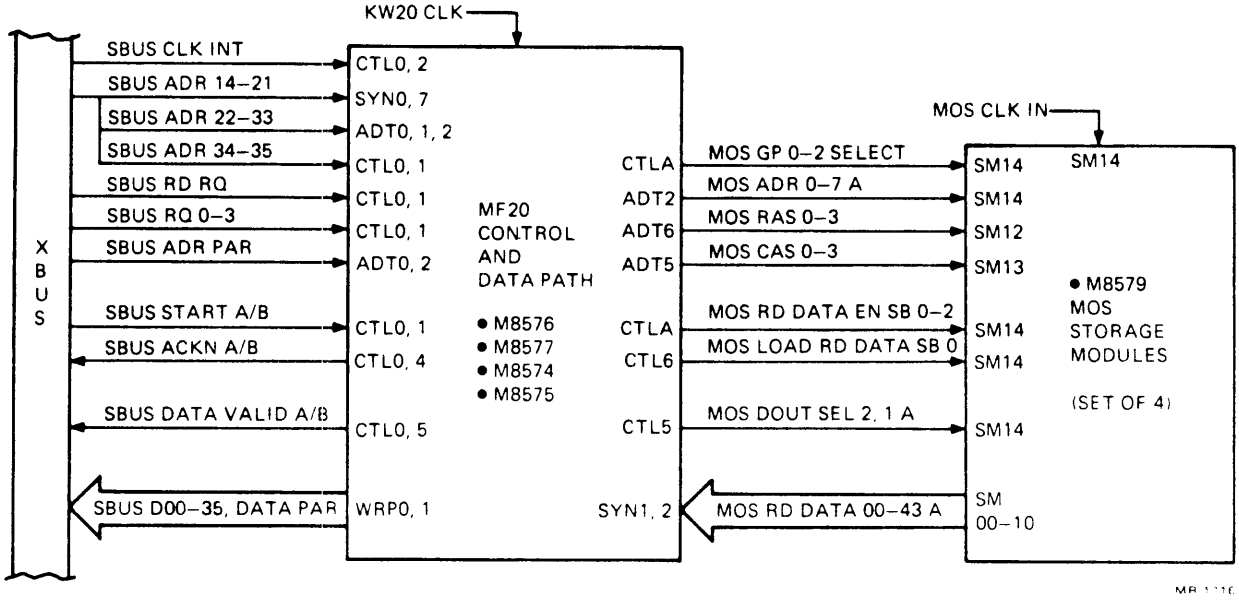


Figure 5-11 Read Cycle Overview

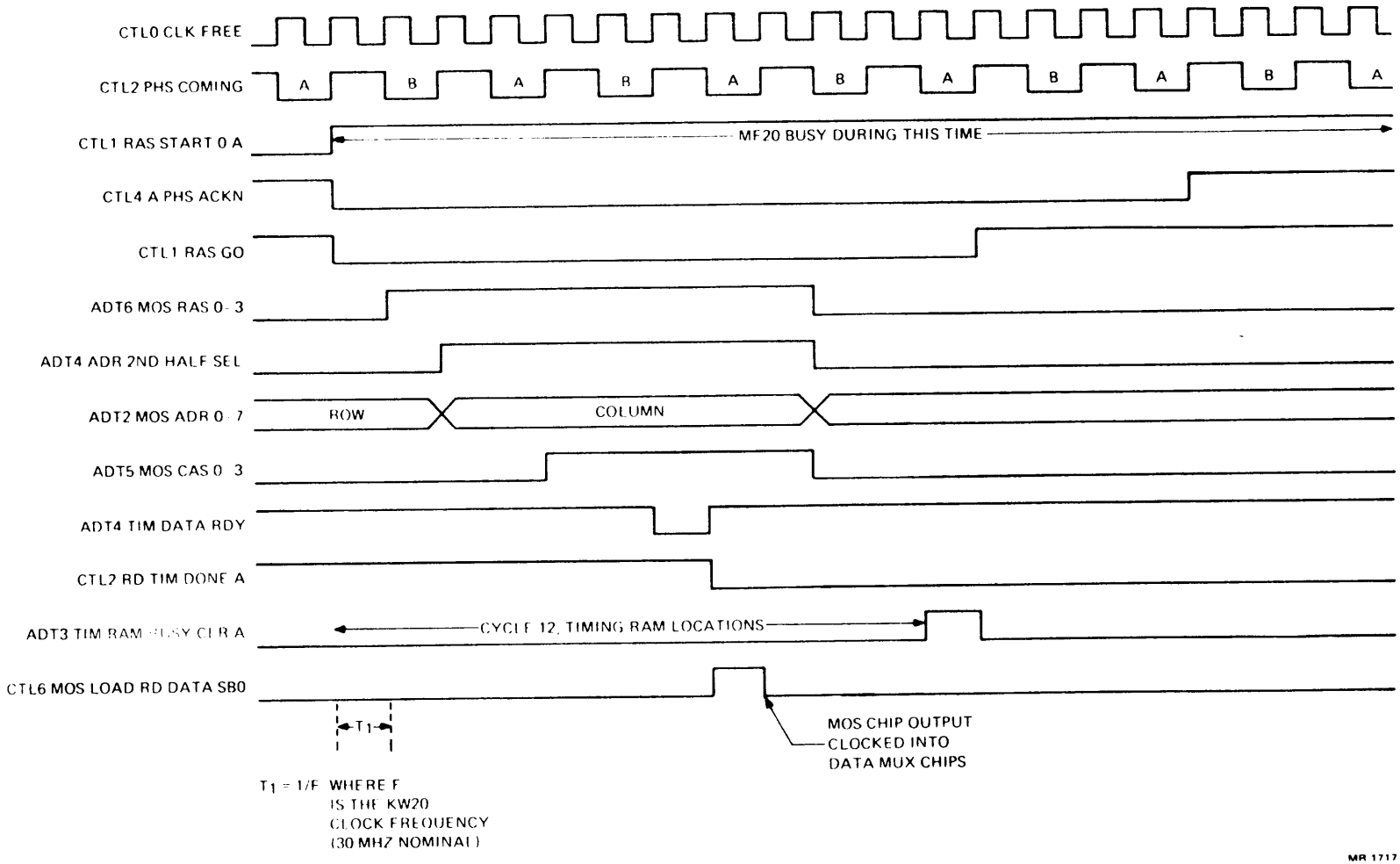


Figure 5-12 Read Cycle Phase 1 Rough Timing Diagram

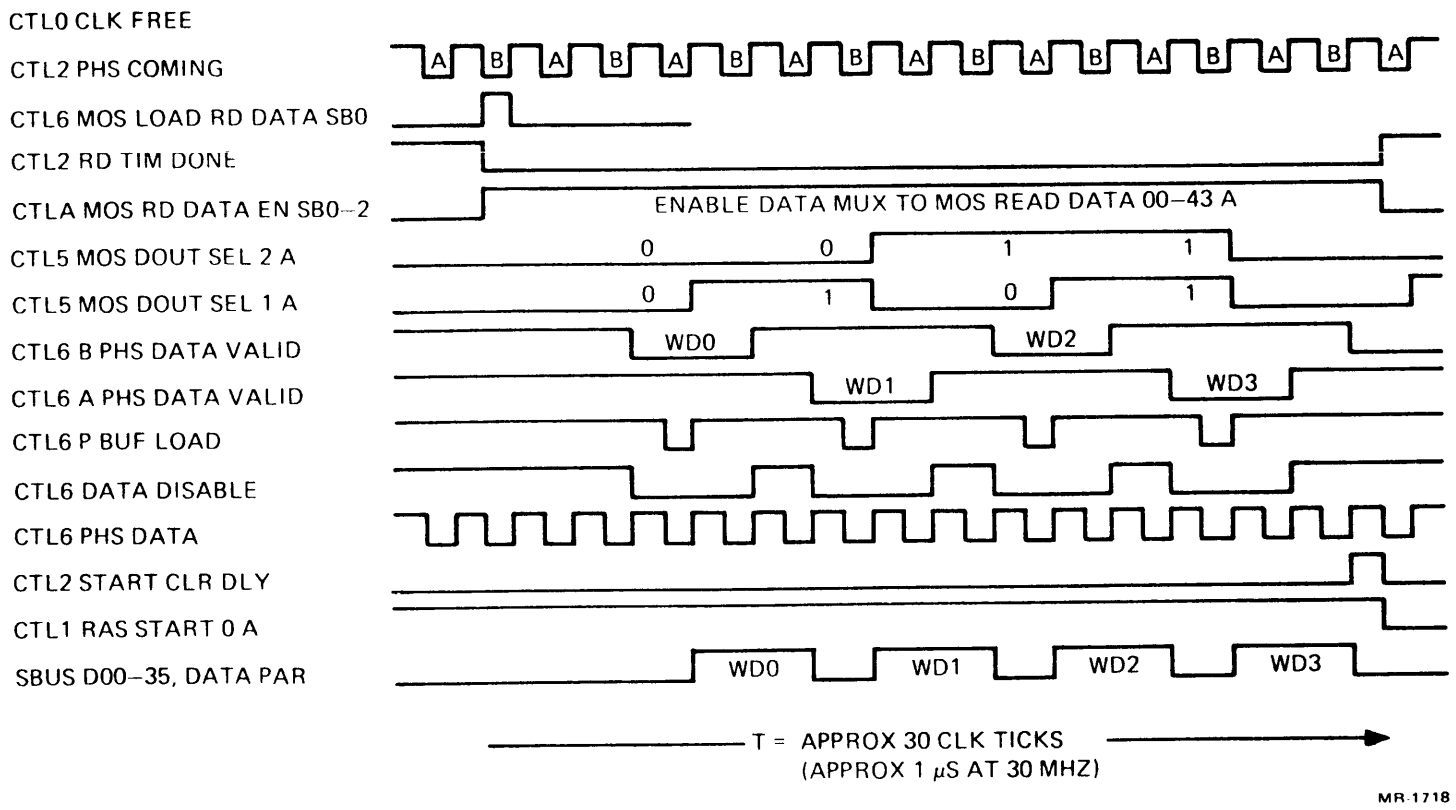


Figure 5-13 Read Cycle Phase 2 Rough Timing Diagram

2. ADT1 P ADR 22-33 <= SBUS TO P ADR 22-33 which will be the prime source for forming the MOS address

3. CTL1 P ADR 34-35 <= SBUS TO P ADR 34-35 which will be used to control the read data mover

4. CTL1 P RD RQ <= SBUS TO P RD RQ which stores the type of request

5. CTL1 P RQ 0-3 <= SBUS TO P RQ 0-3 which stores information concerning which words within the quadword group are requested

RAS START 0 A also enables the ACKN section of the fixed value RAM to generate CTL4 P ACKN EN for the number of clock periods equal to the number of active RQ signals. This would be four for a quadword request. P ACKN EN activates either CTL4 A PHS ACKN or CTL4 B PHS ACKN, depending on whether the cycle was initiated on an A or B phase clock. A/B PHS ACKN in turn asserts CTL0 SBUS ACKN A or B to acknowledge the request back at the MBox.

Now the actual MF20 read cycle can be initiated. It consists of the following two separate and distinct phases.

1. The MOS RAMs are addressed and strobed to read out four words.

2. The words within the 4-word group must be moved from the storage module data multiplexer chips to the XBus.

RAS START 0 initiates the first phase by asserting CTL1 RAS GO which enables the MOS timing RAM logic on ADT4. Once enabled, the timing RAM generates the following sequence of control signals.

```
ADT4 TIM RAM RAS
ADT4 ADR 2ND HALF
ADT4 TIM RAM CAS
ADT4 TIM RAM WE (not significant during a read)
ADT4 TIM DATA RDY
ADT4 TIM RAM BUSY CLR (stops the timing RAM sequencer)
```

At this point the selected group of M8579 MOS storage modules is activated as follows.

1. One of three group select signals, CTLA MOS GPn SELECT is asserted (n = group no.) as specified by the address response RAM outputs SYN7 BLK ADR 1-0.

2. The first seven bits of the MOS address are asserted on ADT2 MOS ADR 0-7A as a function of P ADR 27-33. This is the row address.

3. The timing RAM signal TIM RAM RAS activates one of four

signals, MOS RAS 0-3 as a function of the block address bits P BLK ADR 2-3 from the address response RAM. This combination of row addresses and row address strobe selects one 128-cell row within all the MOS chips in the selected group and block.

4. Next, the timing RAM signal ADT4 2ND HALF SEL switches the MOS address so that ADT2 MOS ADR 0-7A become a function of P ADR 20-26. This is the column address.

5. Now the signal ADT4 TIM RAM CAS asserts one of four column address strobes ADT5 MOS CAS 0-3 as specified by the block address, P BLK ADR 2-3, again a function of the address response RAM output. This combination selects a single cell in each MOS chip within the selected row of 128.

6. Now the signal ADT4 TIM DATA RDY is asserted which generates CTL6 LOAD RD DATA SB0 to allow the next clock to load the MOS data into the data multiplexer chips on the M8579 storage modules (SM00-10). Also the signal CTL2 RD TIM DONE activates one of the three data multiplexer enable signals, CTL1 MOS RD DATA EN SB0-2 as a function of the group select signals.

7. Finally, the signal ADT TIM RAM BUSY CLR is asserted to terminate the MOS timing RAM sequencer.

This completes the first phase of the read with four words being stored in the data multiplexer chips on the M8579 modules. Now, control shifts to the second phase of the read which includes the following major steps.

1. Select one of four words in the multiplexer chips.

2. Move the word through the SYN module to:

   a. Substitute the spare bit if necessary

   b. Check for errors

   c. Correct single-bit errors

   d. Strip off the ECC bits and transmit 36 bits of data plus parity to the WRP module.

3. Load the checked data into the port buffer to be gated onto the XBus data lines and assert XBUS DATA VALID.

4. Repeat steps 1-3 to move all the words requested by RQ 0-3.

To accomplish the task of moving the data from the MOS storage modules to the XBus, the second part of the fixed value RAMs is

activated. The following signals are sequenced to control the process.

CTL5 MOS DOUT SEL 2, 1 A - selects which word out of a possible four is to be moved

CTL5 SET DATA VALID - used to generate an XBUS DATA VALID for each word transferred

The specific sequence is a function of the contents of the RAM. For a quadword transfer starting at a quadword boundary (ADR 34-35 = 00) the following sequence occurs.

1. CTL5 MOS DOUT SEL 2A and CTL5 MOS DOUT 1 A are both negated to select WD0 from the data multiplexer chips on the M8579 (SM00).

2. The 44-bit word selected appears on the MOS RD DATA 00-43A bus and is transmitted to the inputs of the spare bit mixers (SYN1, 2).

3. After selecting the spare bit, if applicable, the data and ECC are checked for errors and used to generate the syndrome (SYN3, 4).

4. If the syndrome indicates a single bit error, it is used to activate the proper correction signal on SYN5.

5. After checking, the data is corrected if necessary (SYN6) and then sent to the WRP module over SYN6 CHK TO P D00-35, DATA PAR.

6. Finally, the checked and corrected data is loaded into the port buffer on the WRP2, 3, 4 prints under control of WRP3 P BUF RD SEL and CTL6 P BUF LOAD.

7. From here the contents of the port buffer are gated out to the XBus data lines (WRP0, 1). The timing signals CTL6 DATA DISABLE and CTL6 PHS DATA on WRP4 control the gating of the XBus drivers.

8. Finally, for each word transferred, the signal CTL0 SBUS DATA VALID A or B is asserted.

9. The fixed value RAM changes CTL5 MOS DOUT SEL 2, 1 A to select the next word to be moved and steps 2-8 are repeated. This continues until all words requested have been transferred.

The entire read sequence ends when CTL2 START CLR DLY is asserted to clear CTL1 RAS START 0 A. At this point the MF20 is ready to execute a new command.

For a more detailed description of these two phases refer to

Paragraphs 5.8.5 and 5.8.6. Figures 5-12 and 5-13 show rough timing diagrams of the 2-phase sequence.

5.7.2 Write Cycle (Figures 5-14 and 5-15)
This paragraph describes the sequence of events that occurs during execution of an MF20 write cycle. It assumes that four words (a quadword) are to be transferred from the MBox to the MF20 and written into the MOS storage array. Like the previous description of the read operation, it will focus on functional logic areas and the generation of key intermodule signals.

When the MBox initiates a quadword write it asserts the following XBus signals.

1. SBUS ADR 14-35 to specify the physical bus address to be used to access the storage array

2. SBUS WR RQ to specify that the MF20 is to execute a write request

3. SBUS RQ 0-3 to specify that four words are to be written

4. SBUS ADR PAR to generate overall odd parity on ADR 14-35, WR RQ, and RQ 0-3

5. SBUS D00-35, DATA PAR which contains the first data word to be written

All MF20s on the XBus use the signals SBUS ADR 14-21 to access a location within the address response RAM on the M8575 module (SYN7). If the address is mapped in the RAM, the signal SYN7 - BOX SELECT will be negated to permit selecting the MF20. Only the MF20 with SYN7 - BOX SELECT negated can use the subsequent START signal to initiate the write.

After allowing sufficient deskew time and time for all address response RAMs to decode ADR 14-21, the MBox asserts SBUS START A/B (phase A or B) to initiate the MF20 cycle control logic on the M8576 module (CTL1).

If the selected MF20 is not busy (refresh or read/write cycle), the SBUS START A/B signal activates the signal CTL1 RAS START 0 A. This signal in turn asserts CTL2 P ADR HOLD which turns on CTL1 ADR HOLD to latch the following bus information.

1. SYN7 P ADR 14-21 <= SBUS TO P ADR 14-21 which is used to access the address response RAM and provide ADR 18-21 to form the MOS address

2. ADT1 P ADR 22-33 <= SBUS TO P ADR 22-33 which will be the prime source for forming the MOS address

3. CTL1 P ADR 34-35 <= SBUS P ADR 34-35 which will be used to control shifting of the data words into the data

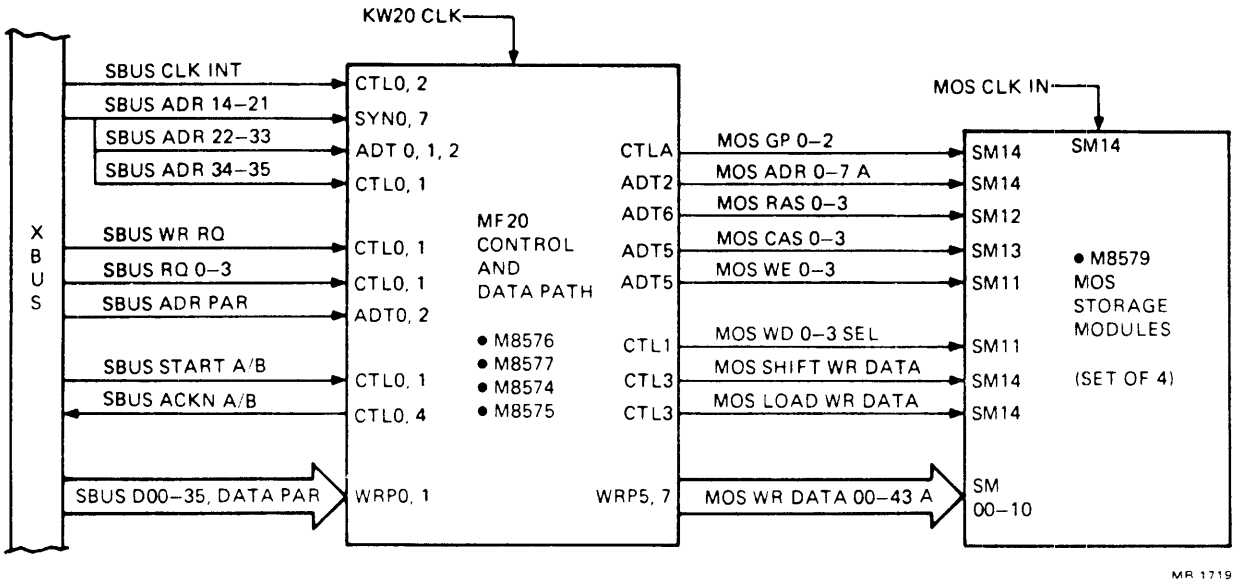


Figure 5-14 Write Cycle Overview

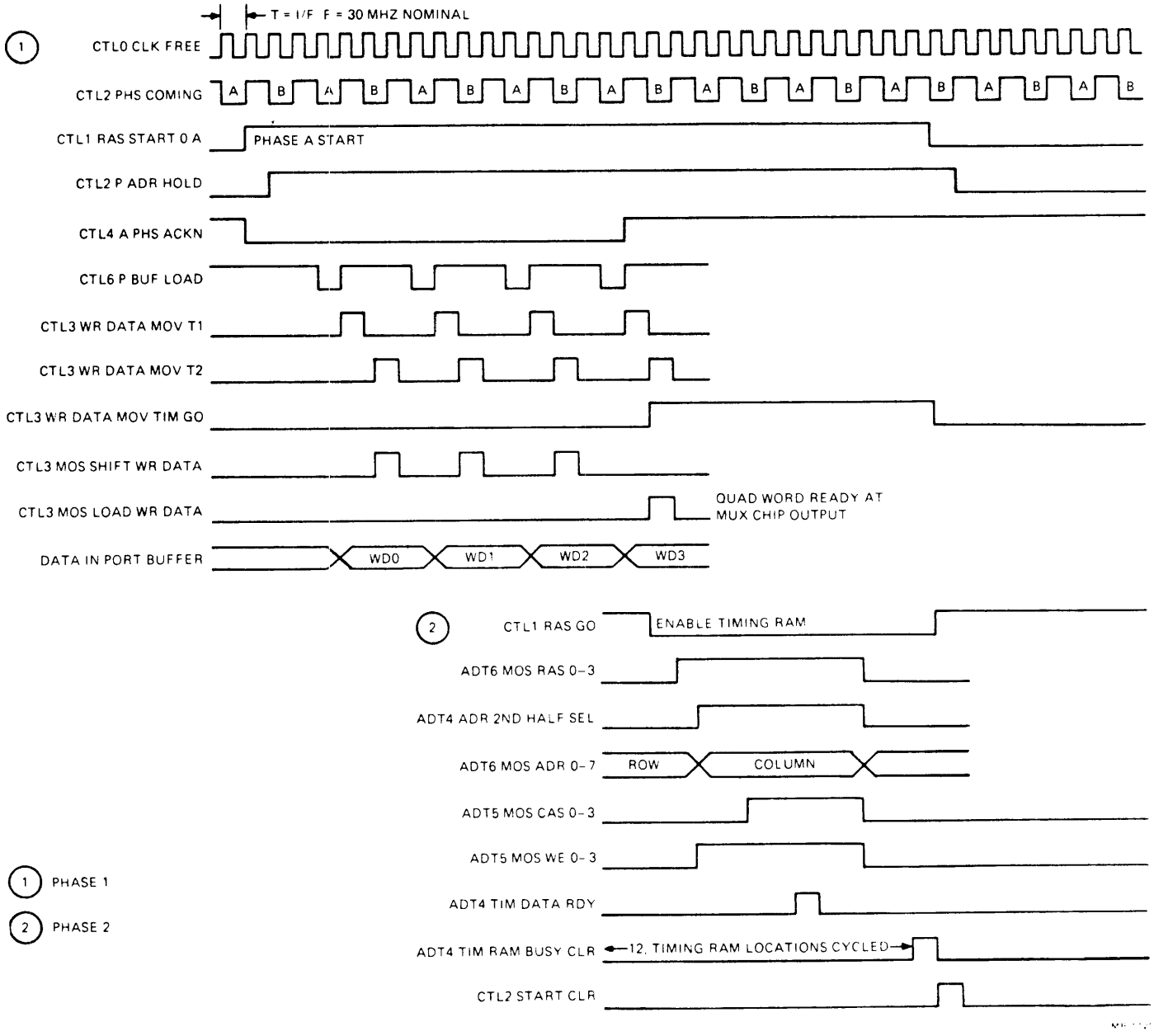


Figure 5-15 Write Cycle Rough Timing Diagram

multiplexer chips - write data mover logic

4. CTL1 P WR RQ <= SBUS TO P WR RQ used to store the write request

5. CTL1 P RQ 0-3 <= SBUS TO P RQ 0-3 which stores information on which words within the quadword are to be written

RAS START 0 A also enables the ACKN section of the fixed value RAM to generate CTL4 P ACKN EN for a number of clock periods equal to the number of active RQ lines. P ACKN EN activates either CTL4 A PHS ACKN or CTL4 B PHS ACKN, depending on whether the cycle was initiated on an A or B phase clock. A/B PHS ACKN in turn asserts CTL0 SBUS ACKN A/B to acknowledge the request back at the MBox.

Now the write cycle can be initiated. Like the read cycle, it consists of two separate and distinct phases.

1. Four words must be taken off the XBus data lines and moved through the WRP module out to the data multiplexer chips on the storage modules.

2. The MOS RAMs are addressed and strobed to write the contents of the multiplexer chips into the MOS array.

When CTL1 RAS START 0 is asserted, CTL4 A PHS ACKN turns on for four clock periods (quadword write). This signal activates four CTL6 P BUF LOAD pulses in succession, coincident with an A PHS COMING timing pulse since the example assumed the cycle started on a phase A clock. Each P BUF LOAD pulse permits clocking the XBus data into the port buffer. It is the responsibility of the MBox to gate out each word to be written in synchronization with the MF20 clocks.

The data in the buffer (WRP2, 3, and 4) is used to calculate a 7-bit ECC check character (WRP6) which is appended to the 36 bits of data. Next, the spare bit is generated if applicable and 44 bits of information are transmitted over WRP5, 7 MOS WR DATA 00-43A. This process is repeated for each word clocked into the port buffer.

To move the data between the WRP and SM modules, a 2-cycle timer is triggered to generate the two signals, CTL3 WR DATA MOV T1 and CTL3 WR DATA MOV T2. The number of times this logic is cycled is a function of the starting state of ADR 34-35. The detailed sequence is as follows.

1. CTL6 P BUS LOAD permits clocking a word into the data buffer.

2. This word is processed by the WRP module and sent to the storage modules over MOS WR DATA 00-43A.

3. The write data move timer is cycled, generating CTL6 WR DATA MOV T2 which activates CTL6 MOS SHIFT WR DATA.

4. CTL6 MOS SHIFT WR DATA permits the next clock to shift MOS WR DATA D00-43A into the M8579 multiplexer chips.

5. Steps 1 through 4 are repeated to load and move words 0, 1, and 2 into the multiplexer chips.

6. Now the final sequence of the write data mover generates CTL3 MOS LOAD WR DATA which loads the last word, WD3, into the multiplexer chips and activates CTL3 WR DATA MOV TIM GO.

7. At this point CTL1 RAS GO is asserted to begin the second phase, writing into the addressed MOS locations.

8. RAS GO initiates the timing RAM sequence to:

   a. Assert a row address strobe, ADT MOS RAS 0-3 to strobe the selected block of MOS chips

   b. Assert ADT4 ADR 2ND HALF SEL to switch the column address into ADT6 MOS ADR 0-7

   c. Assert a column address strobe, AT05 MOS CAS 0-3 along with a write enable signal AT05 MOS WE 0-3 to write the data into the selected MOS array locations.

9. ADT4 TIM RAM BUSY CLR is generated last to terminate the sequence by asserting CTL2 START CLR.

The entire write operation terminates by clearing CTL1 RAS START 0 A, CTL2 P ADR HOLD, CTL3 WR DATA MOV TIM GO, and CTL1 RAS GO. At this point the MF20 is ready to accept a new command.

5.7.3 Read-Pause-Write Cycle (Figures 5-16 and 5-17)

This paragraph describes a read-pause-write cycle as a combination of a single-word read followed by a single-word write. The MBox initiates this type of cycle when executing those instructions that involve retrieving a word from MOS, modifying it to produce a new word (usually), and storing the word back in the same MOS location. The following description discusses only the ways in which this cycle differs from the normal read or write cycle previously discussed. Detailed description of the operation is supplied as an annotated timing diagram (Figure 5-17).

When the MBox executes a read-pause-write cycle, it asserts both XBUS RD RQ and XBUS WR RQ. This condition is stored in the selected MF20 and used to modify the normal read cycle sequence to prevent generation of CTL2 START CLR unless the MF20 control is in the second half of a read-pause-write (RPW) cycle. This state is sensed by the signal CTL1 RPW 2ND HALF being negated. Without CTL2 START CLR, the signal CTL1 RAS START 0 A remains asserted which keeps the MF20 busy to additional cycle requests.

After the CPU has modified the data, it asserts XBUS DATA VALID and gates the results back to the MF20. Data valid asserts CTL0 DATA VALID B IN which reactivates the MF20. CTL6 P BUF LOAD allows clocking the result into the port buffer and on the next CT02 PHS A COMING, the write data move timer is triggered. The contents of the port buffer are written into MOS and the signal CTL2 START CLR is asserted during the second half, which permits resetting CTL1 RAS START 0 A. This causes the MF20 to become ready to accept a new command to read or write.

5.7.4 Diagnostic Function Cycle (Figures 5-18 and 5-19)

This paragraph describes the MF20 sequence generated during execution of a diagnostic function. Since diagnostic function 2 is the function used to access information in the storage modules as well as in the control and data path modules, it will be used as an example to explain the sequence.

When the MBox executes a diagnostic function, it gates the contents of [E] onto the XBus data lines (less parity) and asserts the control signal XBUS DIAG. The MF20 selected by XBUS D00-05 will assert the signal WRP0 CONTROLLER SEL which in turn asserts CTL6 P BUF LOAD. P BUF LOAD enables the next clock to load the XBus data into the port buffer (WRP2, 3, 4). This is the contents of [E], the control information.

ADT0 SBUS TO P DIAG along with CONTROLL. . SEL activates a 3-bit shift register to generate the timing chain.

CTL8 S DIAG T0
CTL8 S DIAG T1
CTL8 S DIAG T2.

T0 and T1 enable the additional control signals CTL8 S DIAG CYCLE

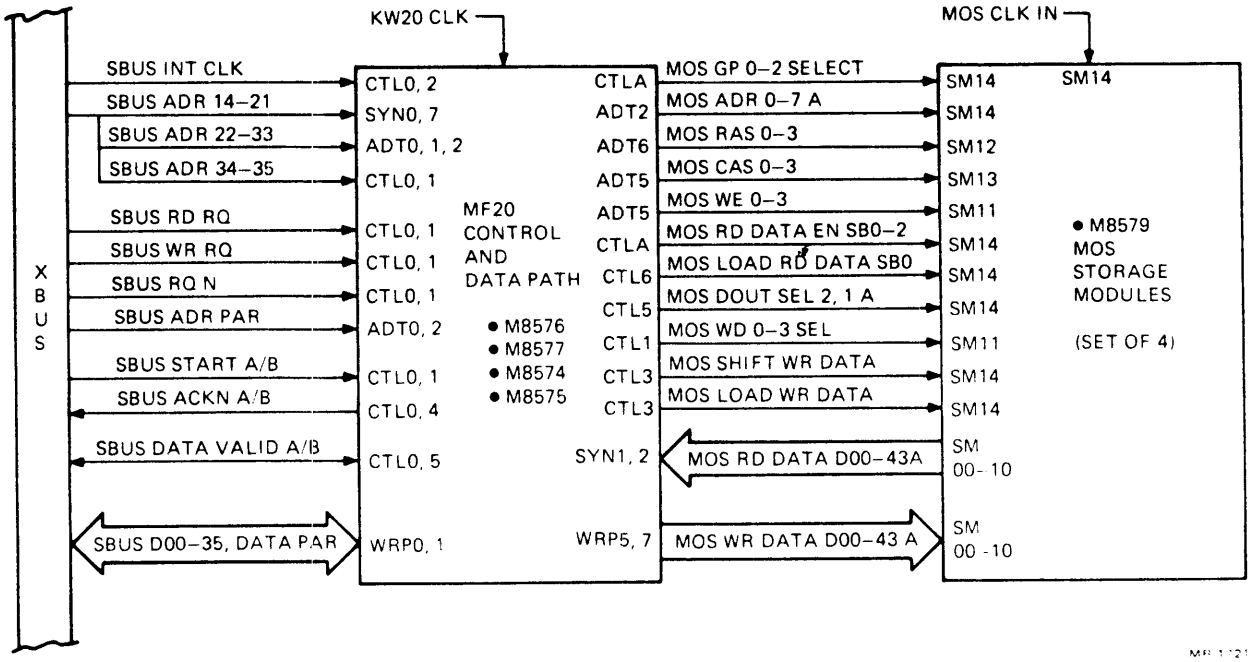


Figure 5-16 Read-Pause-Write Cycle Overview

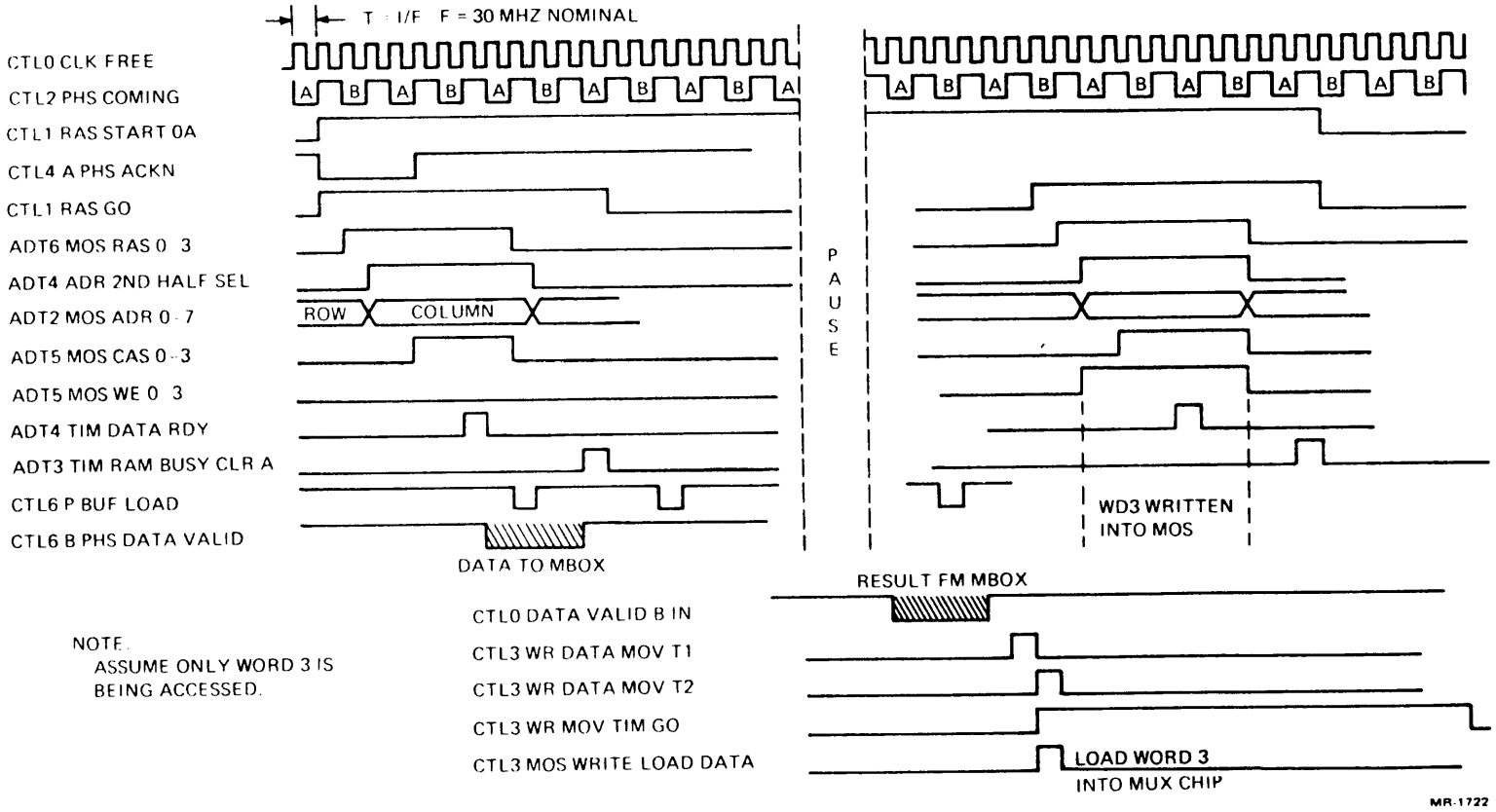


Figure 5-17 Read-Pause-Write Cycle Rough Timing Diagram

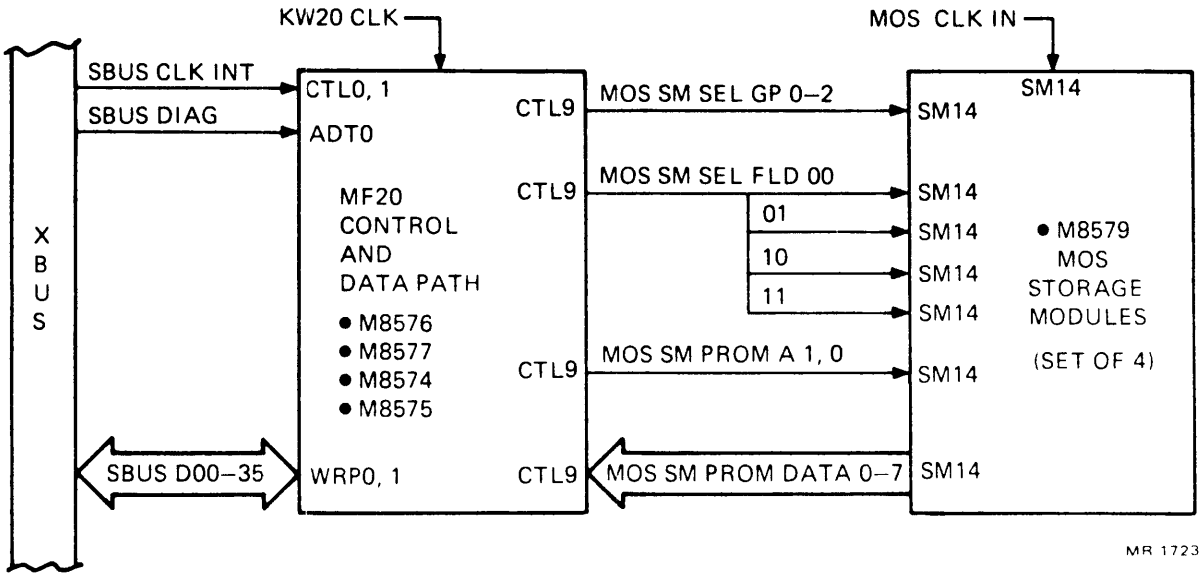


Figure 5-18 Diagnostic Function Cycle Overview

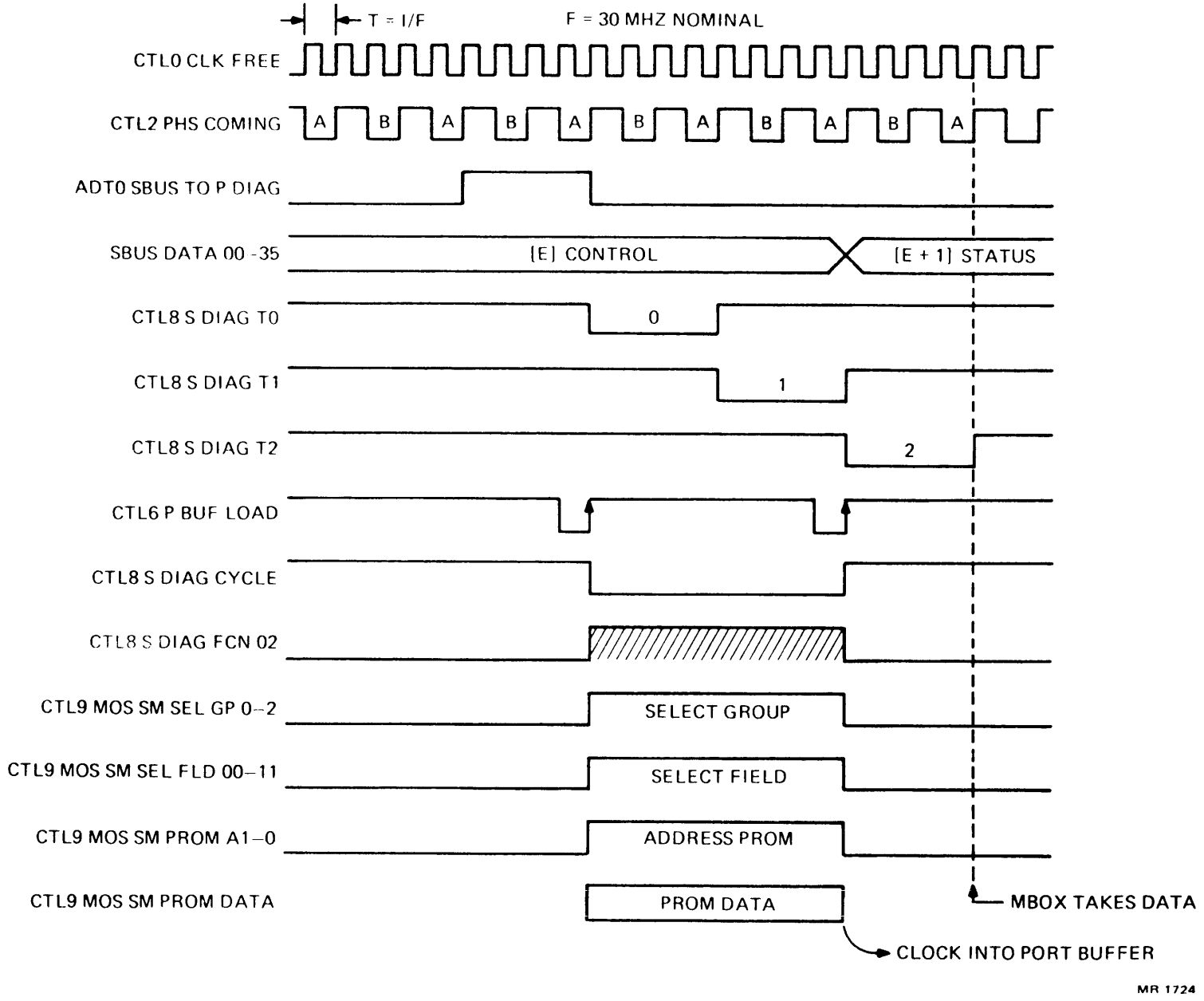


Figure 5-19 Diagnostic Function Cycle Rough Timing Diagram

and CTL8 S DIAG FCN 02. During S DIAG CYCLE the selected PROM is read and its data appears back on the CTL9 print as CTL9 MOS PROM DATA 0-7. On the phase A clock coincident with T1, a second CTL6 P BUF LOAD is generated which allows clocking the data into the port data buffer. From here it is gated out onto the XBus from where it can be taken in by the MBox and stored in [E+1]. The MBox is synchronized to strobe in the XBus data lines on the fourth A phase clock tick following the initial assertion of XBUS DIAG.

5.7.5 Summary

To perform the basic system functions involved with storing and retrieving information from the MOS storage array, the MF20 executes various combinations of four basic functions as directed by the MBox. The read fetches instructions. The write stores data read from a disk. The read-pause-write retrieves an operand, modifies it, and stores it back in MOS. Finally, the diagnostic function retrieves MF20 status for error analysis. Understanding these four operations is basic to comprehending the MF20.

5.8 MF20 LOGIC DESCRIPTIONS

Paragraphs 5.1 through 5.7 showed how the MF20 consists of a network of data and control facilities interconnected to perform the basic hardware functions required to store and retrieve data to/from a MOS storage array. Specific functional logic areas were identified and their interaction during read/write operations was discussed. Description of the detailed logic operation of these functional areas was generally omitted in order to emphasize total system operations. This paragraph will discuss the theory and operation of most of these functional logic areas. The level of discussion assumes that the reader is familiar with DIGITAL's signal naming conventions and logic print symbology.

5.8.1 Error Check Code - ECC

The MF20 uses a simple parity generation and checking scheme that permits detection and correction of a single-bit error (SBE) and detection only of a double-bit error (DBE). Failure modes that cause changes in three or more bits may or may not be detected but can never be corrected. To provide the ECC features the MF20 includes the following additional hardware.

1. Logic to generate seven check bits during any memory write

2. MOS storage elements to store the check bits for each word (approximately a 15 percent increase in required storage capacity)

3. Logic to check the data and check bits during any memory read

4. Diagnostic functions to retrieve the ECC information for fault analysis and error logging

This paragraph begins with a description of the ECC algorithm chosen to illustrate how the check bits are generated and checked. Next the ECC generation logic is discussed, followed by an explanation of the logic that supports the checking and correction processes.

5.8.1.1 **ECC Algorithm** (Figure 5-20) - A modified Hamming code was chosen to implement the ECC feature in the MF20. This scheme requires seven additional check bits be appended to each 36-bit word stored in the MOS storage array. The state of each check bit is a function of the states of the bits within overlapping groups of data bits. Figure 5-20 defines the following associations between data and check bits.

ECC Bit Positions	Data Bit Associations
41 (GEN 1)	0,1,3,4,6,8,10,11,13,15,17,19,21,23, 25,26,28,30,32,34
40 (GEN 2)	0,2,3,5,6,9,10,11,13,15,17,19,21,23, 25,26,28,30,32,34
39 (GEN 4)	1,2,3,7,8,9,10,18,19,20,21,22,23,24, 25,29,30,31,32
38 (GEN 8)	4,5,6,7,8,9,10,18,19,20,21,22,23,24, 25,33,34,35
37 (GEN 16)	11,12,13,14,15,16,17,18,19,20,21,22, 23,24,25
36 (GEN 32)	26,27,28,29,30,31,32,33,34,35
42 (GEN PAR)	0,1,2,4,5,7,10,11,12,14,17,18,21,23, 24,26,27,29,32,33

When generating the check bits, the logic asserts or negates the check bit to generate odd parity for all the bits in the group plus the check bit. For example, if the data word 252525,,252525 (alternating 1/0 pattern) were to be stored, check bits 37, 39, 40, 41, and 42 would be asserted and bits 36 and 38 negated. Note in Figure 5-20 that the total number of cross-hatched boxes (1s) in each of the seven vertical columns is an odd number. Once generated, the ECC bits (36-42) are appended to the data bits and stored in the MOS storage array.

During subsequent read operations the identical check bit/data bit groupings are checked for parity (7 groups of 7 overlapping parity checkers).

The result of this parity check is used to generate a 6-bit variable called a syndrome. Along with the syndrome the overall parity of all bits is checked and saved. Five possible combinations can occur as a result of the ECC check. They are

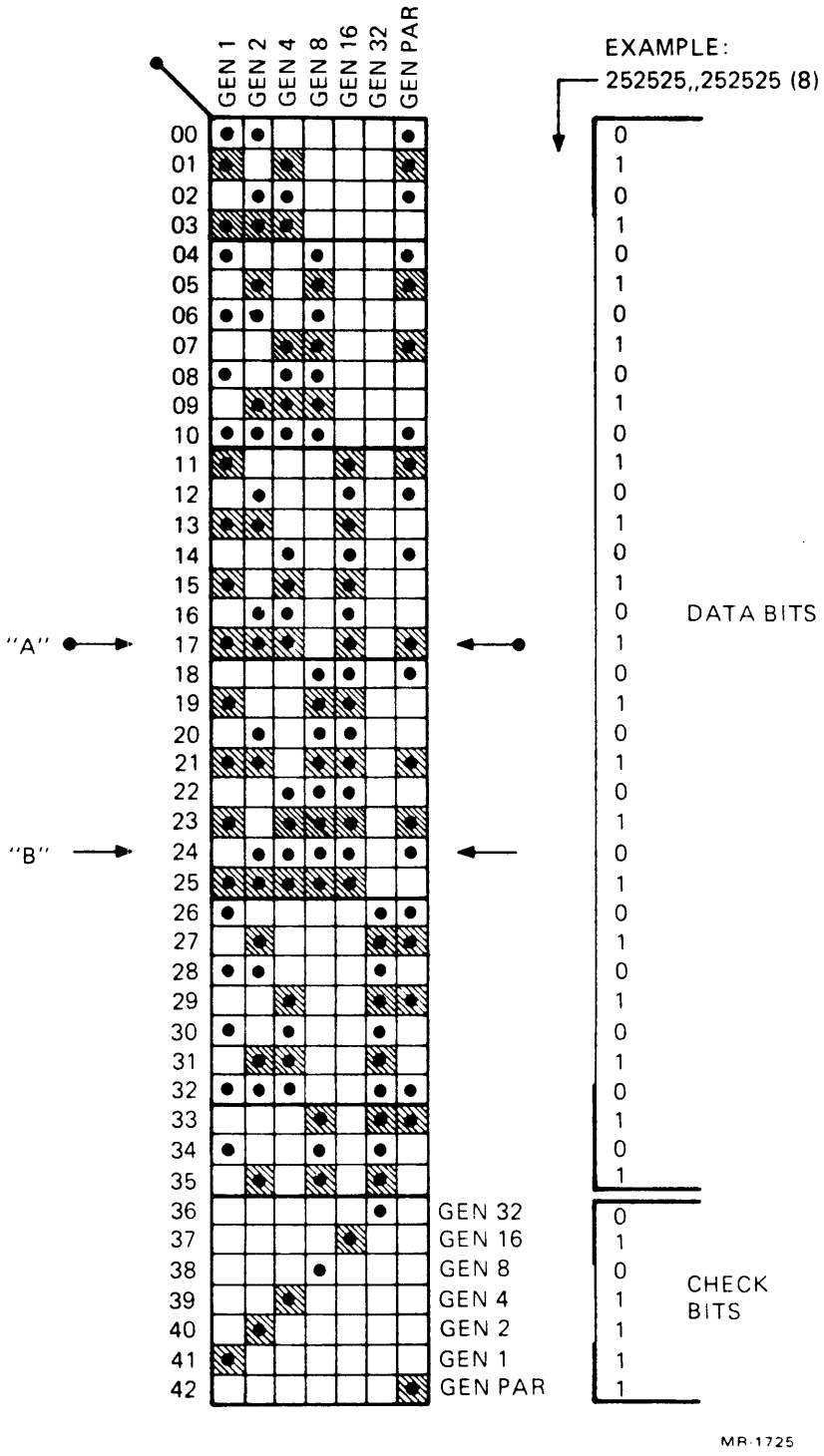


Figure 5-20 ECC Coding Format

listed below.

Syndrome	Parity	Interpretation
All 0s	odd	No error
All 0s	even	Parity bit is bad, data is correct, but flag an XBus error.
Nonzero but only a single bit is set	even	ECC check bit is bad, data is correct, but flag an XBus error.
Nonzero and more than one bit is set	even	Single data bit is bad. Use syndrome to correct data and flag an XBus error.
Nonzero and more than one bit is set	odd	Double-bit error - uncorrectable; flag an XBus error.

For SBEs the value of the syndrome is symptomatic of the bit position in error; therefore, it can be decoded and used to complement the bit position in error before sending the information read back to the MBox.

5.8.1.2 ECC Generation (Figure 5-21) - All the logic that supports the generation of the ECC bits is contained on the M8574 WRP module. Thirty-seven bits of information, SBUS D00-35, DATA PAR, enter this module from the XBus (WRP0, 1); and forty-four bits, MOS WR DATA D00-43A, leave the module to be sent to the MOS storage modules (WRP5, 7, 9). Within the boundaries of the M8574 module, the 36 data bits are decoded and used to generate the seven check bits, D36-42, that will be stored in the MOS array along with the data.

The XBus data signals plus parity are fed through 37 DC009 transceivers shown on the WRP0, 1 prints. From here, 37 signals, SBUS TO P D00-35, DATA PAR are transmitted to the port buffer multiplexer where they are steered into the port data buffer to become P BUF D00-35, DATA PAR. To properly select the multiplexer chips and allow clocking the port buffer chips, the following control signals are used.

P BUF RD SEL - must be negated to select the SBus inputs to the port buffer multiplexer

P BUF LOAD A-B - must be asserted to permit loading the port buffer

WRP0 CLK FREE - synchronizes actual loading of the multiplexer

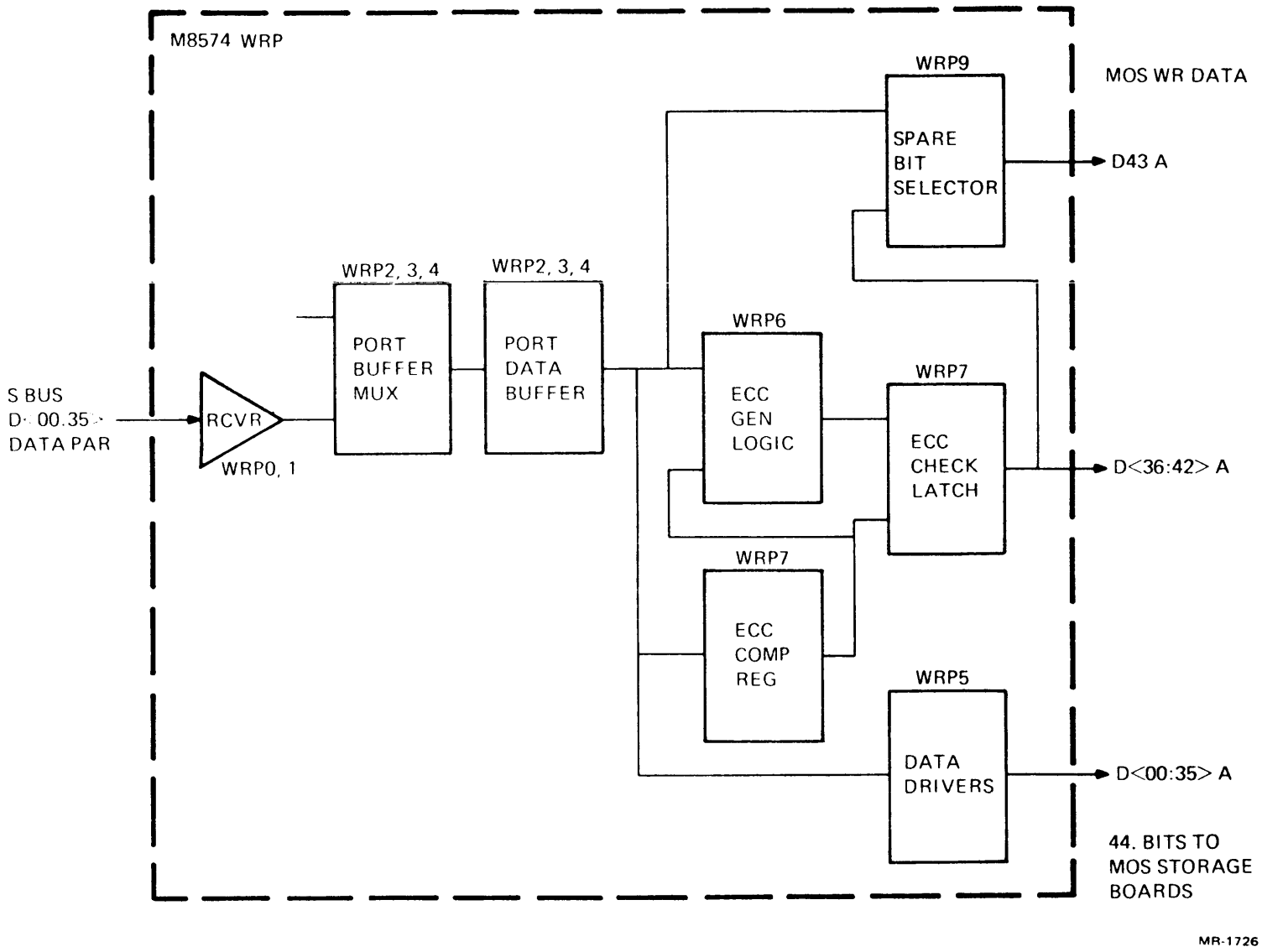


Figure 5-21 ECC Generation - Simplified Logic Block Diagram

output into the port buffer chips

Once loaded, the outputs from the port data buffer are used to activate the ECC generator logic on the WRP6 print. Five 10163 ECC generator chips along with eight 10160 parity chips are used to implement the previously discussed ECC algorithm. Using the example in Figure 5-20, an alternating 1/0 pattern, these chips would activate the following signals on WRP6.

ECC GEN 16
ECC GEN 4
ECC GEN 2
ECC GEN 1
ECC GEN PAR

This causes D36-42 to assume a pattern of 0101111. From here the ECC GEN signals are used to generate WRP7 MOS WR DATA D36-42A. The ECC bits can also be generated under control of the program for the purpose of diagnostic testing. This feature requires the use of an additional register, ECC COMP REG, that is loaded via a diagnostic function 6.

Along with the check bits, the contents of the port data buffer, P BUF D00-35, are sent through 36 10101 data drivers on WRP5. The spare bit selector on WRP9 looks back at both the data and check bits to provide the means of selecting any one of these 43 signals as the one to be stored in the spare bit MOS chip (MOS WR DATA D43A). Paragraph 5.8.2 describes the spare bit substitution mechanism.

If a bit is dropped or picked up on the way from the MBox to the ECC generation logic, the problem is detected by use of the XBus parity bit transmitted from the MBox during any write to memory. The signal WRP6 WP DATA PAR ERR will be asserted if bad parity is detected during a write and will cause the following occurrences.

1. The state of two check bits is complemented (WRP6 ECC GEN 32-16) and written into MOS. This forces a DBE if this location is subsequently accessed via a read.

2. XBUS ERROR is asserted to flag the program that an error has been detected in the MF20.

5.8.1.3 ECC Checking and Correction (Figure 5-22) - All of the logic to support ECC checking and correction is included on the M8575 SYN module. This paragraph describes how that logic operates using examples to explain its response during a memory read.

Data enters the M8575 from the MOS storage boards as a 44-bit parallel data path, MOS RD DATA D00-43 A (SYN1, 2), and, after checking and correction, leaves the M8575 as 37 bits, CHK TO P D00-35, DATA PAR (SYN6). From here the corrected data is sent to the port buffer register on the M8574 where it is gated back to

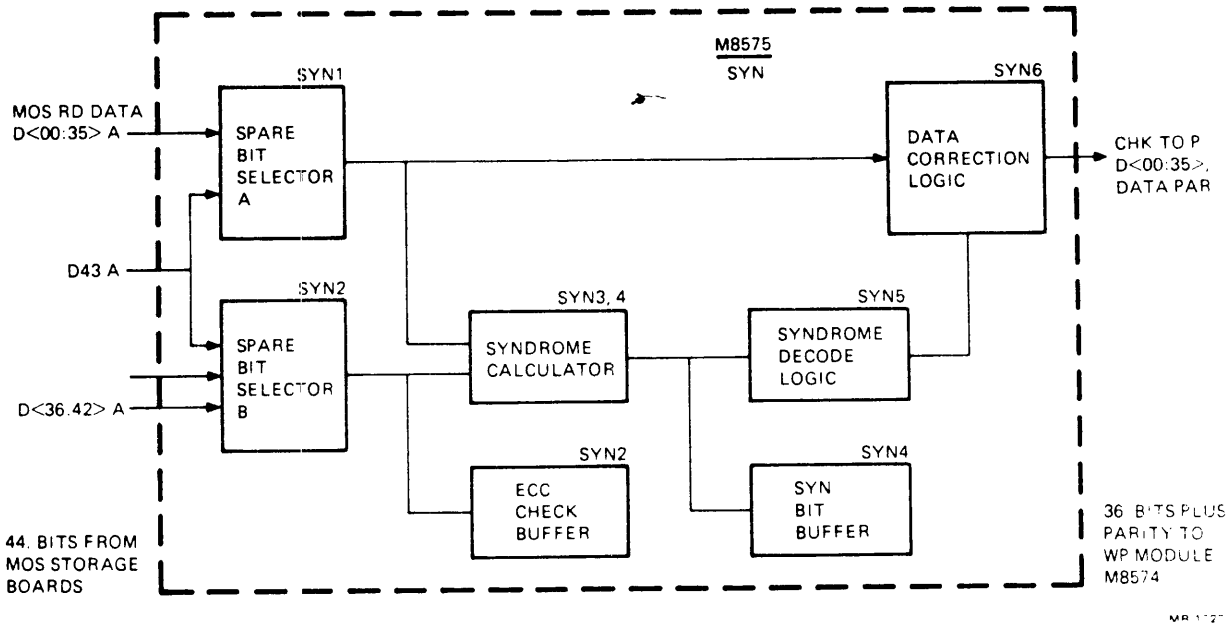


Figure 5-22 ECC Check and Correction

the MBox via the XBus.

The spare bit selectors A and B (SYN1, 2) substitute the spare bit, MOS RD DATA D43 A, into the proper bit position prior to sending the information to the syndrome calculator (SYN3, 4). The details of the spare bit mechanism are described in Paragraph 5.8.2.

Bit positions 36-42, the check bits, are saved in the ECC check buffer (SYN2) where they can be examined via a diagnostic function to permit software to analyze and log this information if an error occurs.

The syndrome calculator consists of a network of 10163 and 10160 ECC generator and parity chips which perform a parity calculation on the seven overlapping bit groups similar to the generation logic described in Paragraph 5.8.1.2. The major difference is that the ECC check bits themselves are included. If no error is detected the following responses exist.

1. The six signals SYN4 SYN BIT 32, 16, 8, 4, 2, and 1 are all negated - syndrome = 00_8.

2. The signal SYN4 SYN BIT PAR OK is asserted to indicate overall parity of the 43 bits was odd.

The syndrome is latched into a buffer (SYN4) where it can be examined by the program via a diagnostic function for the purpose of error logging and analysis. Finally, the syndrome decode logic (SYN5) uses the generated syndrome to control the data correction logic (SYN6).

From the output of spare bit selector A, the data passes through the data correction logic (SYN6) where SBEs are automatically corrected by complementing the bad bit position. The outputs, CHK TO P D00-35, DATA PAR, are transmitted over the backplane to the M8574 (WRP) module.

When an error is detected during a read, the resulting operation depends on classification of the error. The following cases will be discussed.

Case 1 - The parity bit is bad.
Case 2 - A single data bit is in error.
Case 3 - A double-bit error (data, ECC or a combination) occurs.
Case 4 - A single ECC bit is in error.

For case 1 the syndrome decode logic receives all 0s from the syndrome generator and no modification occurs when the 37 information bits pass through the data correction logic. The signal SYN4 - SYN 43 BIT PAR OK will be negated to flag bad overall parity. If correction is not disabled [-CORR DISABLE is asserted (Figure 5-23)], this generates an XBUS ERROR signal to

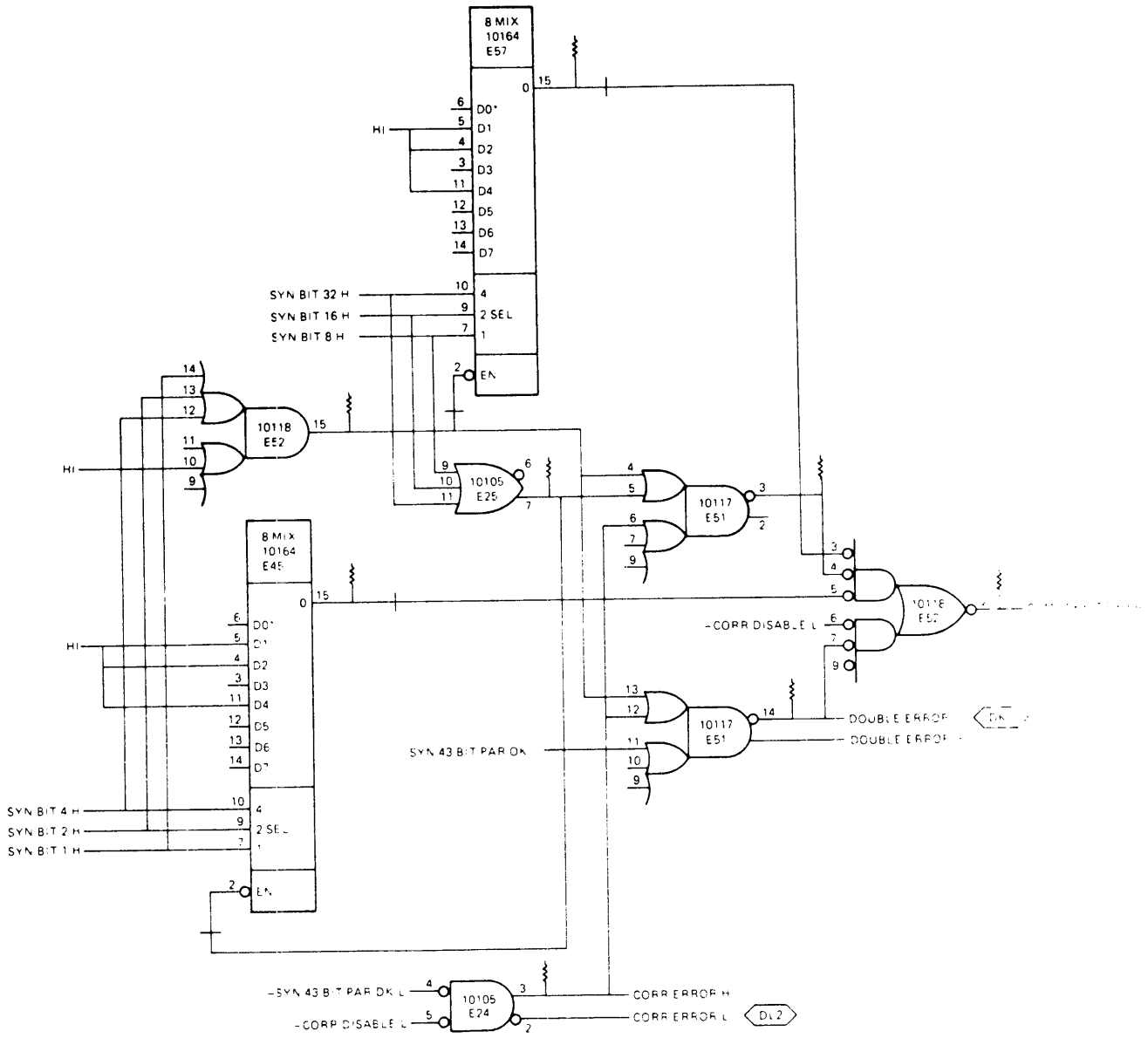


Figure 5-23 ECC Error Flags - Logic Diagram

warn the program that an error was detected even though the data was unaffected.

For case 2 the syndrome decode logic sees a nonzero syndrome (more than one bit set in the syndrome) and the signal SYN4 - SYN 43 BIT PAR OK negated. This flags an SBE that is correctable and the syndrome is symptomat c of the failing bit position. The syndrome decode logic uses the nonzero syndrome to activate the proper control signals to the data correction logic. The high-order syndrome bits (32, 16, 8) are decoded by a single 10161 (SYN5 E93), and if correction is not disabled, the outputs from this chip will enable one of six 10162 decoder chips. These second level decoders use syndrome bits (4, 2, 1) to generate a signal, SYN5 DECODE XX, where XX is the decimal equivalent of the 6-bit syndrome. This signal is sent to an XOR gate on SYN6 where it is used to complement the bit in error. Thus the error is corrected. Table 5-8 describes the relationship between syndrome, decode signal XX, and the bit position corrected. Since complementing one of the data bits disturbs the sense of the overall parity, the parity bit sent to the XBus (SYN6 CHK TO P DATA PAR) must be changed. This is achieved by asserting the signal SYN5 COMP CHK TO P PAR. Finally, the signal SYN4 - SYN 43 BIT PAR OK (refer to Figure 5-23) will be asserted and, if correction is not disabled, it will assert the signal CORR ERR that ultimately generates the XBUS ERROR signal to flag the program that an error has occurred.

In case 3, the syndrome calculator generates a nonzero syndrome and the signal SYN4 - SYN 43 BIT PAR OK is asserted indicating that two bits failed (still odd parity). This condition asserts the signal SYN5 DOUBLE ERROR which:

1. Inhibits the syndrome decode logic so that the bad data passes through the data correction logic unmodified

2. Generates an XBUS ERROR signal to flag the error. The program can read the state of SYN5 DOUBLE ERROR via a diagnostic function 0.

Case 4 causes bad overall parity to be detected which will generate an XBUS ERROR to flag the program, but the data doesn't need to be corrected since it was an ECC bit that failed. This case results in a syndrome which is an even power of 2 (32, 16, 8, 4, 2, or 1) which means only that one of the seven overlapping parity checks failed. Since each data bit affects two or more parity checks, the power-of-2 case could only occur if a check bit were bad.

Other modes of failure are possible, but the response of the error correction logic to these cases is unspecified. It should be possible to use the defined cases to flag impending multiple bit failures before they occur. This would permit the software to take appropriate action to minimize the impact of such failures on system operation.

Two examples of error correction will be discussed using Figure 5-20.

Example A

This example assumes the data word written was 252525,,252525 and on a subsequent read bit 17 failed. The cross-hatched squares show that during the write the check bits written were as follows.

```
GEN  32 = 0
GEN  16 = 1
GEN   8 = 0
GEN   4 = 1
GEN   2 = 1
GEN   1 = 1
GEN PAR = 1
```

During the read those check bit positions associated with bit 17 will be in error, namely PAR, 16, 4, 2, 1. This implies the following syndrome.

```
SYN BIT 32 16 8 4 2 1
        0  1  0 1 1 1
           2     7
Syndrome = 27(8)
```

From Table 5-8, a syndrome of 27_8 will activate SYN5 DECODE 27 which will complement M TO CHK D17 before generating SY06 CHK TO P D17.

Example B

This example also assumes a 252525,,252525 was written, but this time bit 24 has failed during the read. The cross-hatched areas show that the check bit positions associated with bit 24 are PAR, 16, 8, 4, and 2. During the read, these positions will be found in error and will generate the following.

```
SYN BIT 32 16 8 4 2 1
        0  1  1 1 1 0
           3     6
Syndrome = 36(8)
```

From Table 5-8, a syndrome of 36_8 will activate SYN5 DECODE 30 which will complement M TO CHK D24 before generating SYN6 CHK TO P D24.

Table 5-8 Syndrome Decodes

Syndrome (Octal)	Decode XX (Decimal)	Bit Position (Decimal)
03	03	00
05	05	01
06	06	02
07	07	03
11	09	04
12	10	05
13	11	06
14	12	07
15	13	08
16	14	09
17	15	10
21	17	11
22	18	12
23	19	13
24	20	14
25	21	15
26	22	16
27	23	17
30	24	18
31	25	19
32	26	20
33	27	21
34	28	22
35	29	23
36	30	24
37	31	25
41	33	26
42	34	27
43	35	28
44	36	29
45	37	30
46	38	31
47	39	32
50	40	33
51	41	34
52	42	35

5.8.1.4 Summary - All of the ECC logic in the MF20 is located on two modules, the M8574 and the M8575. This additional hardware provides the mechanism for detecting and correcting single-bit errors and detection only of double-bit errors. All errors detected can cause software interrupts to permit the program to log and analyze the faults within the MF20. The advantage of this process is increased availability and reduced repair time.

5.8.2 Spare Bit Substitution

Each set of four M8579 storage modules (a group) contains 16 MOS chips that provide a spare bit substitution mechanism in the MF20. Each chip maps two 8K (8192) segments which means that the spare substitution process affects the substituted bit position in 8192 locations. The program controls when the mechanism will be activated and what bit position is to be substituted by loading specific locations in a control RAM. Using XBus error interrupts followed by diagnostic function reads, the software retrieves the following error data.

1. Group no., block no, and sub-block no.
2. Type of error - correctable error
3. The syndrome - used to calculate the bit position no. to be substituted

Then the software sets up the substitution bit RAM to enable selection of the spare bit. Before doing this, the software must take the appropriate steps to correct the other 8191 locations affected by the substitution. This involves "writing" their current state into the spare bit chip.

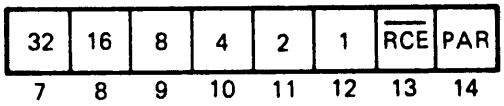
The actual mechanism consists of the following functional logic areas.

1. A programmable control RAM for selecting and activating the spare bit

2. Multiplexers that connect the selected data or ECC bit to the spare bit MOS chip during write

3. Multiplexers that connect the spare bit MOS chip to the selected ECC or data bit during a read

5.8.2.1 Spare Bit Substitution RAM (Figure 5-24) - A set of eight 128 X 1 RAM chips are organized into a 128 X 8 control memory used to activate and control the spare bit substitution process. The format of the bits in each location is described in Figure 5-24. The RAM operates in either diagnostic mode or normal read/write mode, with the latter dependent upon the former to establish the RAM contents. During diagnostic mode the RAM can be read or written and the RAM address is a function of the diagnostic data bits. In normal read/write mode the RAM is read-only and the address is a translation of the XBus address.



BIT 7 WRP8 SUB RAM 32 BIT 8 WRP8 SUB RAM 16 BIT 9 WRP8 SUB RAM 8 BIT 10 WRP8 SUB RAM 4 BIT 11 WRP8 SUB RAM 2 BIT 12 WRP8 SUB RAM 1	THESE SIX BITS ARE LOADED WITH THE BIT POSITION NUMBER TO BE SUBSTITUTED. INITIALLY ALL ACTIVE LOCATIONS ARE LOADED WITH THE NO 43. (POINT SPARE BIT TO ITSELF.)
BIT 13 –WRP8 RPT CORR ERR	THIS BIT, IF SET TO A "1" INHIBITS REPORTING CORRECTABLE ERRORS. NORMALLY SET TO A "1" AFTER ACTIVATING THE SPARE BIT.
BIT 14 ECC SUB RAM PAR	THIS BIT IS SET TO ESTABLISH ODD PARITY WHEN THE SPARE BIT LOCATION IS LOADED. PARITY CHECKING IS PERFORMED EACH TIME THE RAM IS ACCESSED DURING NORMAL R/W OPERATION.

NOTE:
A DIAGNOSTIC FUNCTION 07 IS USED TO ACCESS THE SPARE BIT RAM.

MR-1729

Figure 5-24 Spare Bit RAM Format

The 128 locations in the RAM must provide spare bit control for up to three groups of M8579 storage modules on a single MF20. For 16K MOS RAM chips the spare bit RAM is partitioned into four 32-location segments, with only the first three being used (96 locations). During normal read/write operations the last 32 locations should never be referenced. At configuration time these unused locations are loaded with even parity to force a control RAM parity error should they ever be inadvertently accessed by the hardware. The remaining three segments are used to map the available spare bits to logical memory addresses (refer to Figure 5-25). The correspondence between RAM address and logical memory address is as follows.

Locations 000 - 037	map addresses in group 0
Locations 040 - 077	map addresses in group 1
Locations 100 - 137	map addresses in group 2
Locations 140 - 177	XNU

Each group of 32 is further subdivided into four segments of 8 locations as follows.

Group No.	Block 0	Block 1	Block 2	Block 3
0	000 - 007	010 - 017	020 - 027	030 - 037
1	040 - 047	050 - 057	060 - 061	070 - 077
2	100 - 107	110 - 117	120 - 127	130 - 137

Finally, the relationship between sub-block and RAM address becomes apparent. Figure 5-25 shows this relationship for group 0. The same pattern exists for groups 2 and 3 with changes only in the actual RAM address and spare bit chip activated.

Figure 5-25 shows most locations containing a 43 which prevents substitution of the spare bit for most of the addresses in group 0. A and B indicate that sub-block 2's spare bit chip is being used to substitute bits 21 and 16 in two different 8K areas. This is possible only if the failure mode in the MOS chip affects two separate 8K areas in the 16K chip. C indicates a failure mode in block 1, sub-block 3 that affects all 16K bits in position 15. In this case both locations 13 and 17 in the RAM must activate and select bit position 15.

In summary, the number of spare bit usages is exhaustible; therefore, the software must maintain a history of which bits are free and which ones are in use. When the supply of spare bits is exhausted, the software must reconfigure the available address space to "patch out" areas where no spare bits are available to correct the possibility of DBEs. It will continue to run with a reduction in the total available MOS storage resource.

5.8.2.2 Diagnostic Mode Operation - After determining which address and bit position is failing the program executes a diagnostic function 7 with D15 = 1 to load the proper RAM location. (Refer to Figure 5-26.) WRP8 WP DIAG CYCLE and CTL6 S

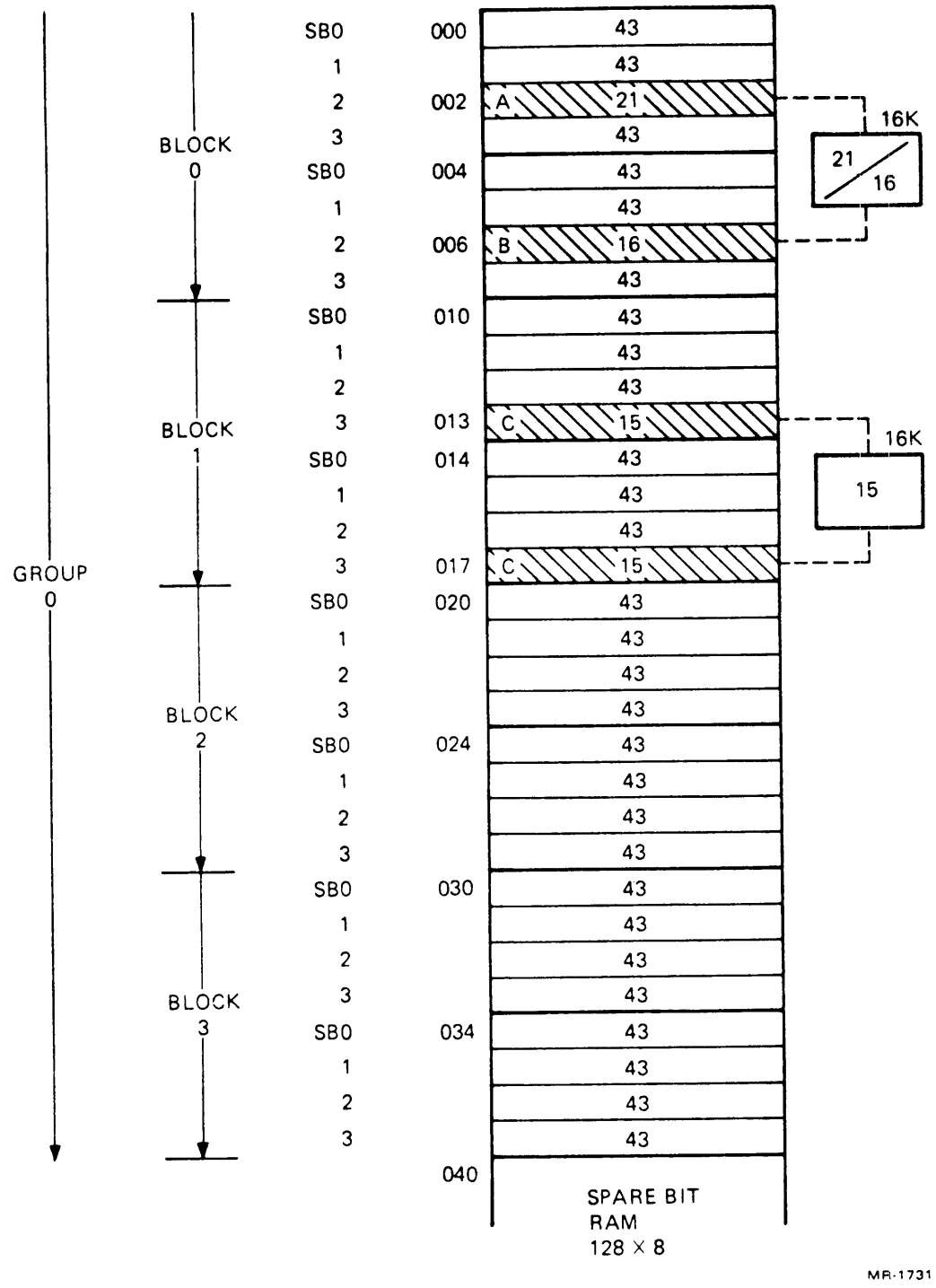


Figure 5-25 Spare Bit RAM Mapping

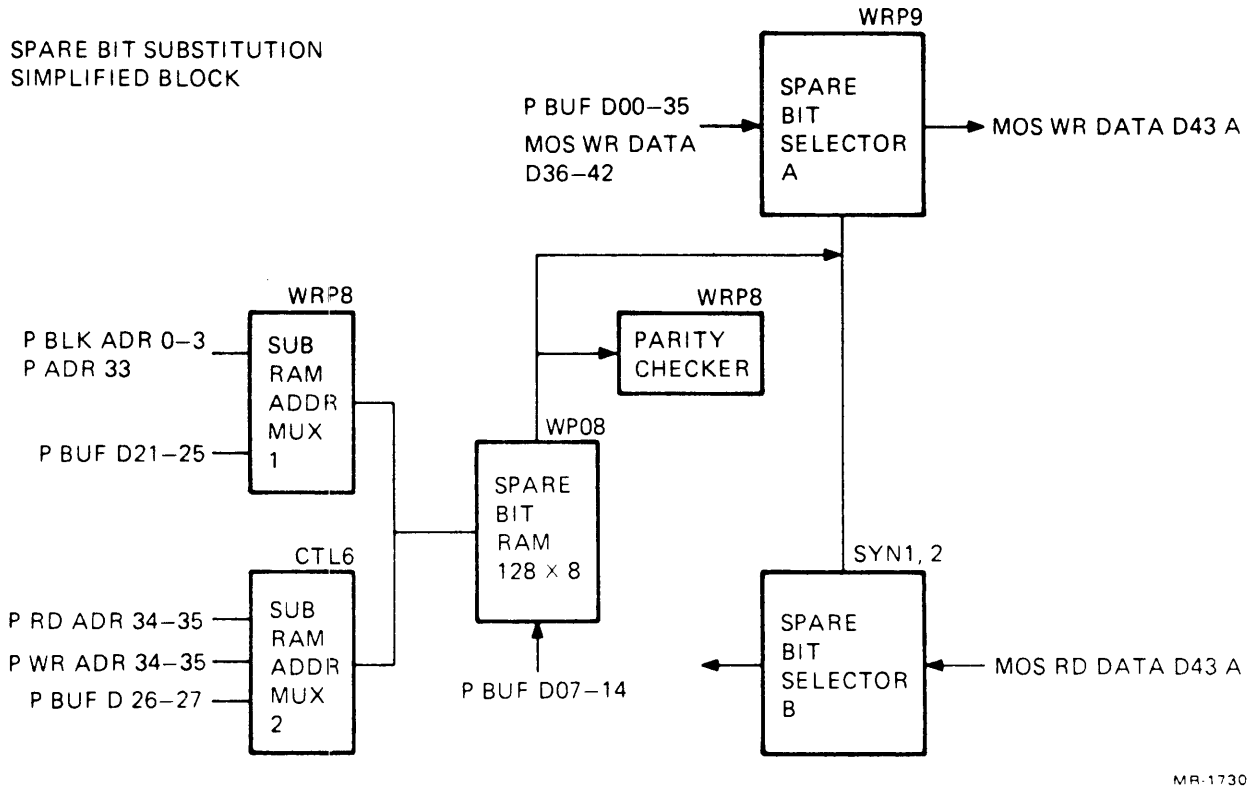


Figure 5-26 Spare Bit Substitution - Simplified Block Diagram

DIAG FCN 07 are asserted to switch the substitute RAM address multiplexers 1 and 2 so that P BUF D21-27 control the RAM address. These bits are set up by the program to point to the proper RAM location. WRP8 FCN 07 and P BUF D15 are asserted to enable the RAM to be written at WP DIAG T0+1 time. The data present in positions P BUF D07-14, also set up by the program, are written into the specified location at this time.

When the program executes a diagnostic function 7 with D15 = 0, the program reads back the RAM to determine its contents in the following way. The same function is executed but this time the contents in the addressed location are not modified. Instead, they are read out (8 bits) and transferred back to the MBox in data positions D07-14 on the XBus. During the read a parity check is made and the signal WRP8 ECC SUB RAM PAR OK is negated. This causes an XBUS ERROR to flag the program that the spare bit RAM failed. The basic 11-based MF20 diagnostics will use the diagnostic function 7 write followed by a read to test and report the condition of the spare bit logic.

5.8.2.3 Normal Read/Write Operation (Figure 5-26) - During normal MF20 read/write accesses by the program the substitution RAM address multiplexers 1 and 2 are in the opposite position since WRP8 WP DIAG CYCLE is negated. This permits the actual logical address being referenced to determine the address fed to the spare bit RAMs as follows.

P BLK ADR 0-3	These come from the address response RAM (SYN7) and are used to define group and block number.
P ADR 33	This defines one of two possible 8K segments within a 16K partition.
P RD ADR 34-35 or P WR ADR 34-35	These come from the cycle control and timing logic (CTL6) and define the sub-block number.

Under these conditions the RAM is read-only and the contents of the addressed location are sent to (WRP9) spare bit selector A and (SYN1, 2) spare bit selector B to activate spare bit selection. If the bit position number is 43, no substitution occurs. Parity is also checked, and if bad, WRP8 ECC SUB RAM PAR OK will be negated to generate a subsequent XBUS ERROR to flag the program. The actual reading and writing of data is unaffected - only the error flag is raised.

WRP9 shows seven 10164 mixer chips used to perform the spare bit selection function during write cycles. This logic network receives:

1. 43 data/ECC inputs
2. 6 substitution RAM control inputs.

It generates a single output, WRP9 MOS WR DATA D43 (the spare

bit). ECC SUB RAM 32, 16, 8, 4, 2, and 1 are the outputs from the spare bit RAM and are encoded with a bit position number. The low-order three bits (4, 2, 1) select one of the inputs to six multiplexer chips, E44, 14, 31, 45, 39, and 40. The high-order bits (32, 16, 8) are used to make the final selection using E28 to generate bit D43, the spare bit. For example, suppose bit 20 must be selected ($20_{10} = 24_8$). The contents of the RAM would be as follows.

WRP8 SUB RAM	32	16	8	4	2	1
	0	1	0	1	0	0
Selects	Input 2 on E28			Input 4 on E31		

This makes the spare bit D43 a function of P BUF 20.

During a read, the selectors on SYN1, 2 perform the reverse process. Eighteen 10174 2 X 4 mixers on WRP1 handle the data bits while seven 10164 mixers handle the ECC bits on WRP2. Three 10162 decoder chips on SYN0 decode the bit position number from the spare bit RAM (WRP8 SUB RAM 32, 16, 8, 4, 2, 1) and generate 22 even/odd multiplexer select signals used to select a pair of adjacent bits. SYN0 SUB RAM 1A-C assume the state of the LSB, ECC SUB RAM 1, and are used to select either odd (SUB RAM 1 = 1) or even (SUB RAM 1 = 0).

For example, suppose D10 must be substituted. This means SYN1 M TO CHK D10 must become a function of M TO CHK D43B. E89 on SYN1 is controlled by SUB SPARE 10, 11 and SUB RAM 1A. The first is asserted and the second negated for substituting bit 10, an even bit. For this case the SEL 2-1 inputs to E89 are 10_2 which selects pins 4 and 12 of the multiplexer inputs, allowing the proper substitution to occur. Note that the data mixers are disabled during a diagnostic cycle and an additional input to the ECC mixers (SYN2), S DIAG FORCE 0's permits program control of SYN 2 M TO CHK ECC 32, 16, 8, 4, 2, 1, PAR.

5.8.2.4 Summary - The MF20 spare-bit logic is a feature that improves system availability. This mechanism, along with ECC correction, reduces the likelihood of simple bit faults causing system failures. Most of the hardware to support the feature is on the (WRP) M8574 module with the exception of the spare bit selectors used during a read. These are on the M8575 (SYN) module. Additional diagnostic hardware provides a means of testing the substitution logic before turning it over to system software.

5.8.3 Storage Module Operation

This paragraph provides a detailed description of the operation of M8579 storage modules. It discusses the control signal sequences involved with storing and retrieving data to/from the MOS storage array. The organization of the storage array itself has already been described in Paragraph 5.3. Appendix B contains more

detailed descriptions of the 4116 MOS RAM chip and the DC008 Data multiplexer chip.

5.8.3.1 Simplified Block Diagram (Figure 5-27) - All data to be stored in the MOS array passes through the M8574 module and appears at the inputs to the M8579 storage modules as 44 signals, MOS WR DATA D00-43A. All data retrieved from the M8579s appears at the outputs as MOS RD DATA D00-43A and is sent to the M8575 module for checking and ultimate delivery to the MBox via the M8574 and the XBus. The selected group of M8579 modules can store 256K (262,144) 44-bit words and a single set of MF20 control boards can control up to three groups. Therefore the ADT (M8577) and CTL (M8576) modules must generate the control signals required to select the proper location within the storage array and ensure passage of the 44 bits to or from the 16K MOS chips.

5.8.3.2 One-Bit Slice (CS-M5979-0-SM03) - This diagram shows a 1-bit slice of the data path within the M8579 module. Each module print set contains 11 diagrams similar to this one to describe a complete field. Since all 44 bits operate the same, it is only necessary to describe one bit position (SM03 was chosen as the example - bit positions 2, 3, 20, or 21). Each 1-bit slice consists of the following.

1. Seven 36179 MOS driver chips to buffer the MOS address

2. Sixteen 4116 MOS RAM chips - Each chip stores 16,384 bits and is organized as follows.

	BLK 0	BLK 1	BLK 2	BLK 3
SB0	E76	E75	E74	E73
SB1	E72	E71	E70	E69
SB2	E68	E67	E66	E65
SB3	E64	E63	E62	E61

3. A single DC008 data multiplexer chip E59 that consists of:

   a. an upper register multiplexer section used for read cycles

   b. a lower buffered shift register section used for write cycles.

NOTES: (Refer to CS-M8579-0-SM03)

1. Each RAS/CAS signal is followed by a 2-digit number (XY) where:

   X = block number
   Y = sub-block number.

2. To determine bit position number the algebraic expression

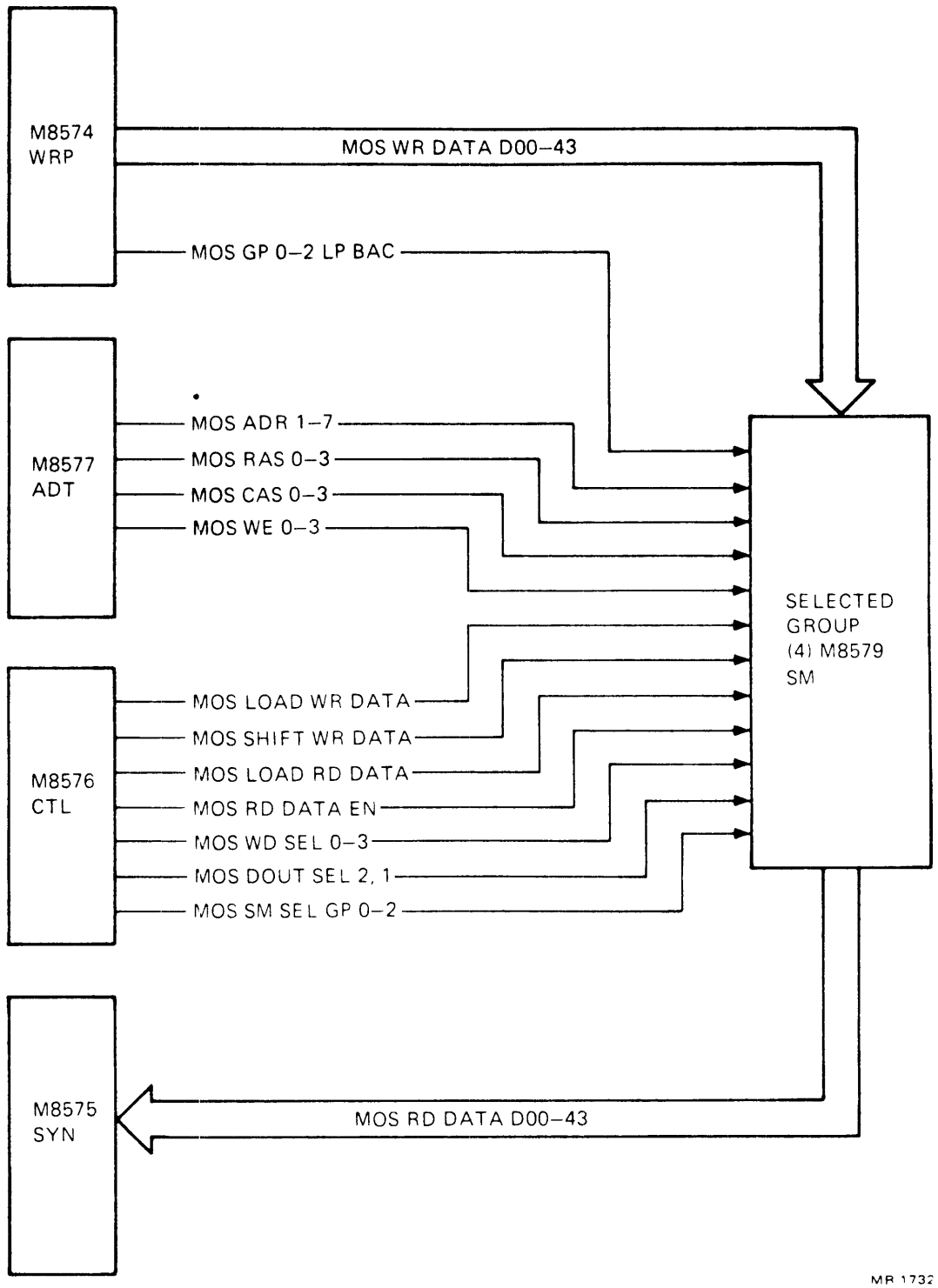


Figure 5-27 MOS Storage Module - Simplified Block Diagram

[T*18+02+U] must be evaluated where:

UT = Field No.

00	0
01	1
10	2
11	3

[N, A, B] specifies backplane slots for each group of four M8579s.

FLD	0	1	2	3
N (GRP0)	10	13	16	19
A (GRP1)	9	12	15	18
B (GRP2)	8	11	14	17

Example: Field 2 for group 1 is implemented by the M8579 that plugs into slot 15.

5.8.3.3 **Write Operation** - The following discussion describes the operation during a quadword write starting at an XBus address that starts on a quadword boundary. First, the data bits for each word, WD0, 1, 2, and 3 must be transmitted over the XBus from the MBox to the WRP module. Then they are shifted one bit at a time into the DC008 buffered shift register. At each SM CLK B, the data bit at the DATA IN input (pin 1) is shifted into QC; QC is shifted into QB; and QB is shifted into QA if the signal SM SHIFT WR DATA is asserted. After three bits have been shifted in, SM LOAD WR DATA is asserted along with the bit for WD3 being present at the DATA IN input. On the next SM CLK B the following transfers occur with the DC008.

D0 <= QA
D1 <= QB
D2 <= QC
D3 <= DATA IN

The bit position being written for each of the four words appears at the output of the DC008. The data is latched and ready to be written into the proper MOS RAM chips as follows.

SM D03 WD0 written into sub-block 0
SM D03 WD1 written into sub-block 1
SM D03 WD2 written into sub-block 2
SM D03 WD3 written into sub-block 3

The ADT module gates out the first half of the 14 bit MOS address, SM ADR 1-7 along with a row address strobe (RAS) as defined below.

RAS 00-03 if writing block 0
RAS 10-13 if writing block 1
RAS 20-23 if writing block 2

RAS 30-33 if writing block 3

This latches the row address within the four MOS chips in the addressed block. After RAS has latched the row address, the ADT module gates out the second half of the 14-bit address, also as SM ADR 1-7, along with a column address strobe (CAS) as defined below.

CAS 00-03 if writing block 0
CAS 10-13 if writing block 1
CAS 20-23 if writing block 2
CAS 30-33 if writing block 3

This latches the column address within the four MOS chips in the addressed block.

During the time CAS signals were establishing the column address, the appropriate write enable signals are activated to allow writing the data into the MOS storage elements as defined below.

SM WE 00-03 enables writing sub-block 0
SM WE 10-13 enables writing sub-block 1
SM WE 20-23 enables writing sub-block 2
SM WE 30-33 enables writing sub-block 3

During a quadword write, SM11 MOS WD SEL 0-3 would permit activating a complete set of write enable signals to enable writing four sub-blocks (4 MOS chips). For other than a quadword (1, 2, or 3), these signals would activate only the write enable signals specified by XBUS RQ 0-3.

5.8.3.4 Read Operation - During a read, the sequence of operations is reversed. RAS is asserted first along with the row address on SM ADR 1-7. Once the row address has been latched in the chip, CAS is asserted along with a column address on SM ADR 1-7. At this point one MOS storage element in each sub-block is selected and four bits are available at the data inputs to the DC008 register chip (pins 13-16). The signal SM LOAD RD DATA is asserted by the CTL module to allow the next SM CLK B to parallel load the quadword into the data multiplexer chip. Now SM RD DATA EN is asserted by the CTL module to enable the signals SM DOUT SEL 2-1 to select the order in which the words will be sent over to the SYN module on MOS RD DATA D00-43. These two signals are generated by the read data mover logic on the CTL module as a function of:

1. The state of XBUS RQ 0-3
2. The initial state of XBUS ADR 34-35.

The following table defines their use.

SEL 2	1	Word available at DC008 pin 11
0	0	WD0
0	1	WD1

1	0	WD2
1	1	WD3

As each word is sent over to the SYN module, it is checked for errors and sent to the WRP module's port buffer to be delivered to the XBus.

5.8.3.5 Summary - This section presented the interaction that must occur between the MF20 control section (ADT, CTL, WRP, and SYN) modules and the SM storage modules during storage or retrieval of data. Paragraph 5.7 discussed more detailed timing of these sequences.

5.8.4 Address Response RAM

Each MF20 controller (M8575 module) contains an address response RAM to provide software configurability. Using this feature, the software can reconfigure memory to "patch out" defective segments of MOS memory. This allows the system to continue to run but with reduced storage resources until corrective maintenance can be conveniently scheduled. In earlier DEC memory systems (MH10, MG10, etc.) reconfiguration required operator intervention to shut down the system, manually reconfigure the memory with physical switches, and then bring the system back on-line.

When the MF20 is initially powered up, the address response RAM must be loaded by the program to define the MF20 configuration. The configuring process involves setting up the RAM with specific information used to translate physical XBus addresses to map to the proper group and block within the MOS storage array. Remember that a single MF20 can contain up to three groups (one group being equivalent to four M8579 modules) with each group containing four 64K blocks. The task, therefore, is to set up the RAM to relate physical XBus address to a group number and a block number. The program uses a diagnostic function 12_8 to load the RAM during system initialization. Figures 5-28 and 5-29 further describe the purpose and format of the address response RAM.

5.8.4.1 Functional Block Diagram (Figure 5-30) - The RAM is implemented using seven 256 X 1 read/write memory chips. To define each location requires an 8-bit address, P ADR 14-21, which is provided by the address mixer latches that are loaded from either:

1. The XBus address (normal operation) or,
2. The XBus data lines (diagnostic mode).

During execution of a diagnostic function 12+05 the address latches are loaded from MOS WR DATA D20-27A, or from the signal set, SBUS TO P ADR 14-21 to make the RAM address a function of the XBus address being referenced by the program. If an error is detected, P ADR 14-21 will be latched into the error address latches. The program can read these latches with a diagnostic function 0 to determine which address caused the error.

Once the address to the RAM has stabilized, the MF20 control logic can use the outputs to perform the following functions.

1. Allow a response if the box was selected

2. Control generating the proper address width to the MOS chips depending on the chip type selected (4K-16K or 16K-64K)

3. Generate the proper group and block numbers:

   SYN7 P BLK ADR 0, 1 group no.

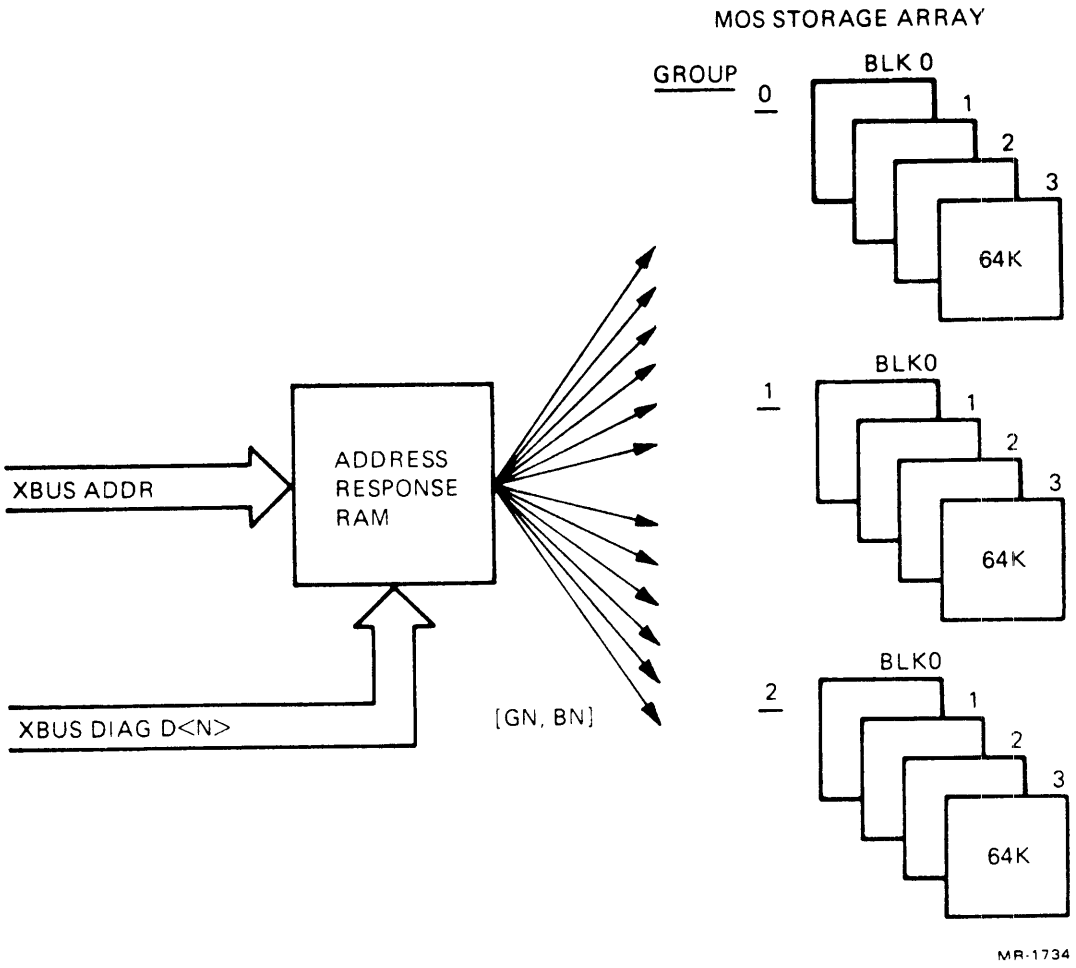
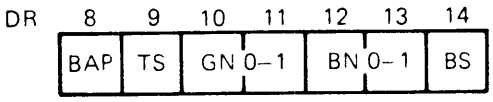


Figure 5-28 Address Response RAM Overview

BIT DESIGNATION	D<N>	FUNCTION
BLK ADR <0:1>	<10:11>	THIS TWO BIT FIELD IS SET UP TO SPECIFY THE GROUP NUMBER (0–2).
P BLK ADR <2:3>	<12:13>	THIS TWO BIT FIELD IS SET UP TO SPECIFY THE BLOCK NUMBER (0–3).
TYPE SELECT	<09>	THIS BIT IS SET UP TO INDICATE THE SIZE OF MOS CHIP USE IN THE STORAGE ARRAY (16K, 64K, ETC.).
–BOX SELECT	<14>	THIS BIT IS SET TO A "1" TO PREVENT BAD BLOCKS OR NON-EXISTANT BLOCKS FROM RESPONDING TO THE XBUS ADDRESS REFERENCE.
BLK ADR PAR	<08>	THIS BIT IS SET UP TO ESTABLISH ODD PARITY BASED ON THE STATE OF THE OTHER SIX BITS IN EACH LOCATION. EACH TIME THE RAM IS USED TO TRANSLATE ADDRESSES THE PARITY IS CHECKED AND A FATAL CONTROL ERROR IS FLAGGED IF BAD PARITY IS DETECTED.



MR-1735

Figure 5-29 Address Response RAM - Bit Format

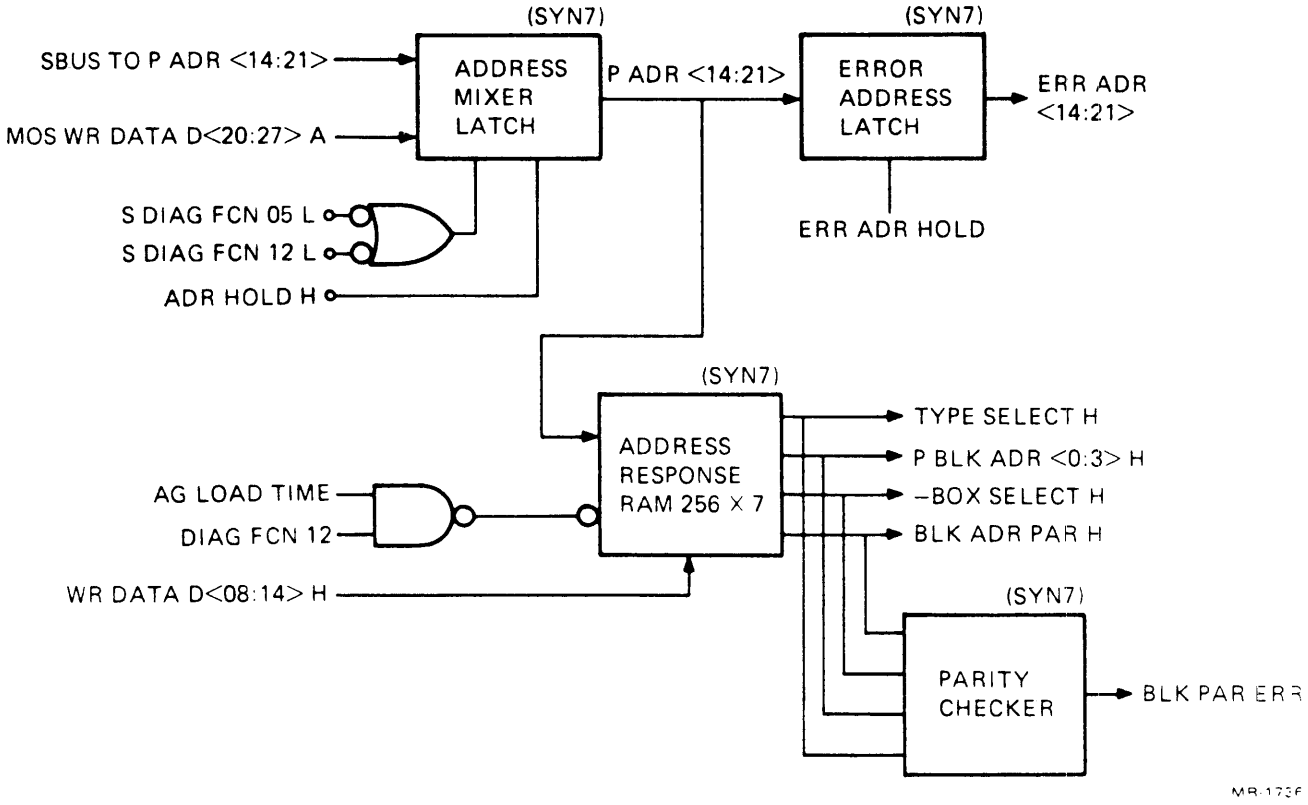


Figure 5-30 Address Response RAM - Simplified Block Diagram

SYN7 P BLK ADR 2, 3 block no.

4. Check for parity errors if a bit was dropped or picked up while accessing the RAM.

Any failure to properly establish the contents of the address response RAM will prevent the MF20 from executing its specified functions.

An XBus diagnostic function 12 is used to set up the RAM initially. The signal, S DIAG FCN 12, controls both the address mixer latches and the RAM itself. During execution of a function 12 the program encodes the data lines [E] as follows.

D15	Controls whether the program will read (=0) or write (=1) the RAM
D20-27	Specifies one of 256 locations within the RAM
D08-14	Specifies the data to be written into the RAM if a write (D15 = 1) is performed

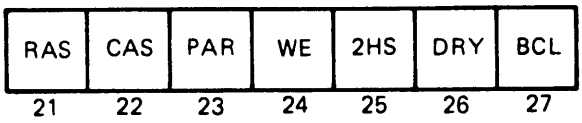
Diagnostic software uses the diagnostic function 12 to write various patterns into the RAM and read them back for verification. It can also force bad parity to test the parity checking logic. The diagnostic programs also use a diagnostic function 5 to provide the RAM with an address during single-step operation.

5.8.4.2 Summary - The address response RAM is loaded during system start-up to configure the MF20. It can be modified during run time to reconfigure the memory to patch out bad blocks. Once configured, this RAM provides the necessary linkage between XBus address and actual group/block numbers within the MF20.

5.8.5 Timing RAM

In most memory systems, read/write timing sequences are hardwired into the logic and are chosen to handle worst-case conditions. This is not the case in the MF20. The pulse widths and repetition rates of the primary timing signals are specified by the contents of a program-loaded RAM called the timing RAM. This firmware mechanism provides a method of modifying MOS chip timing to comply with different chip manufacturers' specifications. It also makes it easier to modify timing to take advantage of faster chip speeds as they become available without extensive circuit changes within the MF20. Timing can be modified by simply loading a different set of bit patterns into the RAM.

5.8.5.1 Functional Block Diagram (Figure 5-31) - The timing RAM consists of seven 128 X 1 RAM chips arranged to provide a control memory capable of storing 128 control characters. Figure 5-32 describes the bit format of each character in the RAM. (At present only the first 14_8 locations are being utilized.) The following functional logic areas are required to support the RAM.



BIT NUMBER	SIGNAL NAME	FUNCTION
21	ADT4 TIM RAM RAS	USED TO ASSERT THE ROW ADDRESS STROBE SIGNALS.
22	ADT4 TIM RAM CAS	USED TO ASSERT THE COLUMN ADDRESS STROBE SIGNALS.
23	ADT4 TIM RAM PAR	USED TO ESTABLISH OVERALL ODD PARITY WHEN EACH LOCATION IS LOADED.
24	ADT4 TIM RAM WE	USED TO ACTIVATE THE WRITE PULSES TO THE MOS CHIPS.
25	ADT4 ADR 2ND HALF SEL	USED TO CONTROL SWITCHING OF MOS ADDRESS LINES FROM A ROW TO A COLUMN ADDRESS.
26	ADT4 TIM DATA RDY	USED TO INDICATE WHEN DATA IS READY DURING A READ FROM MOS.
27	ADT4 TIM RAM BUSY CLR	USED TO TERMINATE THE TIMING RAM SEQUENCE.

MR-1737

Figure 5-31 Timing RAM - Bit Format

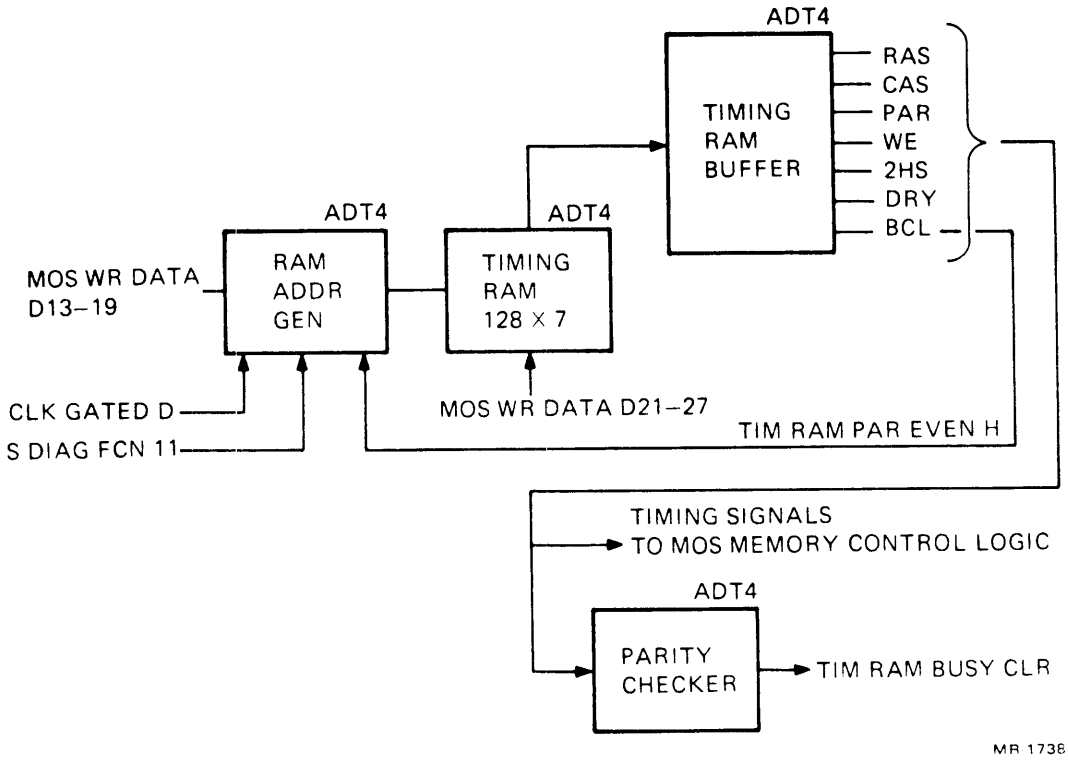


Figure 5-32 Timing RAM - Functional Block Diagram

1. A RAM address generator that provides the proper address signals to the RAM

2. A data buffer to hold control information read out of the RAM

3. A parity checker to verify the output of the RAM as each location is accessed

To load the RAM, the program executes a diagnostic function 11 (ADT4). During the function 11 the program can activate the write signal to the RAM chips (pin 12) if it asserts MOS WR DATA D20B. At the same time it places the RAM address in positions MOS WR DATA D13-19B, and the data to be written in positions MOS WR DATA D21-27B. At this time the signal TIM RAM BUSY IN will be negated, which forces the two 10136 binary counter chips (ADT4) to operate as an address register being loaded from MOS WR DATA D13-19B. The program must execute a separate diagnostic function 11 for each location to be loaded. Following each write operation, a diagnostic function 11 could be executed with D20B = 0 to read back each location for verification.

Once loaded the contents of the RAM establish the proper MOS timing sequence used for read, write, and refresh cycles. When a read or write is initiated from the XBus control, the signal CTL1 RAS GO is asserted which enables the 10136 binary counter chips E58 and E76 (ADT4) to start up counting from 000 when toggled by the signal CLK GATED F.

The counter sequences the RAM addresses from 000_8 through 014_8. At each CLK F the contents of the RAM locations control the turning on and off of the MOS timing signals until the signal ADT4 TIM RAM BUSY CLR is turned on by location 017_8. This signal negates ADT4 TIM RAM BUSY IN which stops the counter and resets it to 000 on the next clock. Figure 5-33 describes the contents of the RAM and shows a rough timing diagram of the signal sequence generated.

As each RAM location is read, a 10160 parity chip E45 (ADT4) tests the validity of the data. If invalid data is sensed (even parity) the signal AT04 TIM RAM PAR ERR is asserted which causes an XBUS ERROR to flag the program that a control error was detected. During setup the program will load all unused locations to contain bad parity so that any undefined access will generate an error interrupt.

5.8.5.2 **Summary** - The timing RAM and its support logic are contained on the M8577 module and supply the MOS timing signal sequence necessary to control reading and writing the MOS storage array. It is initialized by system software at configuration time and may be altered to tailor control timing sequences to individual MOS chip manufacturers' specifications.

ADDR	21	22	23	24	25	26	27
000	1	0	0	1	0	1	0
001	1	0	0	0	1	1	0
002	1	0	0	0	1	1	0
003	1	1	1	0	1	1	0
004	1	1	1	0	1	1	0
005	1	1	0	0	1	0	0
006	1	1	1	0	1	1	0
007	1	1	1	0	1	1	0
010	0	0	1	1	0	1	0
011	0	0	1	1	0	1	0
012	0	0	0	1	0	1	1
013	0	0	1	1	0	1	0
014	0	0	0	1	0	1	0
015	0	0	0	1	0	1	0
016	0	0	0	1	0	1	0
017	0	0	0	1	0	1	0
020	0	0	0	1	0	1	0
021	1ST UNUSED LOCATION*						

*NOTE:
ALL UNUSED LOCATIONS
CONTAIN 050 (8), 0001010 (2).

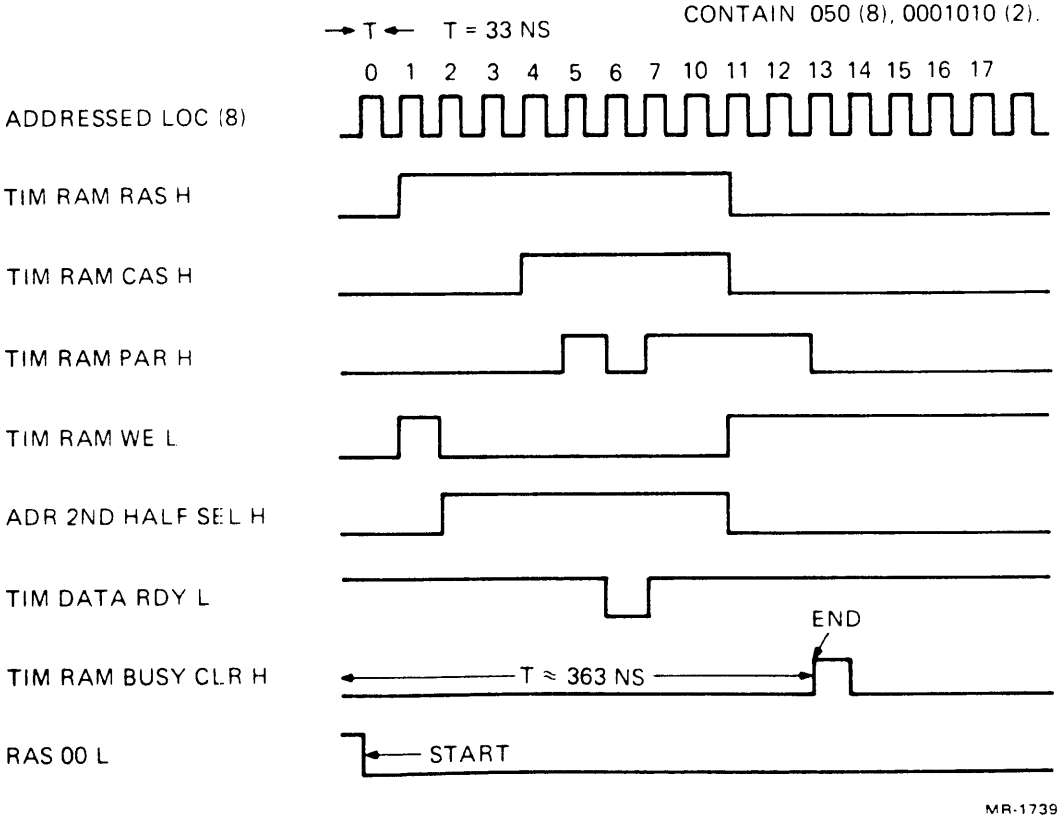


Figure 5-33 Timing RAM Map - Simplified Timing

5.8.6 Fixed Value RAMs

The M8576 control module contains four RAM chips (CTL4, 5); one 128 X 1 and three 256 X 1, that control generation of:

1. XBUS ACKN for both read and write
2. XBUS DATA VALID for a read
3. P RD ADR 34-35 for a read.

As will be shown shortly, the sequencing of these signals is a function of:

1. Which words in the quadword group are to be transferred (RQ 0-3) and

2. Which word is to be transferred first (ADR 34-35).

When configuring the memory, the program loads these RAMs (diagnostic function 3) with the bit patterns to define the proper control signal sequences. In effect the contents of these RAMs is a form of microcode for the MF20 control.

5.8.6.1 Functional Block Diagram (Figures 5-34 and 5-35) - When discussing the operation of the fixed value RAMs, it is convenient to define the following two possible modes.

1. Diagnostic Mode - permits loading and reading back the RAMs using a diagnostic function 3

2. On-Line Mode - sequences the RAM (read-only) to generate the control signals required to process the data

Since the ACKN RAM differs from the other members of the set of four, it will be discussed separately.

The ACKN RAM (CTL4) is implemented with a single 10147 (128 X 1) chip that generates the signal CTL4 P ACKN EN L whenever it is necessary to assert the XBUS ACKN signals A or B. The program uses a diagnostic function 3 to load the RAM with the I/O pattern shown in Figure 5-36. To load each bit in the RAM, the program must supply a 7-bit address to identify which location is to be loaded and the actual data bit, 1 or 0, to be written. This is achieved by placing the address in positions D20-27 (CTL7), the data in position D10 and executing a diagnostic function 3. The algorithm chosen to implement the correct RAM sequence during a read or write creates a minor interpretation problem when analyzing the RAM address to be used. Figure 5-37 attempts to clarify this situation.

During execution of diagnostic function 3, mixers E42 and E27 (CTL1) select DIAG ADR 2-5 as the source of address bits 0-3 (P RQ 0-3) (CTL4). The low-order portion of the address is derived from the binary counter chip E58 (CTL4) which is loaded from DIAG 1, 6, and 7. Once the address is established, the data at pin 11 (E25) (CTL4), a function of P BUF D10, can be written into the selected location. The actual writing occurs at S DIAG TO A time if D14 =

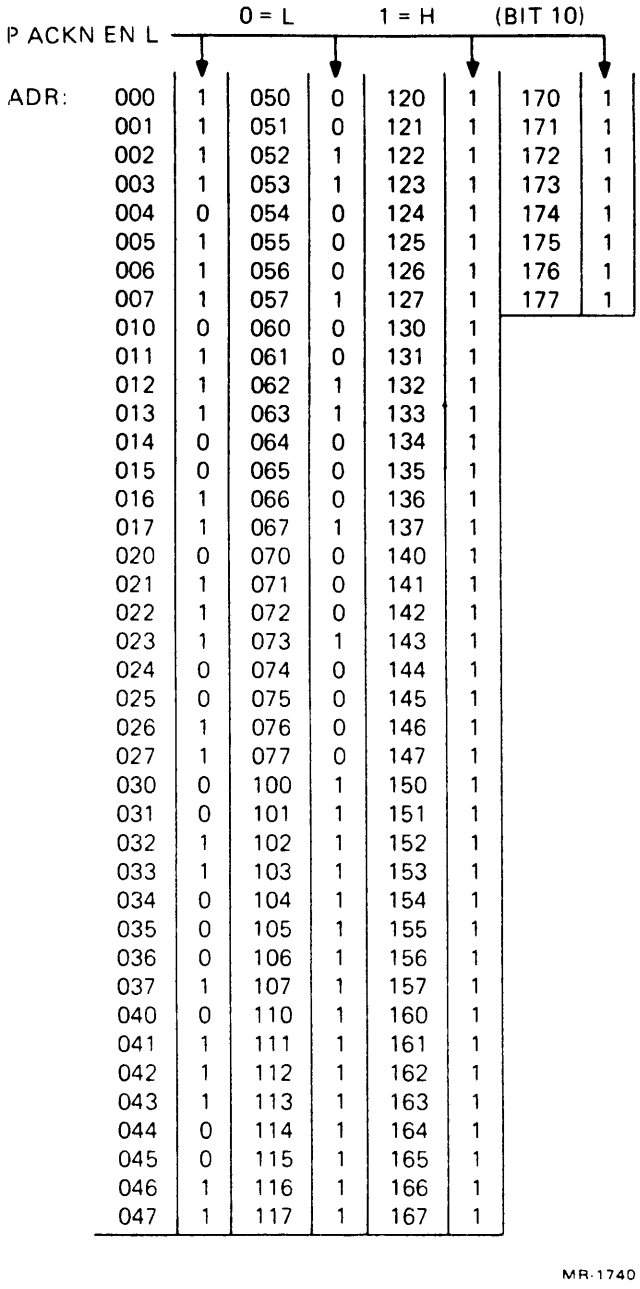
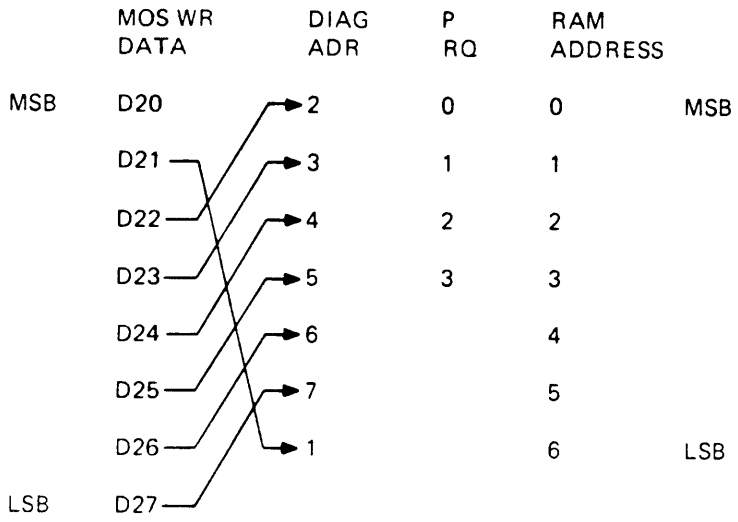
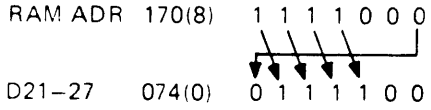


Figure 5-34 Fixed Value RAM - ACKN Bit Map



NOTES:

- THE ADDRESSES USED IN THE BIT PATTERNS OF FIGURE · ARE A FUNCTION OF D20–27.
- THE ADDRESS USED FOR ANALYSIS IS A FUNCTION OF RAM ADR 0–6.
- THIS TRANSLATION ONLY AFFECTS THE ACKN RAM.
- EXAMPLE:



MR-1741

Figure 5-35 ACKN RAM - Address Translation

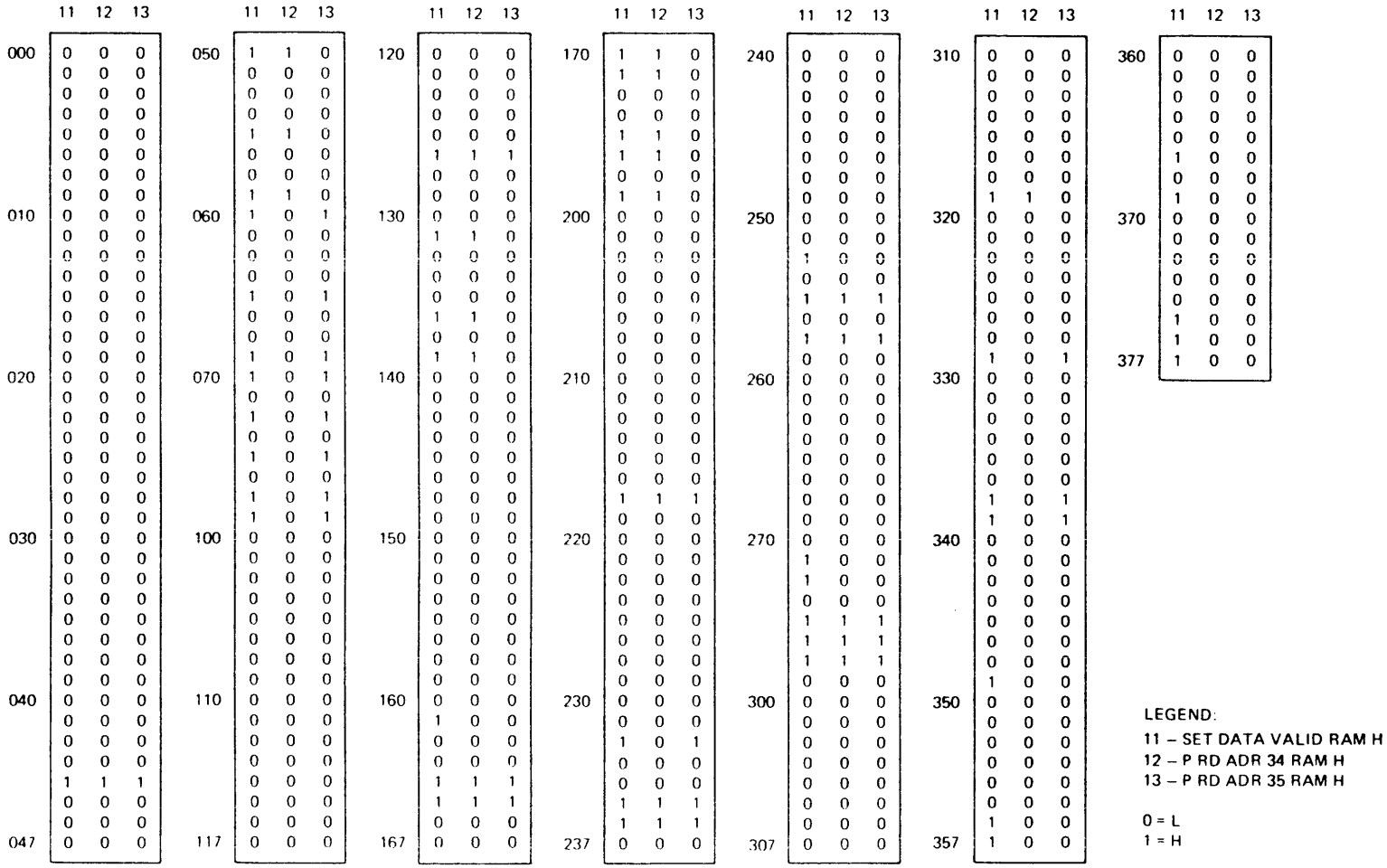


Figure 5-36 Fixed Value RAM - Data Valid Bit Map

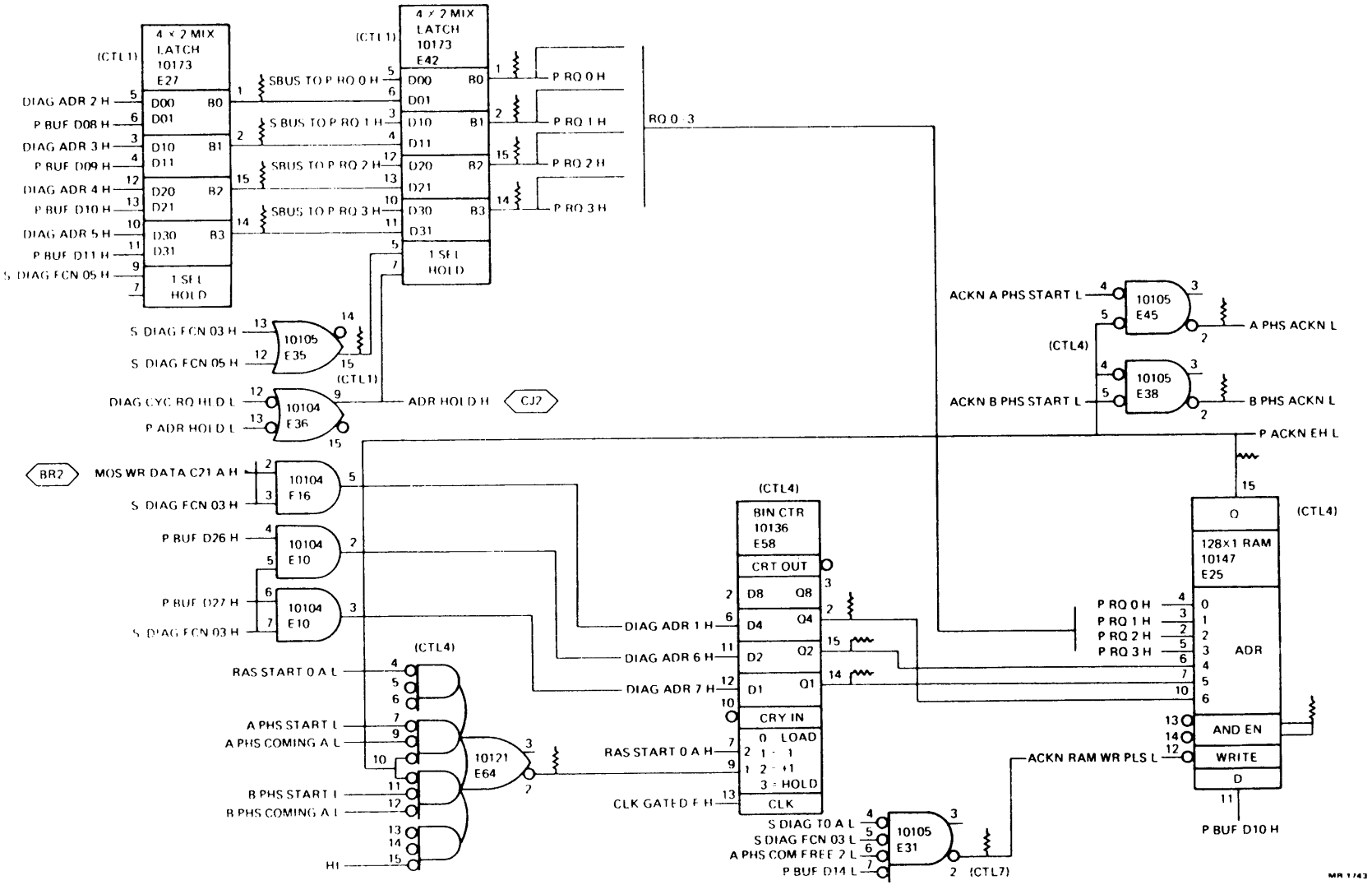


Figure 5-37 ACKN RAM - Logic Block Diagram

1 in the [E] field of the diagnostic function 3. Setting D14 = 0 permits the program to read back and verify the contents of the ACKN RAM. It takes 128 diagnostic function 3 instructions to load the entire RAM.

Once loaded, the ACKN RAM is ready to control XBUS ACKN (phase A or phase B) during a read or write cycle. The XBus specification simply states that ACKN must remain asserted on the XBus for each word requested by the XBUS RQ 0-3 lines. ACKN A or B becomes a level that is asserted shortly after receiving XBUS START A/B and remains asserted n phase A or phase B clock ticks, where n is a function of the RQ 0-3 lines.

RQ 0	1	2	3	n	
0	0	0	0	0	XNU
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	2	
0	1	0	0	1	
0	1	0	1	2	
0	1	1	0	2	
0	1	1	1	3	
1	0	0	0	1	
1	0	0	1	2	
1	0	1	0	2	
1	0	1	1	3	
1	1	0	0	2	
1	1	0	1	3	
1	1	1	0	3	
1	1	1	1	4	

Since n = 4 is a common case (quadword request) it will be used to explain on-line mode. At the start of the memory request the RQ lines are latched into E42 (CTL1) to provide the RAM (CTL4) with the high-order four bits of the address P RQ 0-3. Since n = 4 this is four 1s. The counter E58 (CTL4) has been loaded with 0s to provide three zeros for the low-order address bits (RAS START 0A is negated). At this point the RAM address referenced is 170_8. Translating this address using the rules of Figure 5-36 yields 074_8. From the bit map (Figure 5-36) it can be seen that this location contains a 0 which asserts CTL4 P ACKN EN. This signal:

1. Enables the appropriate A/B PHS ACKN signal to assert the XBUS ACKN A/B signal.

2. Feeds back through E64 (CTL4) to enable CLK GATED F to increment the binary counter E58 if the location read out was a 0 (CTL4 P ACKN EN L is asserted).

The binary counter along with the RQ 0-3 lines generates the address sequence 170, 172, 174, 176, 171 which translates to 74, 75, 76, 77, 174. This sequence contains the bit pattern 0-0-0-0-1

for CTL4 P ACKN EN. At location 176 (translated to 174) the signal is turned off and the counter disabled. The actual algorithm implemented within the RAM specifies that P ACKN EN will be asserted as long as the count in the counter is less than the number of RQ lines asserted. When the sequence terminates, RAS START 0 A will be negated which clears the counter (loads 0) on the next clock.

In summary, the ACKN RAM portion of the fixed value RAMs is used for both read and write cycles. Its sole purpose is to control the timing and width of the XBUS ACKN signals. The program uses diagnostic function 3 signals to initialize the RAM during initial startup.

The remaining portion of the fixed value RAM controls the movement of data between the storage module multiplexer chips and the port data buffer during a read. Besides moving the data in the proper order, the XBUS DATA VALID A/B signals are also controlled by a bit stored in the RAM.

Figure 5-38 shows that the RAM is implemented using three (256 X 1) chips (CTL5).

E11 stores the P RD ADR 35 RAM signal
E5 stores the P RD ADR 34 RAM signal
E4 stores the SET DATA VALID RAM signal

The program will load each of the 256 locations with the patterns shown in Figure 5-39. These patterns define the correct read data move sequences for all permissible combinations of:

1. XBUS RQ 0-3 and
2. XBUS ADR 34-35.

Like the ACKN RAM, these RAMs can best be described by two possible modes - diagnostic load and on-line operation.

To load the RAMs, the software executes a diagnostic function with the address specified in bits D20-27 which controls DIAG ADR 0-7 (CTL7). The data to be written is placed on D11-13 as follows.

P BUF D11	SET DATA VAL RAM
P BUF D12	P RD ADR 34 RAM
P BUF D13	P RD ADR 35 RAM

At S DIAG T0 A time, if D15 = 1, D VAL RAM WR PLS (CTL7) is generated to load the bits into the selected address. Software can read back and verify the RAM pattern by using a diagnostic function with bit D15 = 0. It requires 256 DIAG FCN 03 instructions to properly initialize the RAM. During loading mixers E27 and E42 (CTL1) along with E40 (CTL5) steer the DIAG ADR 0-7 bits to the RAM chip address inputs.

Once initialized these RAM chips are ready to perform the

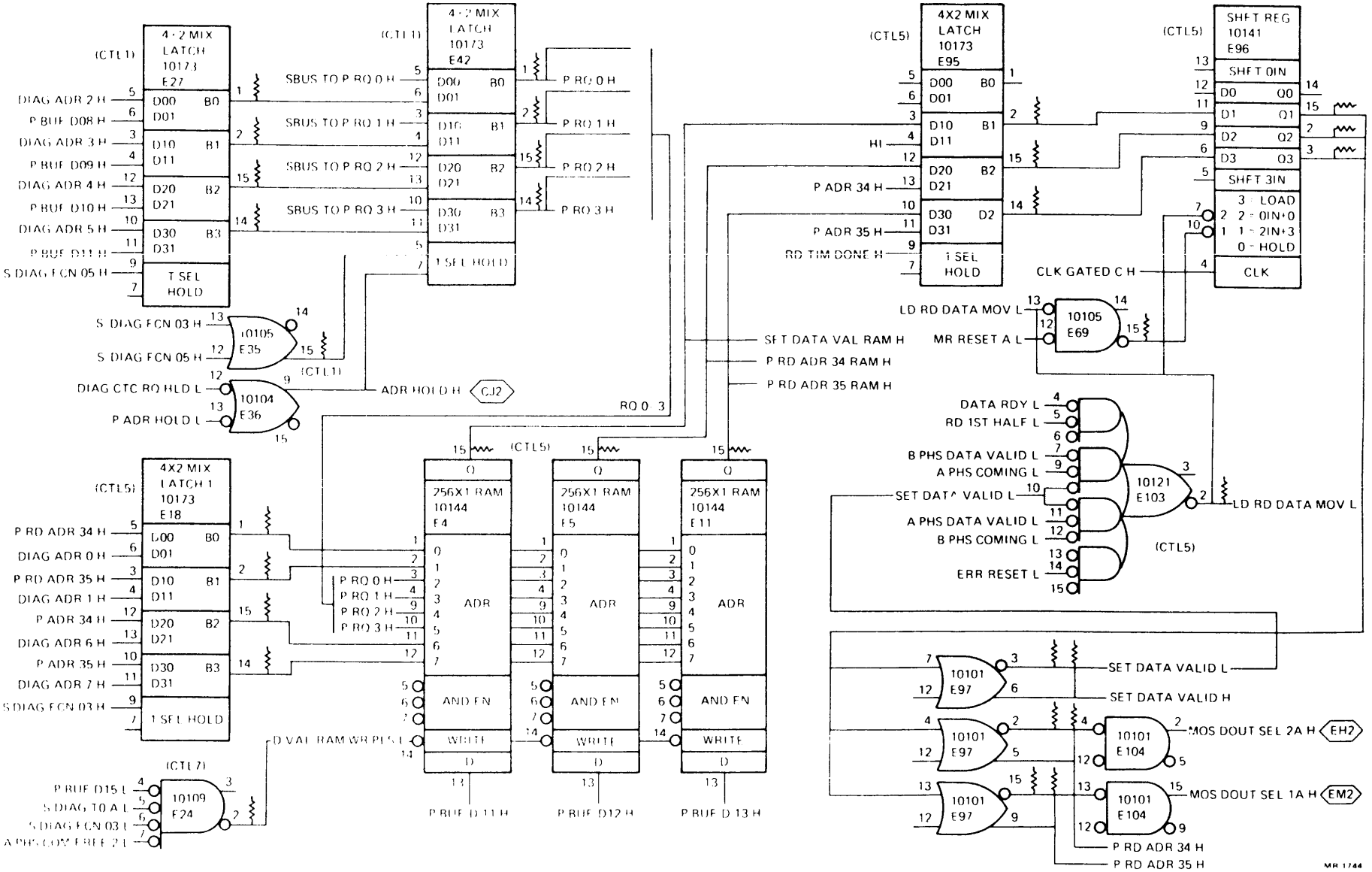


Figure 5-38 Data Valid RAM - Logic Block Diagram

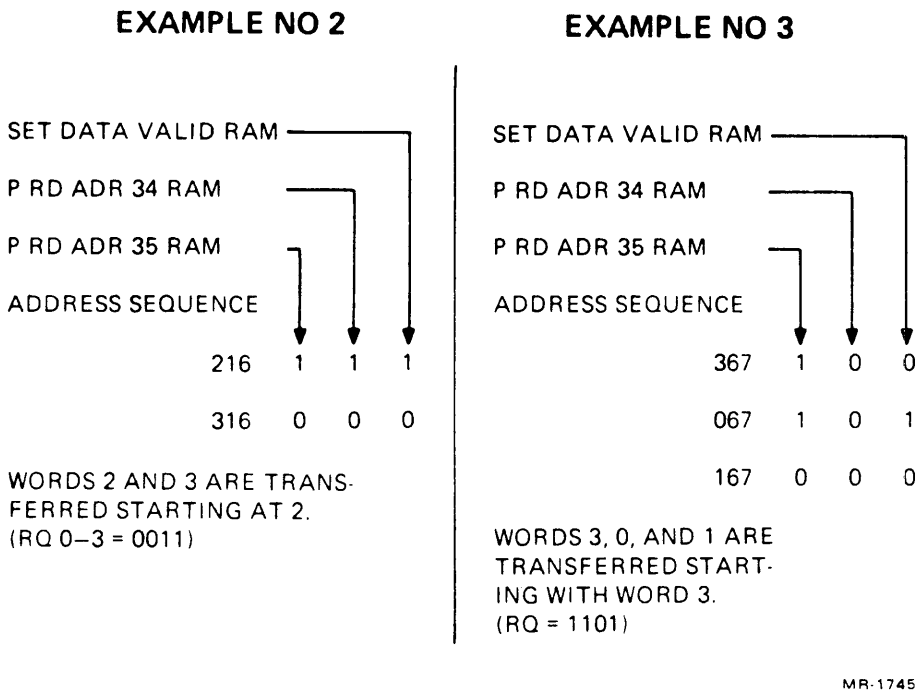


Figure 5-39 Fixed Value RAM - Examples

1. Control the generation of the signals MOS DOUT SEL 2, 1 A to specify the word to be transferred

2. Activate XBUS DATA VALID for each word sent back to the MBox

Like the ACKN RAM the operation is best described by considering some examples.

Example 1 Quadword transfer with starting XBus address on a quadword boundary

This example requires the RQ lines to be all 1s (RQ 0-3 = 1111) and the initial state of P ADR 34-35 = 00. It also implies that four DATA VALID pulses are to be generated and the words in the sequence 0, 1, 2, and 3 are to be moved as follows.

DOUT SEL 2 A	DOUT SEL 1 A
0	0
0	1
1	0
1	1

The initial XBUS ADR 34-35 are latched in E96 (CTL5) along with a HI (DATA VALID) with the signal -RD TIME DONE. When the timing RAM is cycled to access the MOS storage array, the signal DATA RDY will assert (CTL5) LD RD DATA MOV L which permits the next CLK GATED C to transfer these initial signals to E118 (CTL5) where they are used to:

1. Enable generating DATA VALID (set DATA VALID)

2. Set up MOS DOUT SEL 2A-1A to select word 0 and also make P RD ADR 34-35 = 00.

Since the operation takes place in on-line mode, mixer E40 (CTL5) steers P RD ADR 34-35 and P ADR 34-35 to the address inputs of the RAM chips (E99, E54, E26). These four signals (all 0s) combine with P RQ 0-3 (All 1s) to address location 074_8. From the bit map shown in Figure 5-37 this location causes the following RAM outputs.

1 SET DATA VAL RAM H (Asserted)
0 P RD ADR 34 RAM L (Negated)
1 P RD ADR 35 RAM H (Asserted)

These are latched into E96 (CTL5) since the signal, -RD TIM DONE is now negated, and on the next clock (CLK GATED C) are clocked into E118 (CTL5) to generate the following signals.

(CTL5) SET DATA VALID
(CTL5) MOS DOUT SEL 2, A1 = 01 (Select next word WD1)

The RAM address has now been modified by a change in P RD ADR 35 to point to 174_8. Location 174_8 generates the following changes.

1	SET DATA VAL RAM H (Asserted)
1	P RD ADR 34 RAM H (Asserted)
0	P RD ADR 35 RAM H (Negated)

When latched into E96 (CTL5) and clocked into E118 (CTL5), word 2 is selected by MOS DOUT SEL 2, 1A and another DATA VALID enabled. This change in control also modifies the RAM address to point to 274_8. Location 274_8 generates the following changes.

1	SET DATA VAL RAM H (Asserted)
1	P RD ADR 34 RAM H (Asserted)
1	P RD ADR 35 RAM H (Asserted)

These changes are latched into E96 (CTL5), clocked into E118 (CTL5) and used to:

1. Select word 3 for transfer, and
2. Enable the last DATA VALID.

At this point the final RAM address has been modified to access location 374_8 which contains 000_2. This negates all three RAM outputs which prevents generating any further DATA VALID signals; which inhibits the clock from modifying E118 until a new cycle is initiated.

Figure 5-39 describes the RAM sequence for two additional examples. The reader may find it instructive to follow these examples through the logic.

5.8.6.2 Summary - To summarize, the fixed value RAMs provide the control mechanism for generating XBus control signals (ACKN and DATA VALID) and also to control proper movement of the data words between the storage boards and the XBus data bus during a read operation. During initialization the software loads the RAMs with fixed values to generate the control sequence defined by the KL10 architecture.

5.8.7 Personality PROM

5.8.7.1 Overview - Each M8579 storage module contains a PROM that provides a description of that individual board's "personality." The factors considered when determining board personality are listed in Figure 5-40, which also describes the bit format used to encode each characteristic. During initial startup the software reads the contents of the PROMs to establish an association between specific physical hardware and a data base maintained and used by the software. This data base permits the software to gather and log data errors and save this information in a disk file which associates actual module serial number with its specific error statistics over a period of time. Service personnel can print out this file and use the information to isolate and repair faults. The timing information is used to establish MOS read/write timing sequences which may vary from one MOS manufacturer to another. The hardware to support the feature consists of:

1. A 32 X 8 PROM chip on each storage board that is written (first four locations) when the board is manufactured

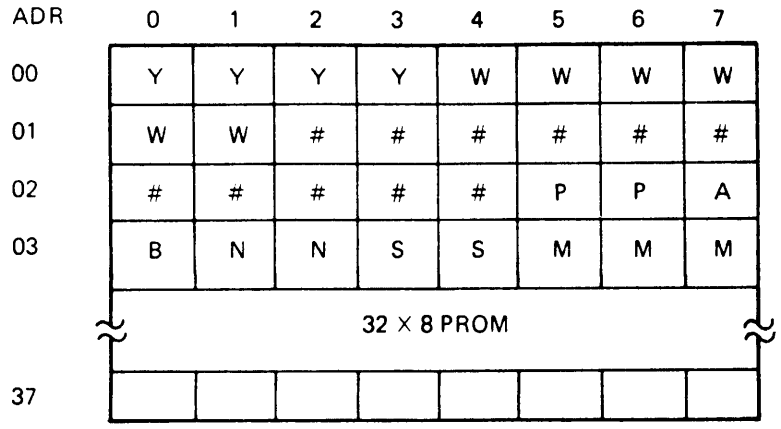
2. Logic to permit software to select and read the first four locations within any PROM.

5.8.7.2 Functional Block Diagram (Figure 5-41) - Each M8579 board (SM14) uses a single 82S123 32 X 8 PROM chip. To enable reading the information from the PROM the software executes a diagnostic function 2 to generate the following signals (CTL9).

MOS SM SEL GP 0-3	Selects which group
MOS SM SEL FLD 00-03	Selects which field (M8579)
MOS SM PROM A 0-1	Selects one of the first four PROM addresses

These signals are generated by decoding P BUF D11-14, with two 10162 chips on (CTL9) to generate single group- and field-select signals which are sent to the storage boards. The 2-bit address signals A0 and A1 are a direct function of P BUF D 13-14 as coded by the programmer.

Once enabled, the selected PROM (SM14) sends back 8 bits, MOS SM PROM DATA 0-7, the contents of the location specified. This data is received by the M8576 (CTL9) and sent to eight 10164 mixers on (CTL7) where they generate CHK TO P D 07-14. From here, the PROM data is sent to the port buffer diagnostic function (M8574) via the (M8575). Once in the port buffer it can be gated onto the XBus data lines. To retrieve all the PROM data, the software must execute four diagnostic function 2s for each storage board. During each PROM read the software must perform the parity check since there is no hardware provided for this function.



YYYY — 4 BIT YEAR NUMBER, BCD EQUIVALENT OF THE LAST DIGIT OF THE CALENDAR YEAR.

WWWWWW — 6 BIT WEEK NUMBER

...# — 11 BIT SERIAL NUMBER

PP — MOS STORAGE BOARD POPULATION CODE

- 00 BLOCK 0 ONLY EXISTS
- 01 BLOCK 0 AND 1 EXIST
- 10 BLOCK 0, 1, AND 2 EXIST
- 11 FULLY POPULATED

A — PARITY – CHOSEN TO GENERATE EVEN PARITY FOR LOCATION 0, 1, AND 2 IN TANDEM.

B — ODD PARITY FOR LOCATION 3.

NN — TWO BIT TIMING NUMBER – REFER TO SUB-SECTION - FOR DETAILS.

SS — MOS CHIP SIZE:

11	4K
10	16K
01	32K
00	64K

MMM — 3 BIT MOS CHIP MANFACTURER CODE.

000	MOSTEK	100	MOTOROLA
001	INTEL	101	T. I.
010	FUJITSU	110	OTHER (DEC ETC.)
011	HITACHI	111	RSVD

MR-1746

Figure 5-40 Personality PROM - Bit Format

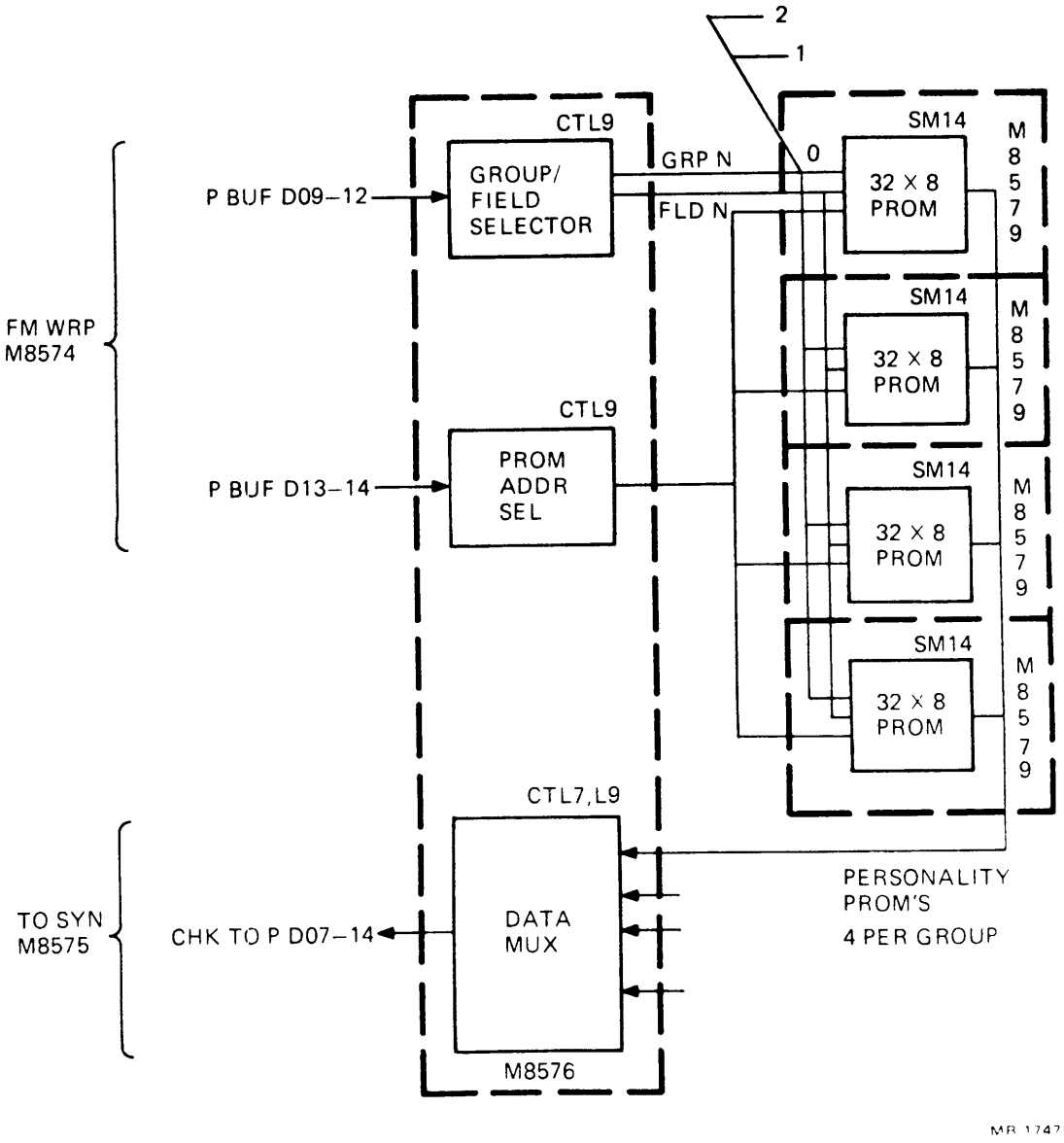


Figure 5-41 Personality PROM - Functional Block Diagram

5.8.8 Refresh Control Logic

The MOS chips used in the storage arrays are dynamic MOS storage elements that require periodic recharge to prevent loss of stored information. This process is referred to as refresh, and the time interval between successive recharges is a function of the chip manufacturing process. The time, called the refresh interval, is 2 ms for the 16K chips used in the MF20. To completely refresh all cells within a 16K chip requires that all 128 possible row addresses be strobed during the 2 ms period. Most of the refresh control logic is contained within the M8577 (ADT) module. It is described in Paragraphs 5.8.8.1 through 5.8.8.3.

5.8.8.1 Refresh Functional Block (Figure 5-42) - To achieve the objectives of the refresh function the following signals to the M8579 (SM) modules must be activated at the appropriate times.

MOS GP 0-2 SELECT - select all groups within the MF20 to receive the row address and strobe

MOS ADR 0-7 - select one of 128 possible row addresses

MOS RAS 0-3 - strobe all cells in the selected row to refresh

Since these signals are also used to store and retrieve information from the MOS array, the refresh control logic must perform its function between normal read/write accesses. The refresh control logic is partitioned into the following functional areas.

1. Refresh Control and Timing (ADT3, 5) - Clock-driven combinational logic network that generates a logically sequenced set of timing and control signals that perform the following functions.

   a. Enable decrementing the refresh interval counter

   b. Enable incrementing the MOS address counter

   c. Trigger a timing RAM sequence to generate row address strobes (RAS)

   d. Enable all <u>groups</u> within a single MF20 to perform a refresh

2. Timing RAM (ADT4) - Generates the proper RAS timing when triggered by the refresh control logic. The timing sequences generated are the same as those for a normal read/write operation as described in Paragraph 5.8.5.

3. RAS Mixer (ADT6) - Activates the appropriate row address strobes under control of the timing RAM and the refresh control and timing logic.

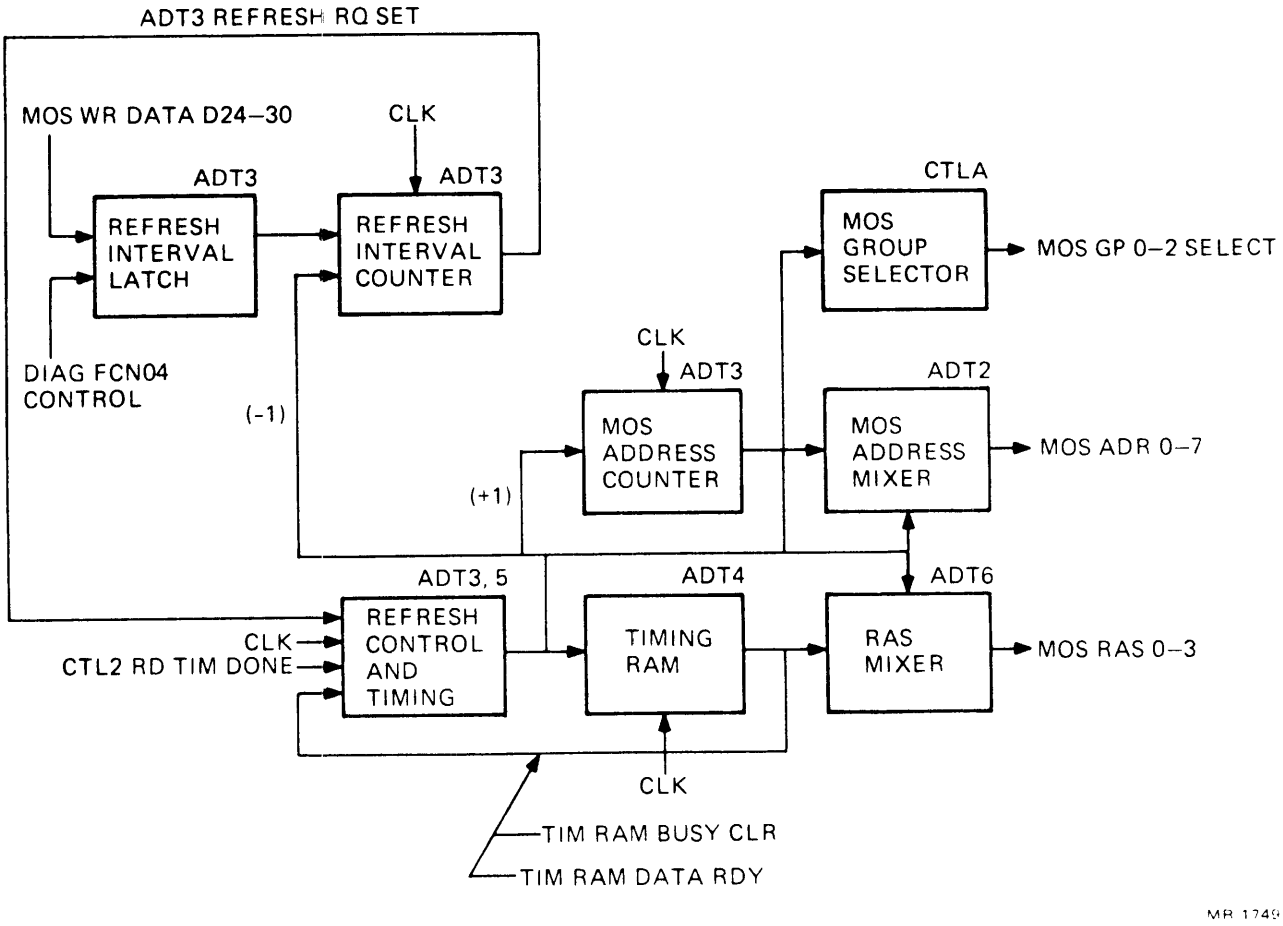


Figure 5-42 Refresh Control Logic Functional Block Diagram

4. MOS Address Counter (ADT3) - Standard 8-bit up-counter that generates the row address sequence 000-127 as directed by the refresh control and timing logic.

5. MOS Address Mixer (ADT2) - Permits the MOS address counter to control the MOS ADR 0-7 lines to the storage modules when selected by the refresh control and timing logic.

6. MOS Group Decoder (CTLA) - Generates the three group-select signals, MOS GP 0-2 SELECT, when activated during a refresh cycle.

7. Refresh Interval Latch (ADT3) - A 7-bit latch that can be loaded by the program via a diagnostic function 4 to specify the time period between successive refresh cycles. The number loaded is a function of the MOS chip specification and the basic clock frequency. For a chip specification that requires 128 refresh cycles every 2 ms, the number is 29 for a base clock rate of 30 MHz.

8. Refresh Interval Counter (ADT3) - Standard binary down-counter that is preloaded each refresh cycle from the refresh interval latches.

5.8.8.2 **Refresh Operation** - During initial startup the software must load the refresh interval latch and enable the refresh control and timing logic. This is achieved by executing a diagnostic function 4. (Refer to Table 5-12.) Once enabled the refresh interval counter begins down-counting toward 000. On the next clock tick the count goes negative and generates the signal ADT3 REFRESH RQ SET which initiates a refresh cycle. Each refresh cycle will refresh 128 cells (a row) in all MOS chips in the storage array.

To accomplish the refresh the refresh control and timing logic performs these functions.

1. Increments the MOS address counter to select the next row to be refreshed

2. Enables all M8579 (SM) modules to accept the refresh signals

3. Switches the MOS address mixer to activate MOS ADR 0-7 (row address lines)

4. Initiates a timing RAM sequence to activate the MOS RAS 0-3 signals to pulse the selected row

5. Finally reloads the refresh interval counter to reestablish the proper interval

Figure 5-43 shows a detailed description of the refresh timing sequence. Figure 5-44 is a timing diagram that describes the timing sequence for a refresh cycle hidden behind a read cycle.

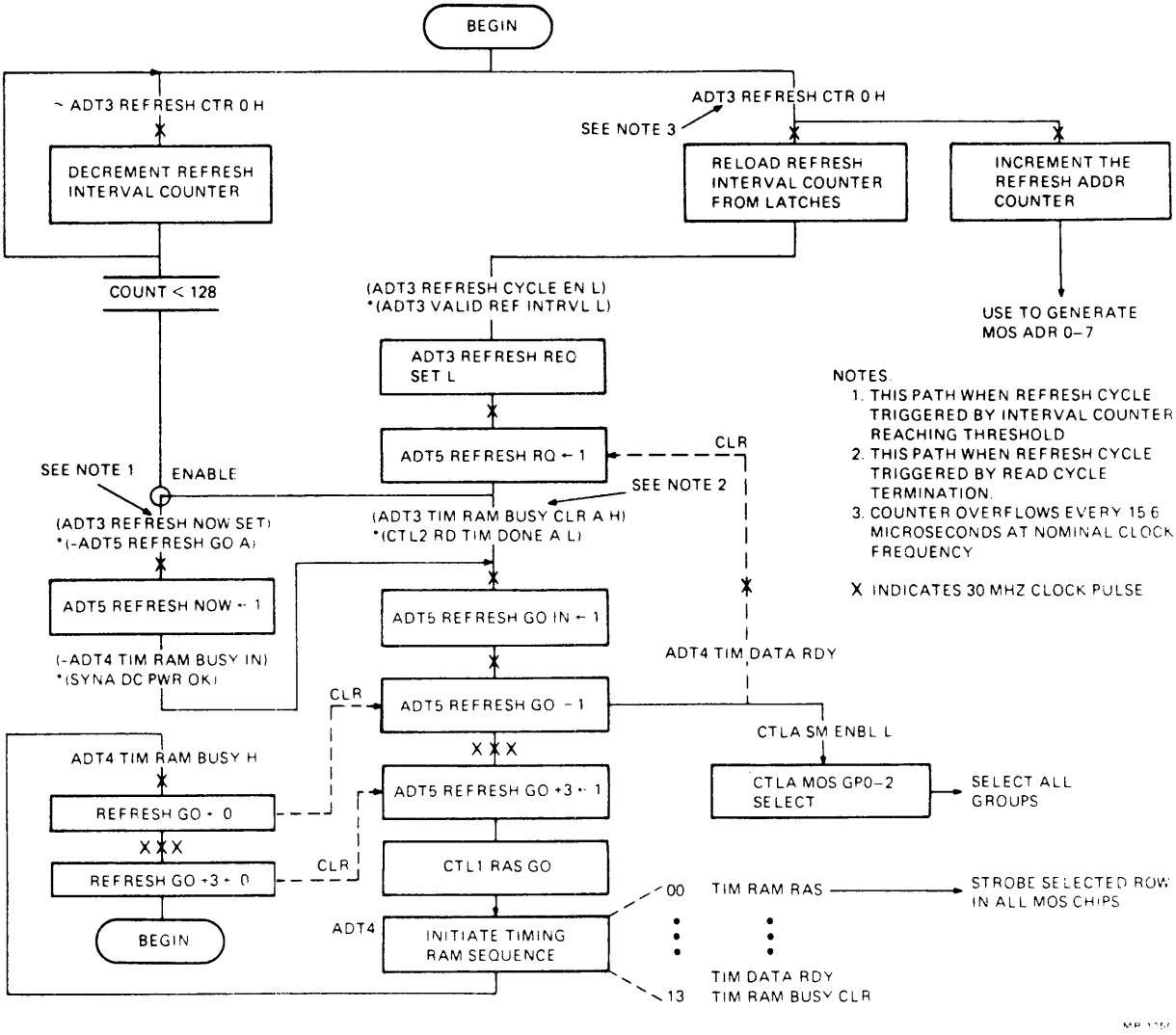


Figure 5-43 Refresh Timing Flowchart

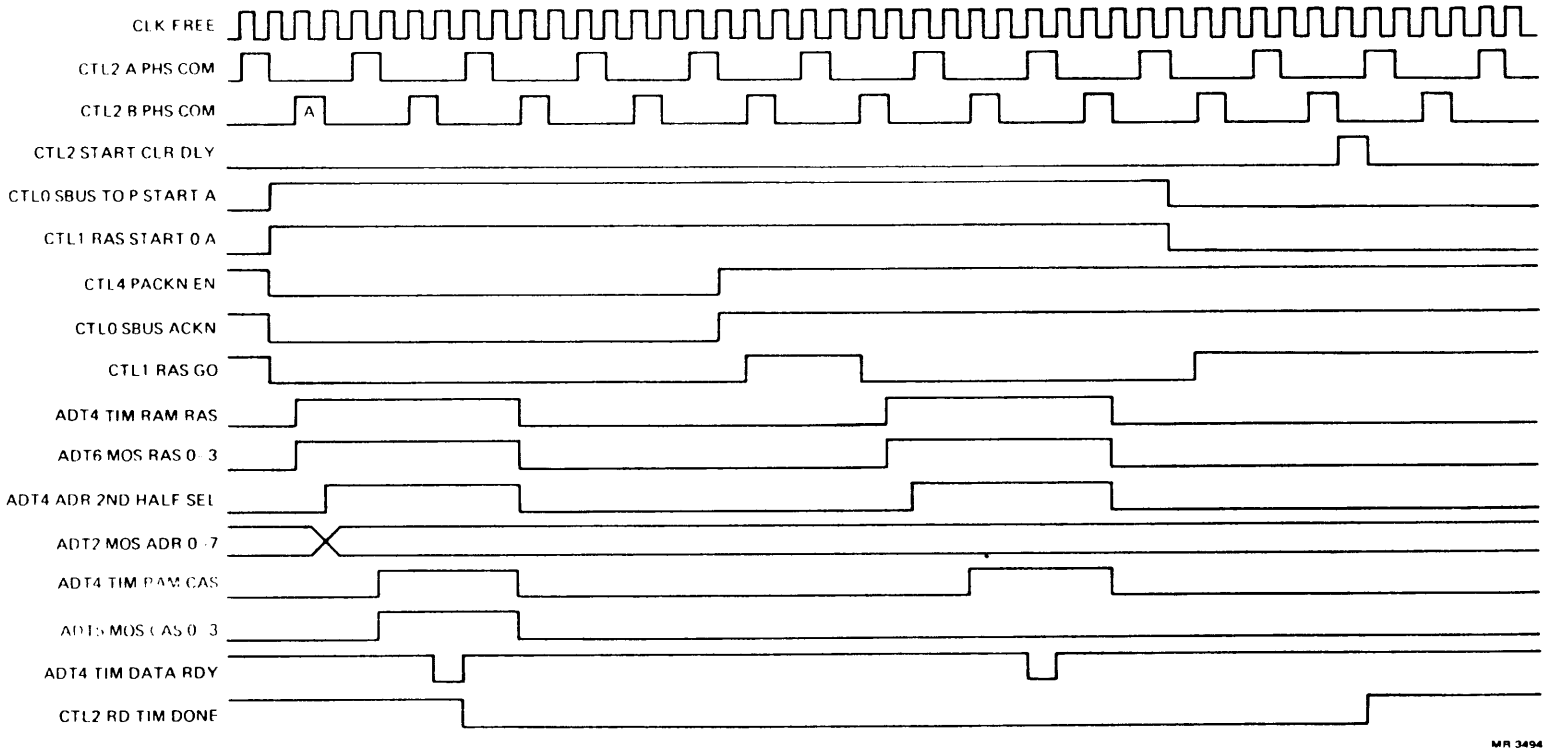


Figure 5-44 Read Followed By Refresh Timing (Sheet 1 of 2)

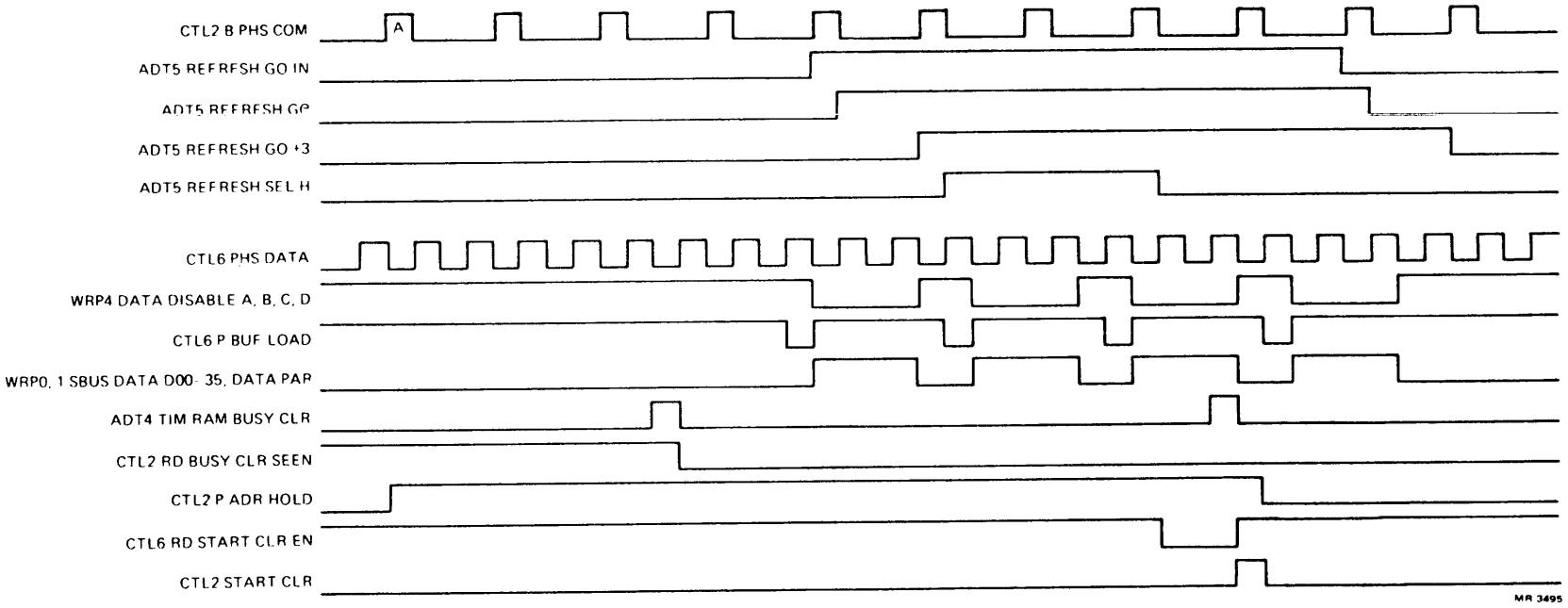


Figure 5-44 Read Followed by Refresh Timing (Sheet 2 of 2)

5.8.8.3 Summary - The MF20 uses dynamic MOS chips that must be refreshed periodically to prevent loss of stored information. This requirement creates a need for the refresh control logic on the ADT module. Once initialized by the software the refresh logic is automatic and transparent to the user. Hiding many of the refresh cycles behind normal memory read operations minimizes the overhead required to perform the refresh function.

5.8.9 Error-Handling Logic

If a hardware failure causes the MF20 to respond in a manner contrary to its design specification, the information stored or retrieved by the software is suspect, and if used for subsequent program operations it could result in a system failure. The MF20 includes a hardware mechanism capable of detecting faults and signaling the software so that it may take the appropriate action to minimize the impact of the fault on system operation. The following paragraphs describe the error-handling logic implemented in the MF20. This logic is described in the context of:

1. The types of errors that can be detected and reported
2. The kind of information that can be retrieved by the software to determine the cause of the error
3. Possible action to minimize the impact of the errors.

5.8.9.1 Error Classification - MF20 errors can be classified into the following categories.

1. Parity errors
2. Data errors
3. Control errors

Parity errors may be either address parity errors or data parity errors. Data parity errors may be further classified as to whether they occur during a read or a write. Figures 5-45, 5-46, and 5-47 describe the transmission paths between parity generator and checker for both data and address parity errors. Data errors occur only during a read operation and indicate that the information received from the MOS storage array is suspect. These errors may be correctable (SBE) or uncorrectable (DBE). Finally, there are several control errors associated with MF20 control logic faults. The following control errors can be flagged by the error-handling logic.

1. Control RAM parity errors
    a. Timing RAM
    b. Address response RAM
    c. Spare bit substitution RAM

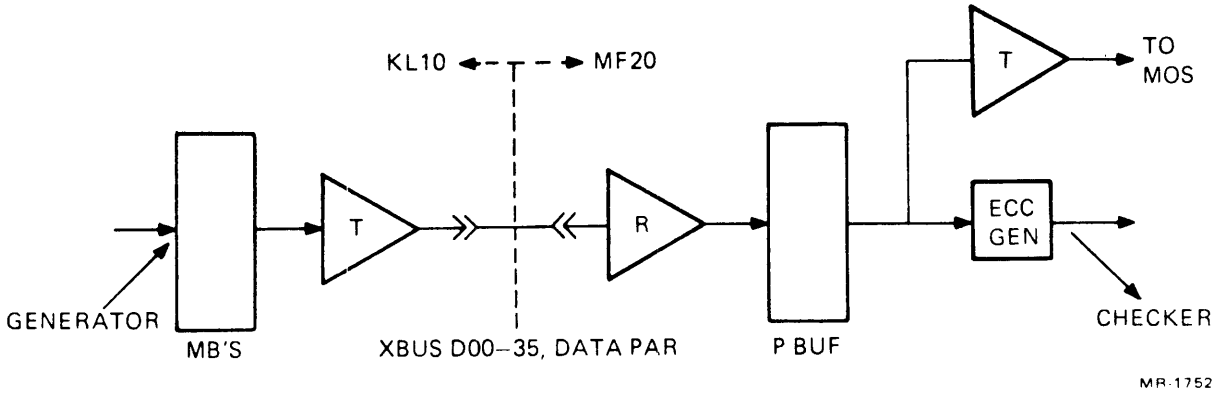


Figure 5-45 Data Parity - Write Transmission Path

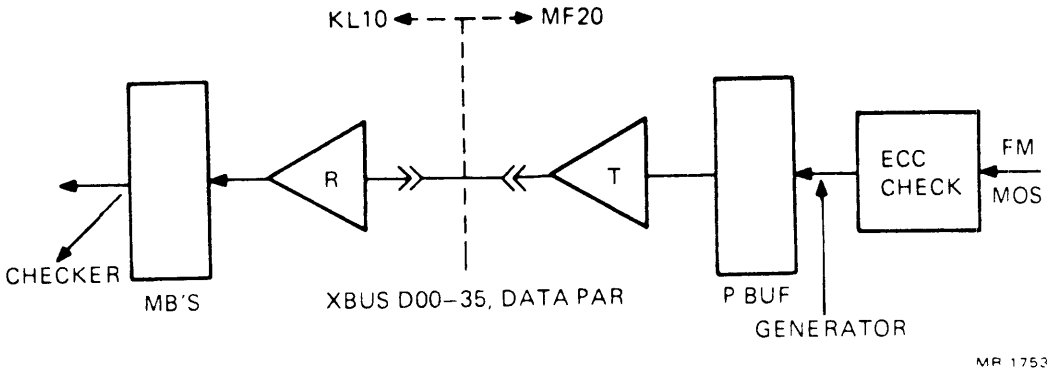


Figure 5-46 Data Parity - Read Transmission Path

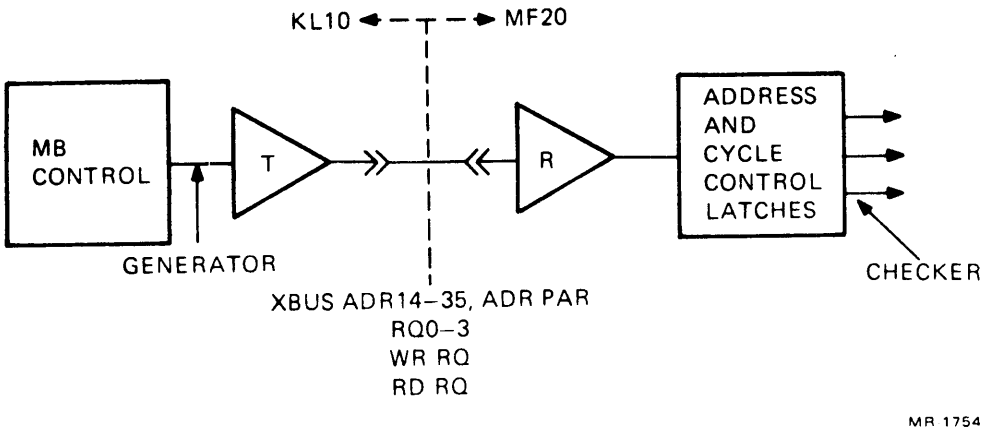


Figure 5-47 Address Parity - Transmission Path

2. Incomplete request error

3. NXM error

4. DC power bad

Whenever a control RAM is accessed during a read or write data operation, its output is checked for proper parity. If incorrect parity is detected, the appropriate error flag is set to cause an XBus error. Incomplete request error occurs if the timing RAM fails to complete its sequence within 4.2 ~s. An NXM is an MBox-detected error that indicates a failure of the MF20 to respond to XBus START by asserting XBus ACKN. This could be caused by improperly configuring the address response RAM.

DC power bad indicates a fault in the MF20 power supply. This error condition inhibits the MF20 from asserting XBus ACKN which forces an NXM error if the MBox attempted a read or write operation.

5.8.9.2 Error Information Retrieval - All MF20-detected errors cause XBus error to be asserted which, if enabled, causes a programmed interrupt back in the EBox. One exception to this is that correctable SBEs can be inhibited from setting the error flag if the RCE bit (not report correctable error) is set in the spare bit RAM. Once the program has been interrupted by an XBus error it can use the appropriate diagnostic functions to retrieve the error information necessary to log and analyze the cause of the error. A diagnostic function 0 is executed to retrieve the following information.

1. Was it a control or parity error?

2. Was it a correctable read error?

3. If a parity error, was it read, write or address?

4. Was it an incomplete cycle?

5. If a read or write error, which words were being requested out of the quadword group?

6. Was it a read or write request?

7. What XBus address was being accessed?

Following the function 0 a diagnostic function 2 is executed to determine the following information.

1. If a multiword request, which word caused the error?

2. If it was a control RAM error was it the timing RAM, spare bit RAM, or address response RAM?

A diagnostic function 6 (subfunction 1) can be executed to retrieve the state of the double-bit error signal, and a diagnostic function 10 to retrieve the state of the dc power inadequate flag. Refer to Chapter 4 for more detailed descriptions of the diagnostic functions and MF20 programming.

5.8.9.3 Error Logic Analysis - The ADT0 print shows the two XBus error signals being generated. Both signals, SBUS [N] ERROR and SBUS [N] ADR PAR ERR are enabled by ADT7 ERR TO BUS and controlled by the state of the signal, ADT7 ADR ERR. All the error flags and control logic required to assert ERR TO BUS are shown on the ADT7 print.

The key signal needed to activate the XBus error signals is ADT7 ERR HOLD. This signal looks back at all the possible error conditions detectable by the MF20 and is asserted if any of the following errors are detected.

1. ADT7 BLK RAM ERR - Asserted if parity error is detected at the output of the address response RAM

2. ADT7 TIM RAM ERR - Asserted if parity error is detected at the output of the timing RAM

3. ADT7 SUB RAM ERR - Asserted if parity error is detected at the output of the spare bit RAM

4. ADT7 INC ERR - Asserted if timing RAM fails to complete its sequence within 4.5 ~s timout

5. ADT7 ADR ERR - Asserted if address parity error is detected during a read or write

6. ADT7 RD ERR - Asserted during a read if an ECC error is detected - either correctable (SBE) or uncorrectable (DBE)

7. ADT7 WR PAR ERR - Asserted if data parity error is detected during a write operation

Once the ERR HOLD signal is asserted, all pertinent error analysis information is latched and no subsequent error condition can be detected until the original error condition is cleared. After retrieving all the necessary error information the software must execute a diagnostic function 0 with the CLR ERR bit set (bit 05) to clear out the original error condition and leave the error logic free so that subsequent errors may be detected. Starting at the ADT7 print it is a simple task to trace back the error signals to locate the actual error detection logic that is spread across the four control boards. Figure 5-48 summarizes this task.

5.8.9.4 Error Recovery and Analysis - During typical system operation the MOS memory resource will be segmented and allocated to several different programs - the monitor and the user jobs. If

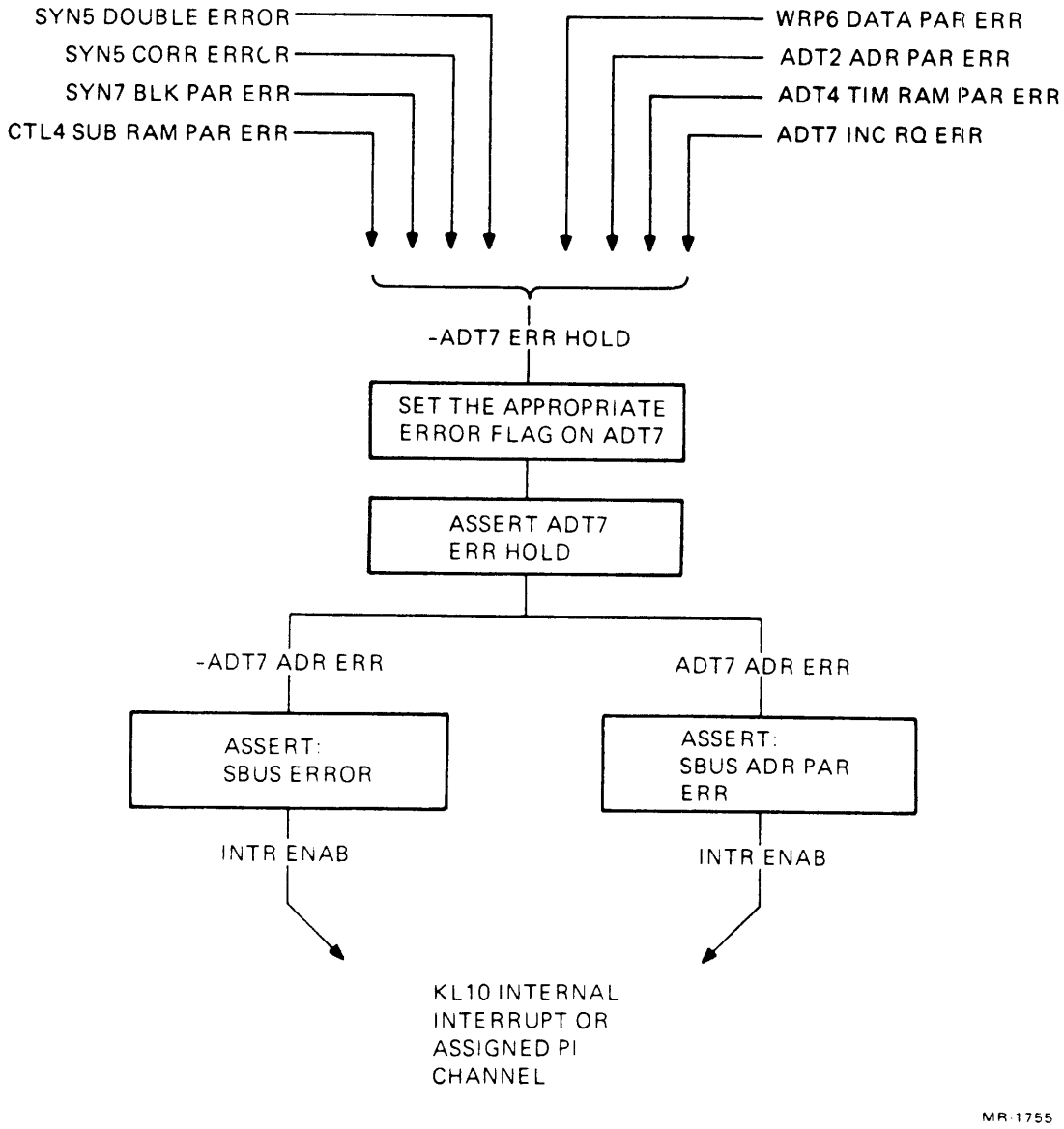


Figure 5-48 Error Handling Logic - Simplified Flowchart

a fault is uncorrectable and local to the segment of MOS allocated to the monitor, it will probably cause a system failure which will require operator intervention.

The majority of faults, however, will probably occur outside of the monitor's executable code space. If the faults are local to a particular user program, the monitor will take appropriate action to localize each fault and patch out the failed area before it can affect the monitor and cause a system crash. If the monitor is intact, it can call the appropriate subprogram to retrieve and log the MF20's error status for subsequent analysis by the service engineer. It can also reschedule the user that caused the error at some convenient restart point to minimize the impact of the memory fault. Chapter 7 includes a more detailed discussion of MF20 fault analysis and the use of the diagnostics to make the required repairs.

5.9 KW20 MASTER OSCILLATOR

This paragraph describes the KW20 master oscillator which is the clock source for all timing, both CPU and memory, in an MF20 MOS Memory System. It includes both physical and electrical descriptions and focuses on how the clocks are generated and distributed throughout the system.

5.9.1 Physical Description (Figure 5-49)

The KW20 master oscillator is a small box mounted in the upper front section of the I/O bay (on the right side as viewed from the rear). Two cable harnesses and five black coaxial cables enter and leave the box. The coaxial cables are used to distribute master clock signals to the CPU backplane and up to four MF20 backplanes. One cable harness connects to the M8572 module in the CPU and is used to carry two clock-select signals to the KW20. The other cable harness carries the dc power required to operate the master oscillator from each of up to four MF20 power supplies and one H770 power regulator.

5.9.2 KW20 Functional Block Diagram (Figure 5-50)

The KW20 consists of a set of three crystal-controlled oscillators and a regulated power supply. Basic clock signals of 30, 31,and 25 MHz are available at the input to a 4-line to 1-line multiplexer. The signals are generated by three crystal oscillators operating at frequencies of 60, 62, and 50 MHz with the oscillator outputs used to toggle three frequency dividers. A fourth input to the multiplexer provides the means of selecting an external timing source.

Two control signals, generated by the CPU, allow program control of the clock source and frequency via an EBus diagnostic function. These signals, CP CLK SEL 2 and CP CLK SEL 1, operate as follows.

SEL 2	SEL 1	CR	
0	0	0	EXT SOURCE
0	1	1	31 MHz
1	0	2	25 MHz
1	1	3	30 MHz (nominal)

The following console command (KLDCP) is used to control the clock.

>.FW 72/NUM NUM = 0-3

To set normal operating speed the operator would type the following.

FW72/3 <CR>

After selecting the frequency the following command would be typed to select the KL10's external source of timing.

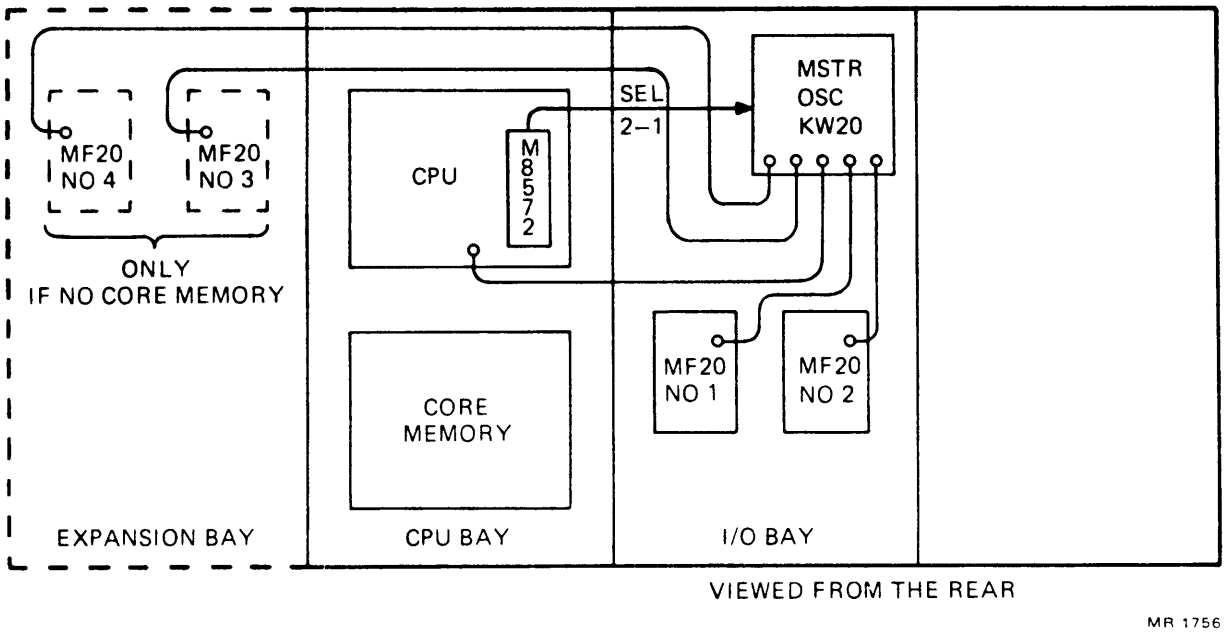


Figure 5-49 MF20 System Block Diagram

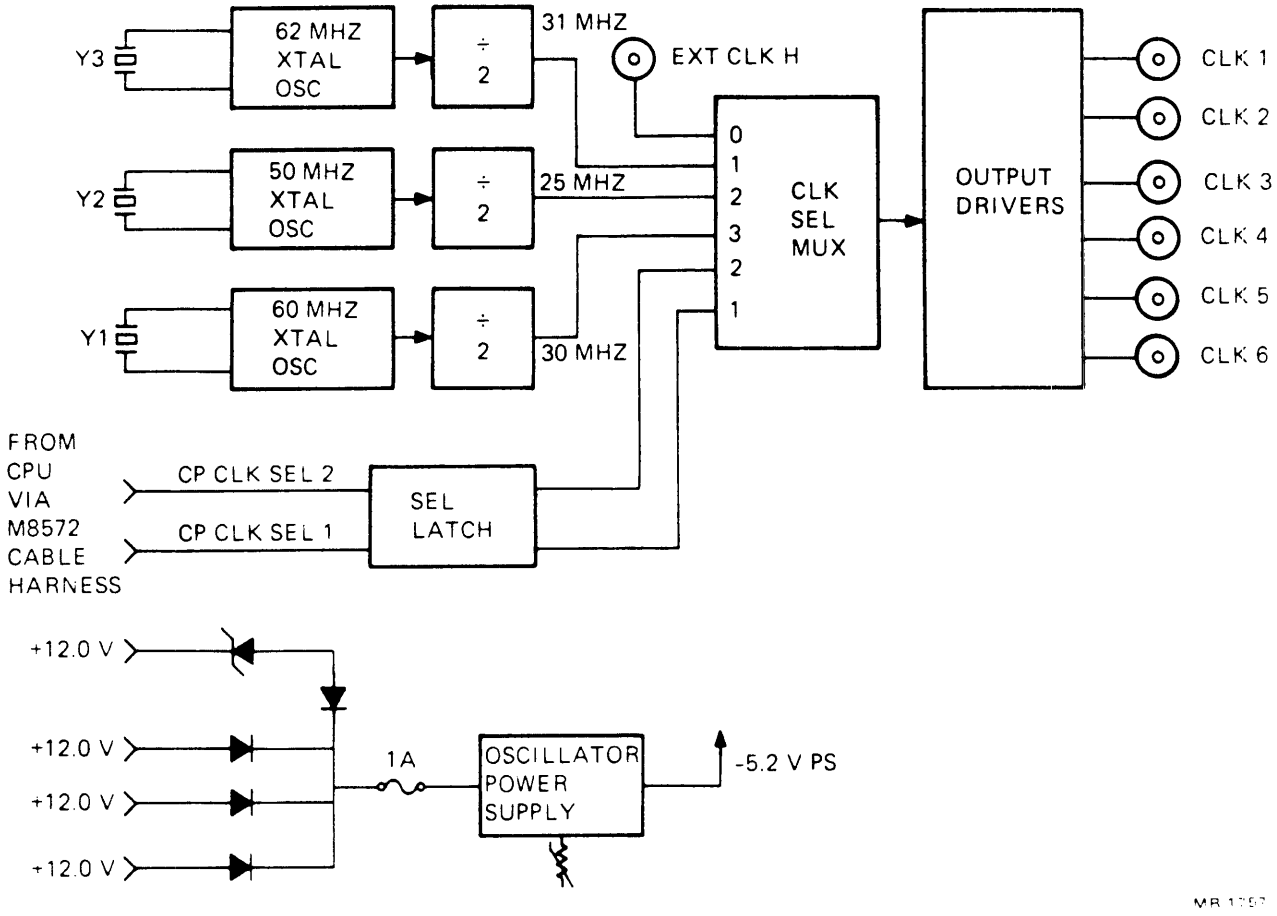


Figure 5-50 KW20 Functional Block Diagram

The power supply is driven by the +12 V supplies in the MF20s and the +15 V supply in the H770. These voltages are fed through isolating diodes and a 1 A fuse to a precision regulator. With more than one source, the KW20 can continue to operate in the event one of the MF20 supplies fails. The supply's output is adjusted to provide a precision-regulated -5.2 V to operate the master oscillator circuits. Use the procedure described in Paragraph to check and adjust the KW20.

After the CPU has established the desired clock-select signals the output of the clock select multiplexer is applied to the output drivers that distribute the clocks to the rest of the system via coaxial cables.

5.9.3 MF20 Clock Signal Distribution (Figure 5-51)

All logic signal events occurring within the MF20 are synchronized by the master clock from the KW20. This clock connects to the MF20 at the backplane via a coaxial cable. From here, it fans out to generate 105 separate clocks distributed to up to 16 logic modules. Entering the M8577 module (ADT6), the single KW20 clock is gated and distributed as four free and fourteen gated clocks designated as follows.

ADT6 CLK FREE 00-03
ADT6 CLK GATED 00-13

The difference between the two types of clocks is that the gated clocks can be turned on and off under control of the program. This feature is used by the diagnostic program to execute single-step operations for fault isolation.

Each of the control modules (M8574, 75, 76, 77) receives a free clock and distributes it to the internal logic on the board. The M8576 and M8577 also receive gated clocks which are distributed by logic on those boards. The remaining twelve gated clocks (00-11) are distributed to each of the M8579 storage modules. Each module that receives either a gated or free clock is deskewed at the factory using a standard precision test setup. This ensures timing compatibility between all boards.

5.9.4 Synchronization (Figures 5-52 and 5-53)

As previously discussed, the MF20 and the CPU are driven by the same master oscillator. Due to differences in logic circuit parameters and cable delays, steps must be taken to ensure that the MF20 is in proper synchronism with the MBox. This is achieved by two adjustments on the M8576 module (CTL2).

The MBox transmits a signal, XBUS CLK INT, which is synchronized with the A and B phase clocks in the CPU. This signal is fed through two adjustable delays on the M8576 and used to generate A and B phase clocks in the MF20. By proper adjustments of the

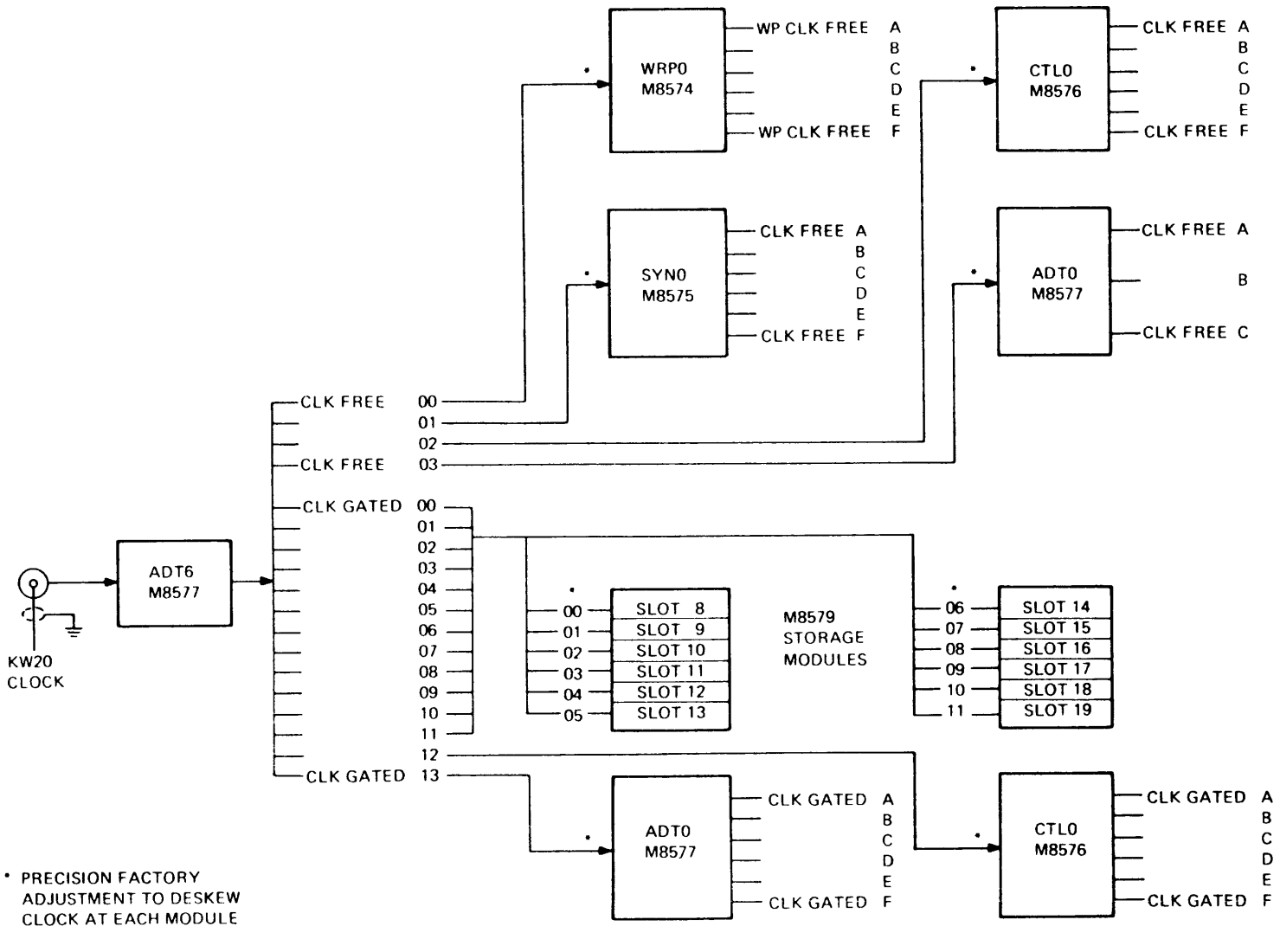


Figure 5-51 MF20 Clock Signal Distribution

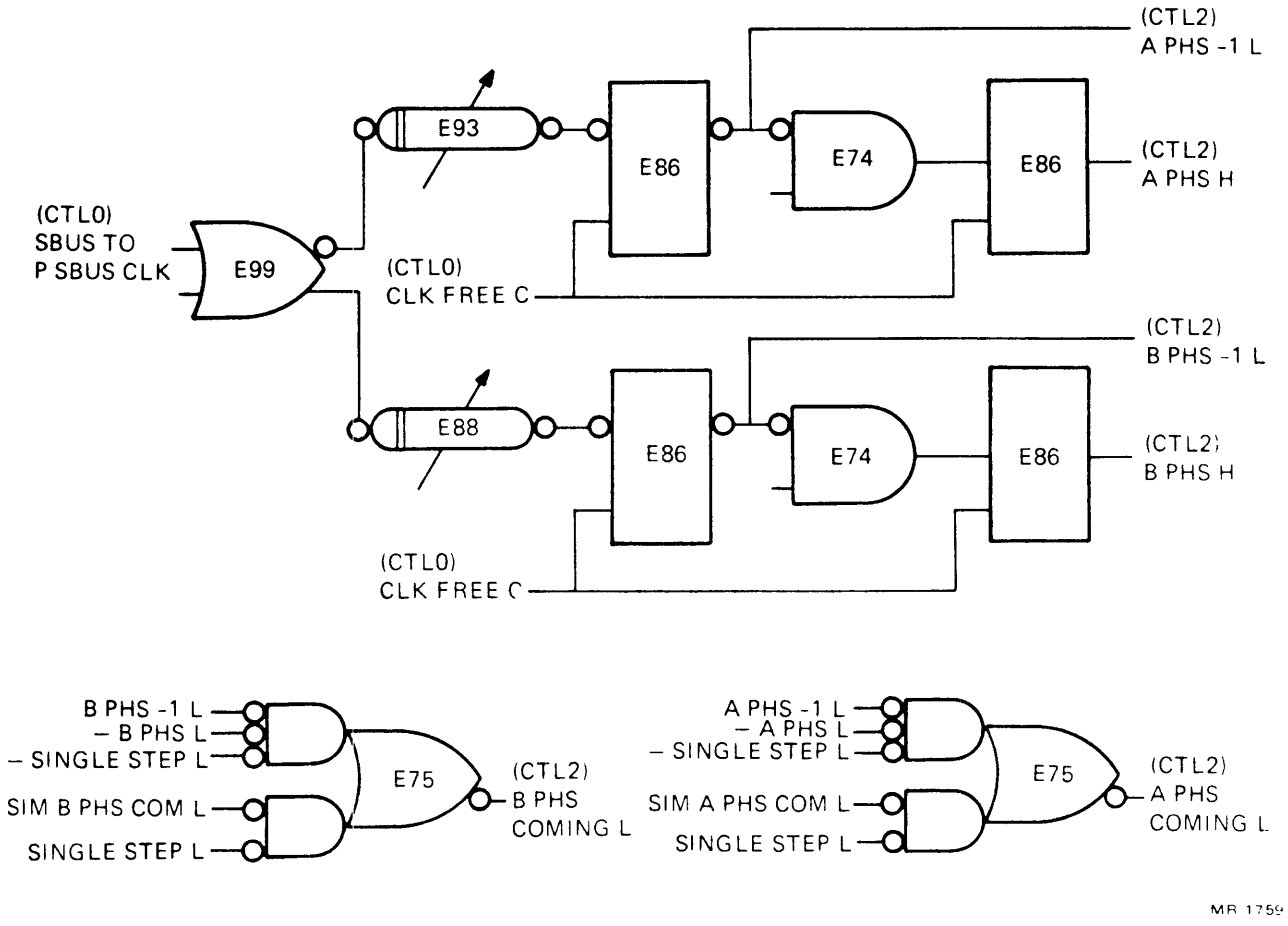


Figure 5-52 Basic MF20 Timing Logic

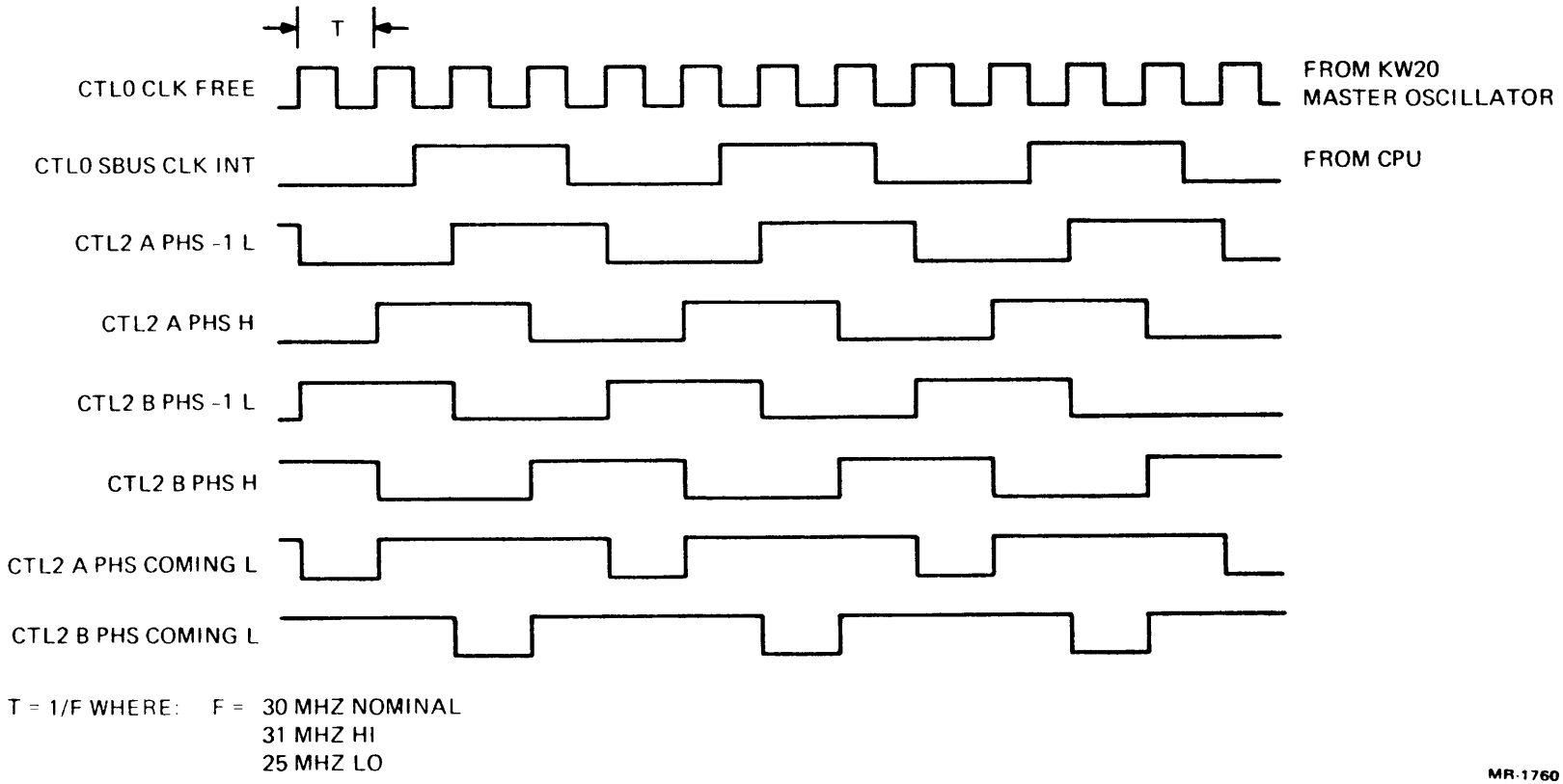


Figure 5-53 Basic MF20 Timing Waveforms

delays, the phase clocks in the MF20 can be synchronized with the corresponding clocks in the MBox. Refer to Paragraph 7.4.2.2 for a description of this procedure.

Once synchronized, the signals CTL2 A/B PHS and CTL2 A/B PHS COMING and their derivatives are used to time all XBus dialogue between the MF20 and MBox.

5.10 H7131 POWER SUPPLY

Each MF20 logic assembly requires a separate power supply unit to provide the necessary ac and dc operating voltages. In the basic DECSYSTEM-20, the MF20 power supplies are mounted in the bottom rear of the CPU cabinet. This paragraph describes the general operating characteristics of the MF20 power supply. Since the current maintenance philosophy dictates that the entire power supply unit be treated as a field replaceable unit, detailed circuit descriptions are not included. Chapter 7 describes the required test procedures and includes detailed removal and replacement procedures for both the power supply and battery pack.

5.10.1 Input/Output Specifications (Figure 5-54)

The power supply receives 115 Vac, single-phase, from the CPU and converts it to the following regulated dc voltages.

-2 V ± 1.5% @ 25 A
+5 V ± 1.5% @ 35 A
+12 V ± 1.5% @ 11 A
-5.2 V ± 1.5% @ 60 A

The +12 V is used by the MOS chips or the M8579 modules and also by the KW20 master oscillator. The M8579s also use the +5 V output. To supply the circuits on the remaining four modules (M8574, M8575, M8576, and M8577) the supply distributes +5 V, -5.2 V, and -2 V.

Regulation is achieved by sensing the amplitude of each of the four supplies at the backplane. Each voltage at the backplane connects back to a terminal board at the top of the power supply panel as follows.

TB3	-2 V sense
TB4	+5 V sense
TB5	+12 V sense
TB6	-5.2 V sense

The sense cable harness consists of a set of four shielded coaxial cables. When the voltage at the backplane changes, the sense cable transmits the change to the regulator circuits within the power supply. These circuits respond by modifying the output voltage to correct the sensed change.

Four terminal blocks, each with four brass connector posts, are used to connect the dc outputs to the MF20 via a 28-wire cable harness.

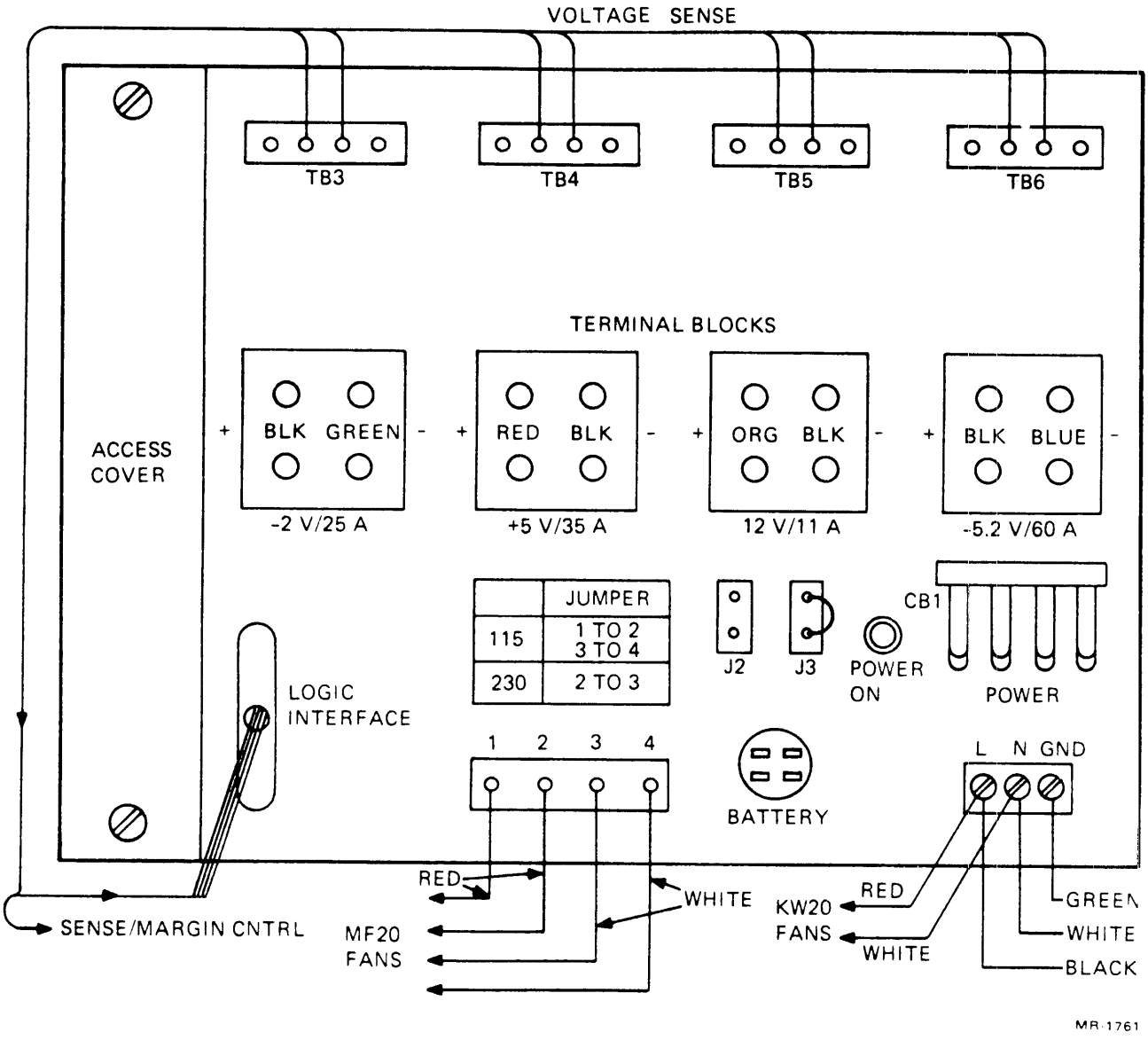


Figure 5-54 MF20 Power Supply Front Panel

CAUTION

A special torque wrench is required to tighten down the nuts on the brass connector posts to 20 in-lbs. Serious damage can result to the brass posts if this torque is exceeded.

A logic interface connector receives control signals from the MF20 and sends status back to the MF20 backplane. The controls consist of a logic signal to reset the dc low condition latched by the power supply and a set of eight voltage margin signals that permit the program to margin the four power supply voltages via a diagnostic function 10. The single status signal is activated by the MF20 when the ac power has dropped below a specified minimum level for greater than 30 seconds. Once detected this condition is latched and must be cleared by the program. When the program clears this condition [a status flip-flop in the MF20 (M8575 - SYNA)], the LED on the MF20 backplane lights, indicating that the software has responded to the power supply status change.

5.10.2 **Power Distribution** (Figures 5-54, 5-55)
The ac power connects into the supply via a standard 3-connector power cable to a terminal board in the lower right corner of the unit. A main circuit breaker, CB1, is used to turn the ac power ON and OFF. When ac current is applied to the regulator, a POWER ON indicator illuminates. Two jacks J2 and J3 allow connection of a remote turn-on control cable. Three pair of control wires (red/white) distribute the 115 Vac to three cooling fans - two on the bottom of the logic assembly and one on the master oscillator.

A 28-connector cable harness connects the dc output voltages to terminal strips at the top and bottom of the backplane assembly. Figure 5-55 explains the distribution of voltages from the power supply to the MF20.

5.10.3 **Battery Pack** (Figure 5-56)
A battery pack, consisting of eight 12 V batteries which contain lead-acid cells, provides a backup dc power source. Circuitry in the power supply connects pin 2 to pin 6 (Figure 5-56A) to obtain 96 Vdc between pins 1 and 7. When the MF20 is powered down, the battery is open-circuited. During normal operation it is connected to the power supply circuits and kept charged. When the primary power fails, the battery is automatically cut in and used to provide alternate power. It is designed to ensure an adequate power source for 30 seconds to permit activating emergency power for UPS (uninterruptable power systems). See Paragraph 1.2 for battery pack specifications.

The battery is constantly trickle-charged with a current of 4 mA. Each 30 second discharge duration uses about 2.5% of the storage capacity of the cells and 18 hours of recharge time is required to recover this charge. Power failures of durations longer than 30 seconds are latched by a relay in the power supply which

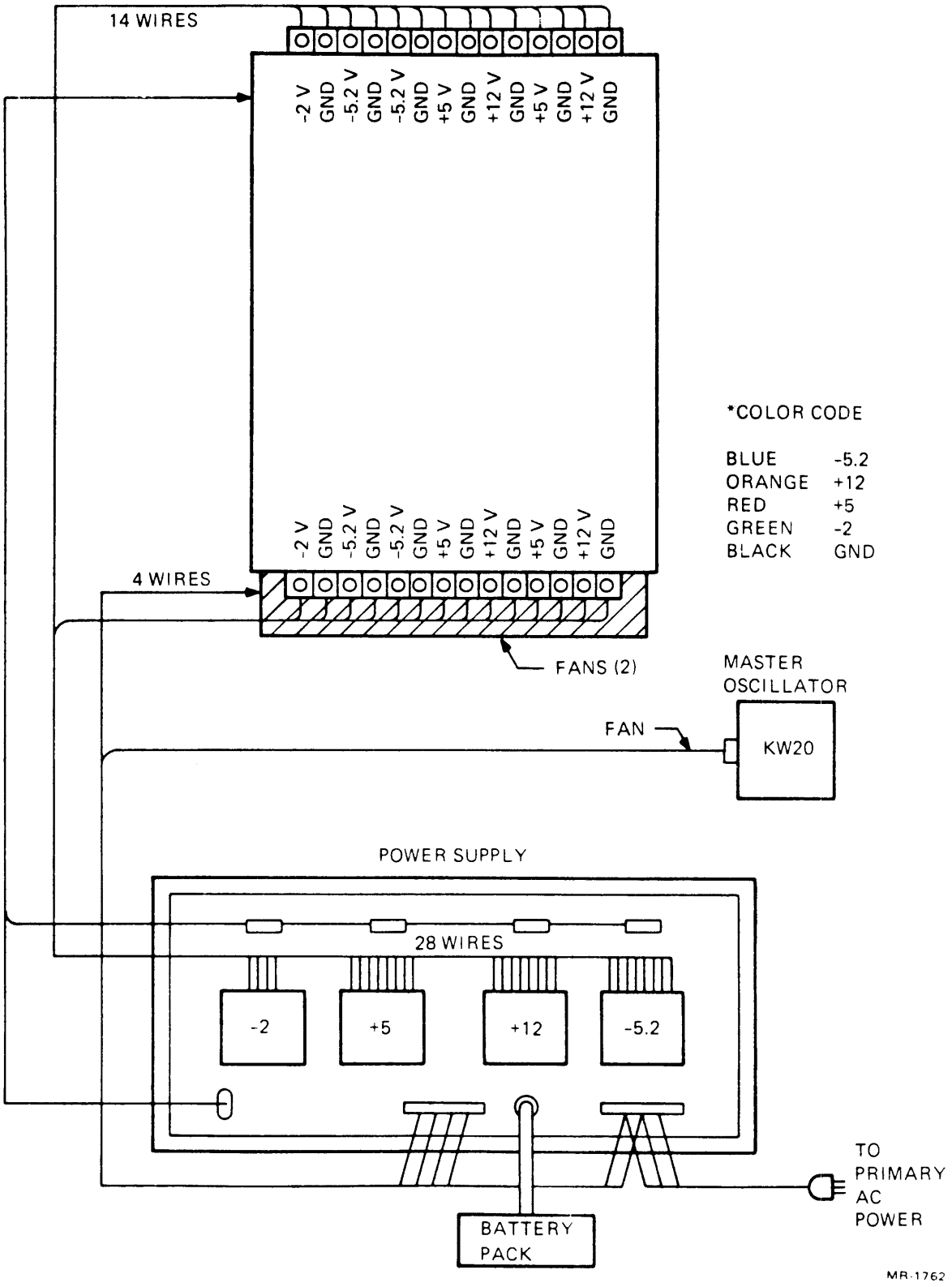


Figure 5-55 MF20 Power Distribution

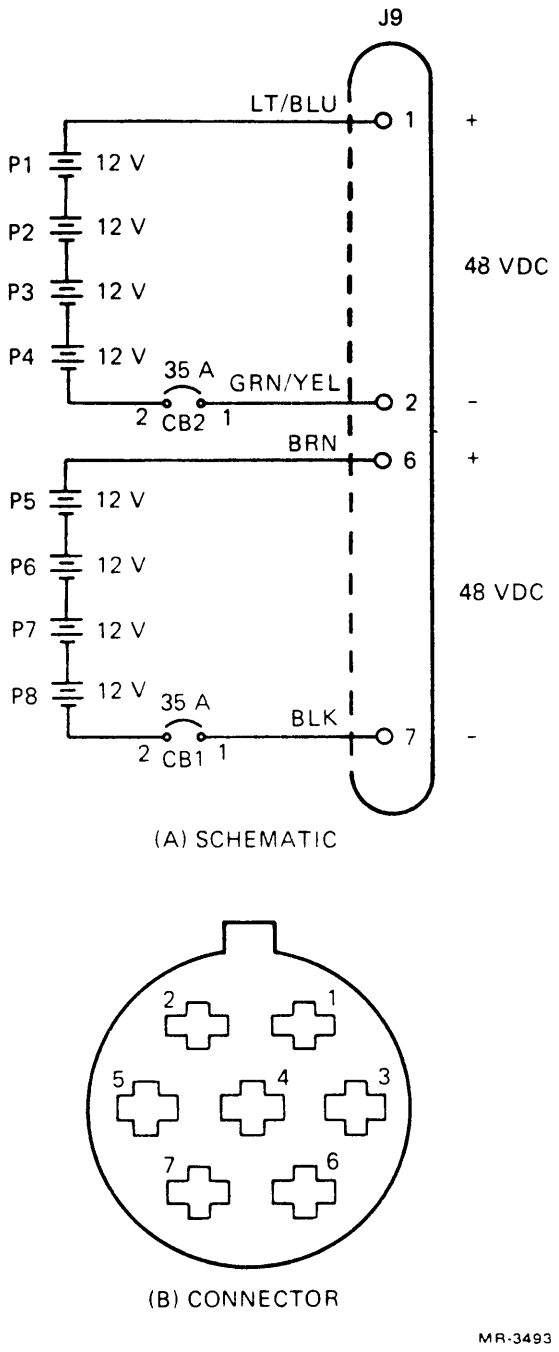


Figure 5-56 Battery Pack

disconnects the battery. Figure 5-56 shows the female cable connector that plugs into the H7131.

5.10.4 Summary

Each MF20 logic assembly uses a separate standalone power supply unit. Each unit provides all the necessary dc and ac voltages required to operate the MF20 system. A battery back-up supply is used to preserve data integrity within the MOS for 30 seconds after a power outage

CHAPTER 6
PREVENTIVE MAINTENANCE

6.1 INTRODUCTION

One of the key performance characteristics of today's computer systems is system availability. A comprehensive preventive maintenance program is one of the factors that can contribute to improving system availablity. If the program is carefully planned and effectively implemented, the need for corrective maintenance can be detected and scheduled during off-peak hours. Good PM procedures, coupled with a clear, easy to use record-keeping system, can also help to identify trends that permit correcting imminent problems before they become catastropic and require expensive corrective maintenance along with excessive down-time.

This chapter describes a suggested preventive maintenance program for the MF20 and includes:

1. The required test equipment
2. The detailed PM procedures.

The user may wish to adjust these procedures to best meet the needs of the individual DECSYSTEM-20 site. PM worksheets (EK-MF20-WS-000) are included in the DECSYSTEM-20 Site Manager's Guide. The detailed procedures described in this chapter are also included in the MF20 PM Procedures (EK-MF20-PM-000) on microfiche.

6.2 TEST EQUIPMENT REQUIRED

1. KL10-E model B processor
2. Digital VOM with ±20 millivolt accuracy
3. Dual trace oscilloscope - TEKTRONIX 475 or equivalent.
4. Three equal-length X10 oscilloscope probes.
5. W9025, 12-inch extender module
6. Revision 3A or later KLAD-20 diagnostic pack.

6.3 PM PROCEDURES

6.3.1 Run SYSERR

NOTE

Refer to the Tops-10 Tops-20 SYSERR Manual (AA-D533A-TK) for more detailed procedures on using SYSERR and analyzing the error reports.

1. Log onto the DECSYSTEM-20 as a normal user, preferably on a terminal that can furnish hard copy.

2. Run SYSERR by typing:

   @ R SYSERR <CR>

   Note that the @ is typed by TOPS-20 as a prompt.

3. When SYSERR indicates it is ready to accept a command (displays an *), type:

   * TTY:=/BEGIN:-1D <CR>

4. SYSERR will process all entries made in the error file during the past 24 hours and type out a summary on the terminal.

5. Analyze the error summary to identify trends that may indicate possible MF20 MOS storage subsystem problems.

6. File the typeout to maintain a record of system performance. This step will be subject to the individual site management procedures.

NOTE

The error retrieval process can be expanded and automated using a batch control file that resubmits itself periodically.

6.3.2 Run TGHA

1. Log onto the DECSYSTEM-20 as a privileged user.

2. Enable all privileges by typing:

   @ ENABLE <CR>

3. Start TGHA running by typing:

   @ R TGHA <CR>

4. Dump the history file TGHA.DAT on the line printer by typing:

   * DUMP <CR>

5. Retrieve the error history printout for the MF20 MOS storage subsystem. Refer to Paragraph 7.2.3.4 for a complete discription of how to analyze the TGHA output.

6. Kill the job and log off by typing:

   @ LOGOFF <CR>

CHAPTER 6
PREVENTIVE MAINTENANCE

6.1 INTRODUCTION

One of the key performance characteristics of today's computer systems is system availability. A comprehensive preventive maintenance program is one of the factors that can contribute to improving system availablity. If the program is carefully planned and effectively implemented, the need for corrective maintenance can be detected and scheduled during off-peak hours. Good PM procedures, coupled with a clear, easy to use record-keeping system, can also help to identify trends that permit correcting imminent problems before they become catastropic and require expensive corrective maintenance along with excessive down-time.

This chapter describes a suggested preventive maintenance program for the MF20 and includes:

1. The required test equipment
2. The detailed PM procedures.

The user may wish to adjust these procedures to best meet the needs of the individual DECSYSTEM-20 site. PM worksheets (EK-MF20-WS-000) are included in the DECSYSTEM-20 Site Manager's Guide. The detailed procedures described in this chapter are also included in the MF20 PM Procedures (EK-MF20-PM-000) on microfiche.

6.2 TEST EQUIPMENT REQUIRED

1. KL10-E model B processor
2. Digital VOM with ±20 millivolt accuracy
3. Dual trace oscilloscope - TEKTRONIX 475 or equivalent.
4. Three equal-length X10 oscilloscope probes.
5. W9025, 12-inch extender module
6. Revision 3A or later KLAD-20 diagnostic pack.

6.3 PM PROCEDURES

6.3.1 Run SYSERR

NOTE

Refer to the Tops-10 Tops-20 SYSERR Manual (AA-D533A-TK) for more detailed procedures on using SYSERR and analyzing the error reports.

1. Log onto the DECSYSTEM-20 as a normal user, preferably on a terminal that can furnish hard copy.

+12 V (orange/black)
+5 V (red/black)
+12 V (orange/black)

6.3.6 KW20 Clock Checks

With power applied to the system and KLDCP loaded into the PDP-11 front end check the KW20 regulated power supply and clock frequencies using the procedure outlined below.

1. Use the VOM to measure the -5.2 V supply as described in Figure 7-6. Record the reading in the appropriate worksheet.

2. Set the KW20 master oscillator frequency to 30 MHz by typing:

   >.FW 72/3 <CR>

3. Now use the oscilloscope to measure the clock waveforms at the CPU backplane and at each MF20 backplane. Record both PW (pulse width) and (pulse repetition time) PRT times. Use the clock testpoint at the upper right hand corner of the backplane. The PW should be approximately 10 ns and the PRT 33.3 ns.

4. Set the KW20 frequency to 31 MHz by typing:

   >. FW 72/1 <CR>

5. Repeat step 3 above to record PW and PRT at 31 MHz.

6. Set the KW20 frequency to 25 MHz by typing:

   >. FW 72/2 <CR>

7. Repeat step 3 above to record PW and PRT at 25 MHz.

8. Reset the KW20 clock to its nominal rate by typing:

   >. FW 72/3 <CR>

6.3.7 Run Diagnostics

1. Mount the KLAD pack and place the drive on-line.

2. Bootstrap the system to load KLDCP into the PDP-11 front end.

3. Set the PDP-11 SR = 000002.

4. Load DHKBF by typing:

   >. P DHKBF <CR>

5. Start the program by typing:

   >. SED 3 <CR>

6. Once under control of DIACON (* typed out by program) type:

   * TS1 <CR>

7. Let the program run a minimum of three passes and return control to KLDCP.

8. Now load DHKBG by typing:

   >. P DHKBG <CR>

9. Start the program by typing:

   >. SED 3 <CR>

10. Once under control of DIACON type:

    * TS1 <CR>

11. Let the program run a minimum of three passes and return control to KLDCP.

12. Now load MEMCON to configure the MF20 for running DFMMH. Type:

    >. P MEMCON <CR>

13. Start MEMCON by typing:

    >. SED <CR>

14. Determine physical resources by typing:

    > DP <CR>

15. If typeout indicates proper resources based on the known configuration, configure the memory by typing:

    > CM <CR>

16. Type Control-C to exit to KLDCP.

17. Set the PDP-11 console switches to 100000. Set the right-hand switches to 407000 by typing:

    >. SW407000 <CR>

18. Load SUBRTN by typing:

>. P SUBRTN

19. Load DFMMH by typing:

    >. P DFMMH <CR>

20. Start DFMMH by typing:

    >. STM <CR>

21. Wait for one complete error-free pass of DFMMH.

6.3.8 Run Voltage and Clock Margins

1. Mount the KLAD pack and place the drive on-line.
2. Boot front end to load KLDCP
3. Load DHKBF by typing:

   >. P DHKBF <CR>

4. Set the PDP-11 switch register to 100002(8) and then start the program by typing:

   >. SED <CR>

5. Select the fast clock rate (31 MHz) by typing:

   */MO 1 <CR>

6. Run one pass at this clock rate by typing:

   *TS1 <CR>

7. After one pass control will be returned to KLDCP (SW15=1)
8. Restart the program by typing:

   >. SED <CR>

9. Now set the clock rate to 25 MHz by typing:

   */MO 2 <CR>

10. Run one pass at this clock rate by typing:

    *TS1 <CR>

11. After control is returned to KLDCP restart the program by typing:

    >. SED <CR>

12. Select nominal clock rate (30 MHz) by typing:

    */MO3 <CR>

13. Now run one pass at 30 MHz by typing:

    * TS1 <CR>

14. Now load DHKBG by typing:

    >. P DHKBG <CR>

15. Repeat steps 4 through 13 above to run this program at all three clock rates.

16. At this point you are ready to run voltage margins using DHKBF and DHKBG.

17. Set the PDP-11 console switches register equal to 100002.

18. Load DHKBF by typing:

    >. P DHKBF <CR>

19. Start the program by typing:

    >. SED <CR>

20. Enable SHMOOO testing by typing:

    */SM

21. Now run one pass of the program by typing:

    * TS1 <CR>

22. After one pass of DHKBF load DHKBG by typing:

    >. P DHKBG <CR>

23. Repeat steps 19 through 21 to run one SHMOOO pass of DHKBG.

24. Enable using cache by typing:

    */CU <CR>

25. Now run one final pass of DAKBG.

26. Restart DHKBG by typing:

    >. SED <CR>

27. Enable the cache use switch by typing:

*/CU <CR>

28. Disable the SHMOOO test switch by typing:

    * /SM <CR>

29. Now run one pass by typing:

    * TS1 <CR>

6.3.9 Check Overtemperature Sensors (semi-annual)

These checks will test each of the airflow sensors capabilities to detect a cooling failure in any of the MF20 logic box assemblies, and subsequently provide a visual indication that a malfuntion has occurred.

> **CAUTION**
> **You will be instructed to disconnect power from a fan assembly for a given period of time. If for any reason you fail to receive the desired results (e.g., failure of an airflow indicator to light), do not allow any of the MF20 logic box(es) to run beyond a 2 minute interval with any given fan assembly disconnected. This may result in an overtemperature condition which will cause permanent damage to the system.**

1. Power down the system, and MF20 subsystem components.

2. Set the system fault to "override."

3. Separately disconnect each of the MF20 cooling fans.

4. Power up the system.

5. After a short period of time (not to exceed 45 seconds), the system airflow fault indicator should light.

6. Repeat steps 1 through 5 for each fan contained in the MF20 memory, and KW20 master oscillator until all fans have been tested.

7. Should any individual airflow sensor fail to activate within a 1 minute interval, this sensor should be considered defective, and must be replaced.

6.3.10 Check Door Interlocks

With the system and MF20 subsystem conponents powered down, set the system fault override switch to the override position. Apply power to the system and MF20 subsystem components.

1. Individually open each door to the MF20 memory logic box subsystem assemblies (one at a time).

CAUTION

Do not allow these doors to remain open for any time interval greater than 2 minutes. To do so may result in permanent system damage due to overheating conditions.

2. Every time a new door is opened, check to see that the "door open" indicator located in the KL10-E mainframe is illuminated.

3. Close and secure the door opened in step 2 above.

4. Check to see that the "door open" indicator is extinguished.

5. Repeat steps 1 through 4 until all logic box doors have been tested.

6.3.11 Check Battery Backup

At MF20 scheduled maintenance, check MF20 power supply for correct open circuit charge voltage and short circuit charge current from the battery connector of the power supply.

1. Disconnect battery cable from the power supply.

2. With a VOM, measure voltage between pins 1 (+) and 7 (-) to be 145 $\pm$8 Vdc.

3. Switch the VOM to current mode and measure current of approximately 16 milliamperes.

Every two years replace battery pack and return old pack to Marlboro.

When not connected to MF20 power supply, each pack must be recharged every six months or irreversible deterioration in battery capability may occur.

6.3.12 Check System Cabling Integrity

1. Ensure that the CPU and memory subsystems are powered down.

2. Inspect all power harness/cables for chaffing and general deterioration of insulating material.

3. Inspect all coaxial transmission cables from the KW20 master oscillator to the CPU and MF20 memory for possible breaks in the insulating sleeve.

4. Inspect all XBus cables between the CPU and the MF20 for correct cable routing and proper strain relief. Ensure that there is freedom of movement of the cables at the pivot point of the rear mounting door. Ensure that there are no sharp bends in the XBus cables, or breaks in the insulating material. Inspect XBus jumper cables (if memory exists) between MF20 and MF20 expansion beyond 768K.

5. Inspect all logic-box air filters for excessive dust/contaminants; clean where necesssary.

CHAPTER 7
CORRECTIVE MAINTENANCE

7.1 INTRODUCTION

This chapter begins with a description of the diagnostic software designed to detect and localize faults in the MF20 MOS memory. Recommended troubleshooting procedures follow the diagnostics. These procedures focus on how to relate the diagnostic program's error reports to specific hardware subassemblies to isolate the fault to a field replaceable unit (FRU). Next, the procedures for removing and replacing the faulty subassemblies are described. Finally, the procedures for performing any necessary electrical adjustments are specified.

The information and procedures included in this chapter are based on the following assumptions.

1. A single fault exists wthin the boundaries of the MF20 subsystem.

2. The maintenance philosophy dictates isolation to the field replaceable subassembly level (module, cable, power supply, etc.).

> NOTE
>
> Although chip-level isolation is not described in this chapter, the experienced field engineer should be able to couple the theory presented in Chapters 4 and 5 with the procedures given in Chapter 7 to perform chip-level isolation.

7.2 DIAGNOSTICS

For the purpose of corrective maintenance the term diagnostics includes any program or software mechanism that may be used to test the MF20. The diagnostic software designed to support the MF20 may be classifed into several categories. Three possible ways to classify them are as follows.

1. Whether the program is 11-based or 10-based

2. Whether the program is a test or a utility program

3. Whether the program runs in user mode or is a stand-alone program

The following paragraphs describe most of the diagnostic software supplied with the MF20 in terms of its purpose and general operating procedures. The following list summarizes the set of diagnostic software designed to support MF20 maintenance.

1. Executive Mode

   a. 11-Based 10

      DGQFB - MEMCON Diagnostic Memory Boot Program
      DHKBF - MF20 Diagnostic Part I
      DHKBG - MF20 Diagnostic Part II

   b. 10-Based 10

      DFMMH - KL10 4096K Memory Diagnostic

2. User Mode

   a. 11-Based 10 (none)
   b. 10-based 10

      TGHA - MF20 MOS RAM Error Analyzer
      SYSERR - System Error Report Program

3. Utilities

   a. 11-Based 10

      KLDCP - KL10 Diagnostic Console Control Program (REV 17 or higher)

      DIACON - 11-Based Diagnostic Control Program (imbedded within 11-based 11 diagnostic)

   b. 10-Based 10

      SUBKL - KL10 Subroutine Program

7.2.1 DGQFB (MEMCON)

MEMCON is the 11-based memory configurator program designed to run under control of KLDCP. It is written to support core-only, MOS/core, and MOS-only memory configurations, using either internal (MA20, MB20, MF20) or external (MG10, MH10, etc.) memory subsystems. MEMCON is loaded and run as part of the basic B or BT command string when bootstrapping the KL10 for the purpose of running diagnostics. A subset of MEMCON is included as part of the front-end software (RSX20F) to facilitate configuring memory prior to bringing up the TOPS-20 monitor. The following discussion will focus on the use of DIGQFB in conjunction with KLDCP to test and diagnose MF20 operation in executive mode.

7.2.1.1 Purpose - MEMCON provides the mechanism for initializing the MF20 after powering up the system. This initialization process involves loading all of the MF20's control RAMS so that it can respond properly to subsequent XBus requests for information storage and retrieval. It also permits the user to determine and modify the available memory configuration based on the needs of 10-based programs and the operational state of available memory

7.2.1.2 General Program Operation (Figure 7-1) - The user communicates with MEMCON via a set of commands typed on the operator's console. Until the KL10 memory is configured, no code can be loaded into the 10-side memory. To enable KL10 to communicate with the memory subsystem of the 10-side, 10-based code is loaded into the KL10's AC blocks. This program is executed to communicate with the 10-side memory.

MEMCON is resident in PDP-11 memory and consists of the following routines.

1. 11-based control routines interface with KLDCP to:

   a. Decode and execute configuration commands typed in by the user

   b. Display the status of the program's response to execution of any command or detection of a memory-related error.

2. 10-side routines are loaded into the KL10's AC blocks and executed to communicate with the 10-side memory subsystem.

7.2.1.3 Operating Procedures - To load and start MEMCON (DGQBF) the user types:

P MEMCON <ALT>

After the program has been loaded and started by KLDCP, it displays the message:

DIAGNOSTIC MEMORY BOOT VER X.X

Where X.X is the program's version number.

The program then signifies readiness to accept commands by displaying the prompt:

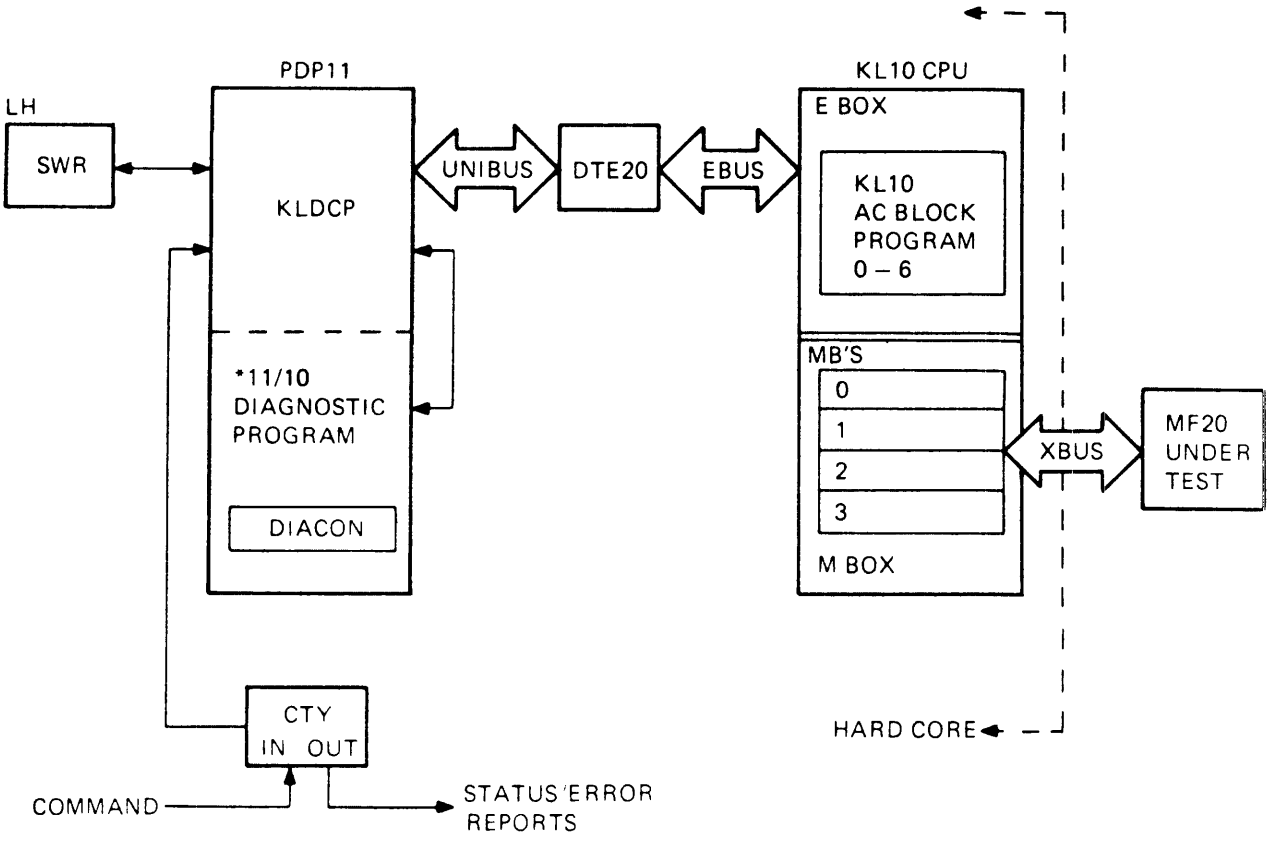
> <TAB>

At this point the user may type any one of the commands shown in Table 7-1. To obtain updated information on the commands listed in the table, the user should type:

> H <CR>

This will result in a typeout of MEMCON's help file. Any program changes not reflected in this manual will be displayed in the help file information.

The following examples illustrate the user-machine dialogue for typical uses of MEMCON. Underlining is used to indicate the user input commands.



*11/10 PROGRAMS

DHKBF MF20 DIAGNOSTIC (PART I)
DHKBG MF20 DIAGNOSTIC (PART II)
MEMCON MF20 MEMORY CONFIGURATOR

MR 1999

Figure 7-1 MF20 Diagnostic Overviev
11/10 Operational Block Diagram

Table 7-1 MEMCON Command Summary

Command	Function
Basic Memory Boot Commands	
DP	Determine physical memory resources and report. Starts from scratch.
KP C1, C2 - S1, S2	Kill physical resources S1 through S2 in memory controllers C1 through C2.
RP	Report physical resources as modified by previous KP command.
DL	Determine logical configuration, report it, but do not set it.
CM	Determine, report, and set the configuration. Then clear various memory boot variables set by previous commands. This command by itself is sufficient to configure memory.
TC	Test configuration.
DP, DL, and CM Switches	
0, 1, 2, or 4	Force MA20/MB20 interleave unless memory loss would result. Force DMA20 bus mode if legal. 0 (default) gives optimal results.
F	Force MF20 address reconfigure.
K	Keep bad MF20 blocks (ignored if /F switch is given).
R	Reverse configuration where possible. Normally used with /F switch.
S N SP	Substitute spare bit for bit N (decimal). SP is space or carriage return. If N is omitted no swapping occurs. Bit N is swapped in all sub-blocks of all MF20s.
	NOTE **No switches typed causes use of previous switches. DD, DL and CM switches may be in any order following the command.**
KP Command Format	
C1	A required controller number (octal)

Table 7-1 MEMCON Command Summary (Cont)

Command	Function
C2	An optional second controller number which specifies the last controller to be processed octal.
S1	A storage module number (MA20/MB20) or block number (MF20) to be deleted (octal). Absence of this parameter means entire controller is deleted.
S2	Optional second storage module or block octal number to indicate a range of storage modules or blocks to be deleted.
Utility Commands	
↑ C	Exit MEMCON
↑ Z	Exit MEMCON
DR R N	Dump the contents of the control RAM specified by R in MF20 controller number N R may be F, B, T, or A F = Fixed Value RAM B = Bit Substitution RAM T = Timing RAM A = Address Response RAM N = may be 10_8 through 17_8.
IC N	Force initialization of MF20 no. N.
MO N	Select master oscillator source. N = 3 30 MHz N = 0 external N = 2 25 MHz N = 1 31 MHz
RI	Reinitialize memory boot.
SD W	Execute SBus diagnostic function specified by W (octal) and display result.
SR	Execute an SBus reset.

Table 7-1 MEMCON Command Summary (Cont)

Command	Function
Debug Commands	
DA	Dump AC blocks and KL10 state.
PD	Patch diagnostic code.

NOTE

The file DBGOVL.A11 must be available for access by MEMCON to execute the DA and PD commands.

Example 1 - This example illustrates use of the DP command. The program response indicates a single MF20 with an address of 10_8 exists.

```
CMD:
>. RPO
>. F MEMCON
MEMCON.A11   VER 0.6   05-SEP-78
>. SED
DIAGNOSTIC MEMORY BOOT VER 0.6
>       DP

MEMORY RESOURCES:
CONTROLLER ADDRESS  TYPE  MODULES/GROUPS
                          7 6 5 4 3 2 1 0

        10          MF20
```

Example 2 - This example illustrates use of the KP command to delete three MF20 blocks, 0-2, and then report the modified configuration using the RP command.

```
>       KP 10 0,2
>       RP

MEMORY RESOURCES:
CONTROLLER ADDRESS   TYPE   MODULES/GROUPS
                            7 6 5 4 3 2 1 0

        10           MF20   0 0 0 0 0 0 0 1
```

Example 3 - This example illustrates use of the CM/F command to force a memory reconfigure after the KP command from example 2.

```
>       CM/F

LOGICAL MEMORY CONFIGURATION.
  ADDRESS   SIZE   INT   TYPE  CONTROLLER
   000000   64K     4    MF20   10
```

7.2.2 11-Based Diagnostics (Figure 7-2)

There are two 11-based MF20 diagnostics, DHKBF and DHKBG, that provide the primary troubleshooting tools for testing and repairing the MF20. Either program, when loaded, resides in the bottom 16K of PDP-11 core memory, and interfaces with KLDCP which resides in the upper 16K of PDP-11 core. To test the MF20, a KL10 test program must be loaded into the KL10's AC blocks and executed under control of the 11-based diagnostic. This test code, initially imbedded within the 11-based diagnostic, is transferred to the KL10 via the DTE20 under control of KLDCP and the 11-based diagnostic control code.

Once loaded and started, the AC block program executes 10-side code to test the MF20 via the MBox and appropriate XBus.

The results of the test are retrieved from the AC blocks by the 11-based control code via the DTE20 and KLDCP. If any errors are detected, they are displayed on the console terminal. This status/error information is transferred from the 11-based diagnostic to the console terminal via calls to KLDCP.

DIACON is assembled in with the 11-based diagnostic to provide the user with the means to control execution of the diagnostic. DIACON is enabled using a PDP-11 console switch (left-hand switches). Once enabled, it permits the user to selectively control execution of the diagnostic using a set of keyboard commands. Typing an H instructs DIACON to type out a short help file that summarizes its command set.

7.2.2.1 Diagnostic Test Flow (Figures 7-2 and 7-3) - Most of the individual tests with DHKBF and DHKBG exhibit a common test structure and program flow sequence. A typical diagnostic test consists of the following physically separate, but functionally interrelated programs.

1. An 11-based control program
2. A 10-based MF20 test program

The binary image of the 10-based test code resides in PDP-11 core initially. After executing a common initialization routine, the 11-based control program transfers this binary image into the KL10's AC blocks via KLDCP and the DTE20. Depending on the length and complexity of the test, more than one AC block may be used. The 11-based tests use the KL10's AC blocks as follows.

AC BLK0-5	These six blocks are used as needed to store up to 96 words (max) which comprise the 10-based test routines.
AC BLK-6	This block is initialized to contain a fast SBus diagnostic program called by the 11-based program.
AC BLK-7	Reserved for use by the KL10's microcode.

Pages 7-11 to 7-71
missing from original
document

positive-going edge of Y tick and before the positive-going edge of X tick.

9. Move probe 2 from E86 pin 5 to E86 pin 6. Rotate the upper switch to align the negative-going edge of CT B CLK DLY L (probe 2) with the negative-going edge of A tick (probe 1). If this optimal setting is not possible, the negative-going edge of CT B CLK DLY L must be aligned within the following range: after the positive-going edge of A tick and before the positive-going edge of the pulse following A tick.

10. Semi-Final check - Remove probe 1 from pin 9 of E86 and place it on pin 5. The waveforms observed on channels 1 and 2 should be logically opposite of one another.

11. Final Check - At the scope, move the probe monitoring A CHANGE COMING L from the EXT SYNC to CHANNEL 1. Sync internal on channel 1. Now remove the probes attached to the dip clip and place the dip clip on E61. Put channel 2 probe on E61 pin 2 (signal A PHS COM FREE 1 L in MF20) and place the ground clip on E61 pin 16. These signals should be identical.

NOTE

This equipment generates, uses, and may emit radio frequency. The equipment 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 radio frequency interference. Operation of this equipment in a residential area may 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.

SERVICE NOTE

To maintain the emissions integrity of this product, it is important that door seals, required grounding, etc. remain intact and are not modified.

8.1 INTRODUCTION

The MF20F external cabinet for the KL10-E or KL10-R is a MOS memory subsystem with storage up to 1.5 megawords. This is in addition to the existing memory.

8.2 PHYSICAL LAYOUT

The MF20F physical layout is shown in Figure 8-1.

8.3 MF20F CONFIGURATIONS

MF20F-LH - FCC compliant MF20-LH plus 1 MF20-LK, orange cabinet

MF20F-LM - FCC compliant MF20-LM plus 1 MF20-LK, blue cabinet

MF20F-LP - FCC compliant MF20-LP plus 1 MF20-LK, KW20, orange cabinet

MF20F-LS - FCC compliant MF20-LS plus 1 MF20-LK, KW20, blue cabinet

8.4 SITE PREPARATION

The MF20F external cabinet is an addition to the existing KL10-E or KL10-R. The KL10-R already consists of MF20 MOS memory found in the I/O cabinet. The existing KL10-R can have up to 1.5 megawords. Adding the MF20F cabinet will give the KL10-R up to 1.5 megawords more. Table 8-1 provides the mechanical, electrical and environmental data needed to plan the installation. Figures 8-2 and 8-3 show the required service area for KL10-R or MF20F cabinets. Refer to the KL10-Based Site Preparation, Power System Installation Manual for detailed information on site preparation.

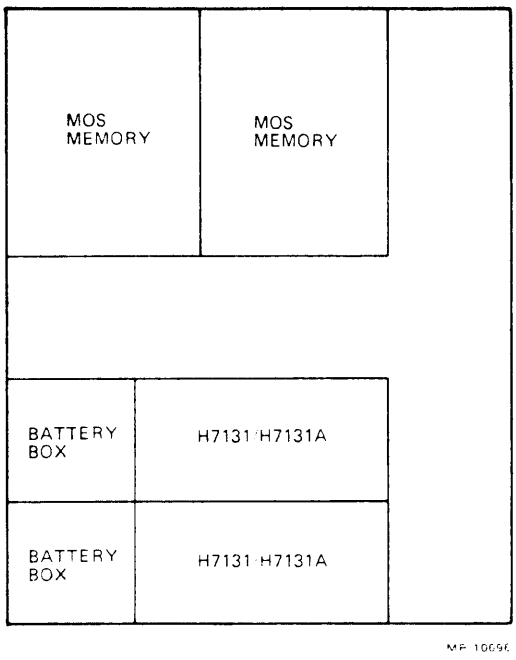


Figure 8-1 MF20F Physical Layout

Table 8-1 MF20F Characteristics

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
FS	249 kg 550 lb	152 cm 60 in	116 cm 45 in	76 cm 30 in	7020324	9907075-01

POWER (AC)

AC Voltage			Frequency		Steady State	Surge	Surge
Low	Nom	High	Tolerance	Phase(s)	Current (CRMS)	Current	Duration
104	120	127	60H 60 Hz ±	1	7.6 A	38 A	less than 200 ms

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
30 s Note 1	602 kg cal/hr 2390 Btu/hr	700	0.91	N/A Note 2	N/A Note 2	.5 mA

ENVIRONMENTAL (DEVICE)

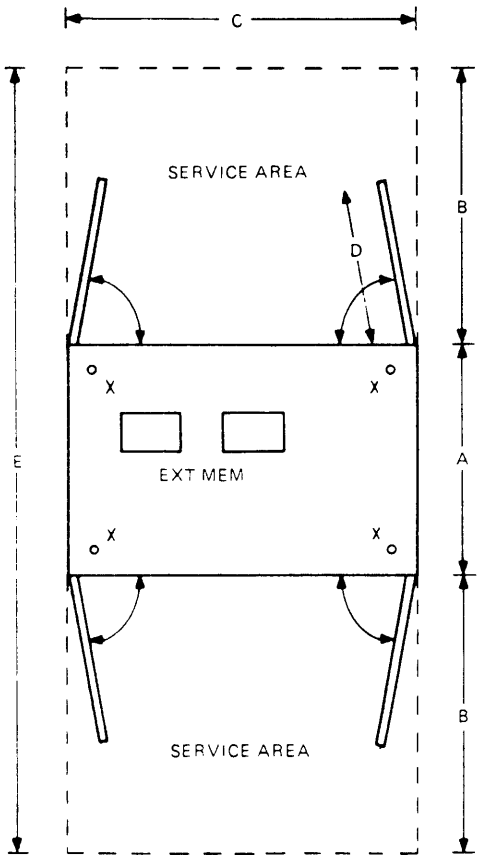
Temperature		Relative Humidity		Rate of Change		Air Volume
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	Inlet
15° to 32°C 59° to 90°F	-40° to 66°C -40° to 151°F	20 – 80%	0 – 90%	7°C/hr 12°F/hr	2%/hr	600 ft³/min

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
N/A	N/A	N/A	N/A	N/A	N/A

MAXIMUM CABLE LENGTH AND TYPE(S)

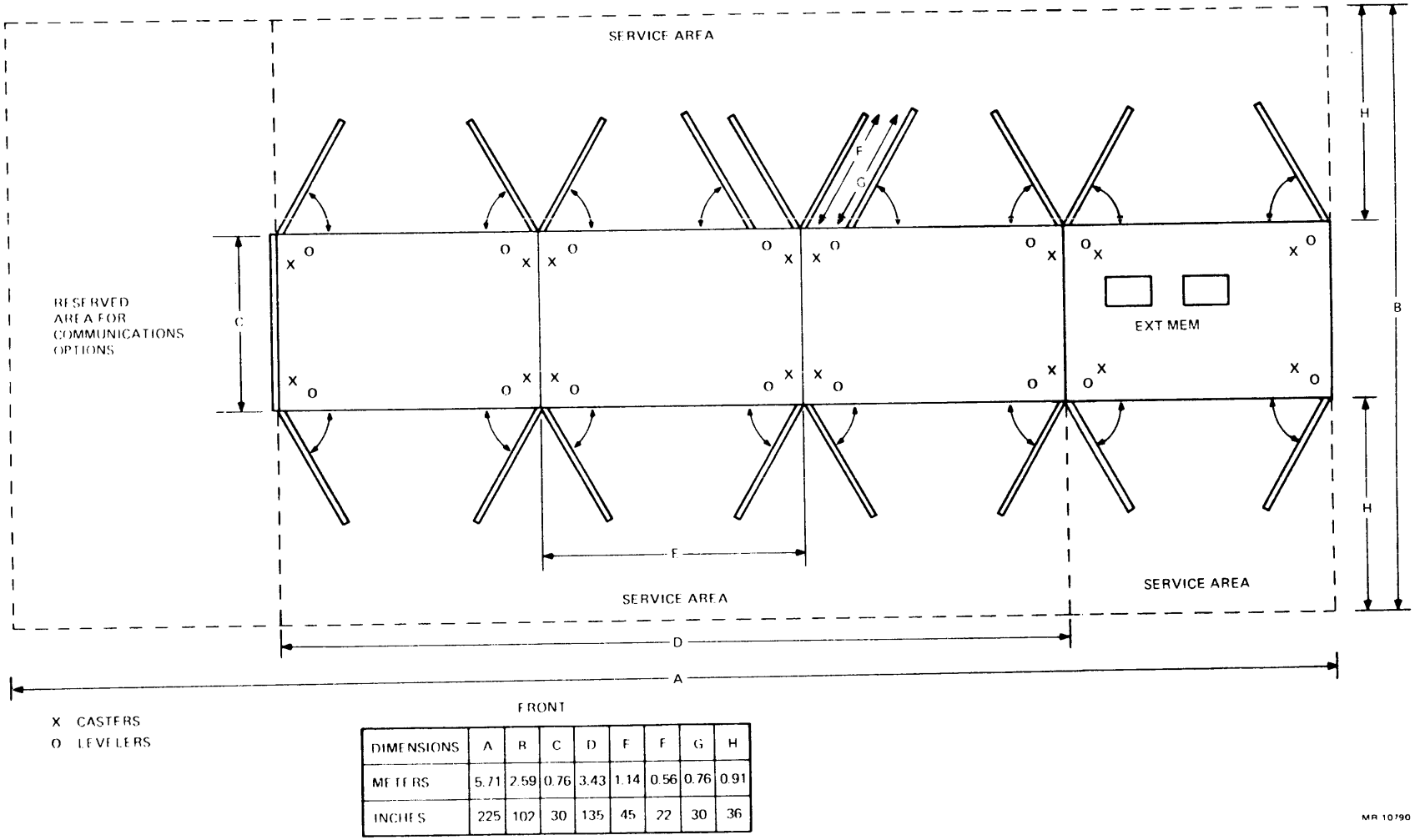
Memory	I/O Bus	Massbus	Device	Other
XBus Note 3	N/A	N/A	N/A	N/A



DIMENSIONS	A	B	C	D	E
METERS	.76	.91	1.14	.56	2.59
INCHES	30	36	45	22	102

MR 10412

Figure 8-2 MF20F Service Requirement Area



DIMENSIONS	A	B	C	D	E	F	G	H
METERS	5.71	2.59	0.76	3.43	1.14	0.56	0.76	0.91
INCHES	225	102	30	135	45	22	30	36

Figure 8-3 KL10-R and MF20F Service Requirement Area

8.5 REFERENCE DOCUMENTS

MF20F Field Maintenance Print Set (MP01711)
MF20 Field Maintenance Print Set (MP00622)
KL10-R Field Maintenance Print Set (MP01708)
KL10-Based Site Preparation, Power System, Installation Manual (EK-OKL10-SP)
MF20F Installation Procedure (A-SP-MF20F-0-1)

8.6 DIAGNOSTICS

1. KLDCP.BIN
2. DIAGB.RAM
3. DHKBA.A11
4. DHKBF.A11
5. DHKBG.A11
6. UB.RAM
7. MEMCON.A11
8. DFMMH.A10
9. SUBKL.A10

8.7 STANDARD TOOLS AND EQUIPMENT

1. Scope: Tektronix 475 or equivalent
2. Digital voltmeter (DVM)
3. DIP clip
4. Phillips screwdrivers (no. 2 and no. 3)
5. Dykes
6. Blade screwdriver
7. Adjustable wrench

8.8 UNPACKING AND INSPECTION

8.8.1 Unpacking

Digital has implemented strict shipping regulations to ensure that all hardware and software is received as stated in the customer's order. All boxes and skids should be left intact to avoid damaging or losing equipment, accessories, parts, or manuals. Before unpacking the MF20F from its shipping container the following checks should be made to verify the integrity of the shipment.

1. Ensure that all containers are sealed as evidenced by labels placed across the tape/band.

2. Verify that the box and skid housing the MF20 do not have any visible damage.

3. Verify that the correct number of boxes and skids was received. (These numbers will appear on one of the gummed labels affixed to the shipment.)

If you see any damage or if there are any missing items, report this to the shipper and the local DIGITAL field service branch supervisor responsible for the installation.

When ready to begin the actual installation, move the equipment to a convenient work area close to the DECsystem-10/DECSYSTEM-20. Remove the plywood panels and polyethylene bag used to protect the MF20F during shipment.

NOTE

When unpacking the MF20F shipment, save all the shipping containers and mounting hardware. These may be needed to return a damaged MF20F.

8.8.2 Inspection

Once the covers have been removed, carefully inspect the hardware. During the inspection, take careful notes of any discrepancies found and report them to the shipper and the local DIGITAL branch supervisor. Any discrepancies should also be reported on the Marlboro Manufacturing Feedback Form so that shipping problems may be corrected at the source. Inspection should include the following steps.

1. A complete inventory of all items as shown on the shipping list. The shipping list itself is part of the system.

   NOTE

   If any items are missing (i.e., cables, modules, etc.) do not proceed with the installation. Taking the customer's system down to install the MF20F and then finding you are unable to complete the job on schedule because of a missing part could cause a serious customer relations problem.

2. Inspect the logic assembly, power supply and battery box for any physical damage (dents, bent pins, broken wires, missing screws, etc.).

3. Verify that all the logic modules are plugged into the correct slots as shown in the module utilization list (MUL).

Once it has been verified that all the components needed for the installation are available, the actual MF20 installation may be performed as outlined in the following paragraphs.

8.9 INSTALLATION TO KL10R

(For installation to KL10-E, refer to the MF20F Installation Procedure.)

1. Remove the cabinet from the skid using the procedure on the shipping carton.

2. Remove the end panel, front and rear doors, top cover, and bezel from the CPU cabinet of the KL10.

3. Remove front and rear doors, top cover, and bezel from MF20F cabinet.

4. Place the MF20F at the KL where the end panel was removed.

5. Level the cabinet at the same height and bolt to KL. There must be a good fit on the cabinet gasket seal. Do not install if the gasket is missing or damaged.

6. See drawing E-UA-MF20F, sheet 5, for cable routing.

7. Route power supply ac cords along bottom rear of cabinets to 863 power control J25 and J32.

8. Connect fault harness J6 to P6.

9. Route clock coax cables to KW20 clock oscillator and connect to J2 and J4. Remove W2 and W4 in the KW20 (Figure 8-4).

10. Route remote turn on cable (P/N 7008288-8F) to upper MF20 power supply in the KL. Connect to J3.

11. Connect the M8572-YA "X" bus cable from MF20 no. 1 to slot 3 in the CPU.

12. Replace all doors, top covers, and bezels.

13. Using an ohm meter, check backplane dc voltages for shorts to ground (-2 V, -5.2 V, -5.2 V, +5 V, +12 V, +12 V).

14. Install controller number jumpers on backplanes. Refer to Table 3-1.

15. Power up the system.

16. Check the voltages at the MF20F backplane. They should be $\pm$ 2%.

17. Check S-BUS "VREF" by placing the voltmeter probe on pin F5A1 on MF20. It should read +1.40 $\pm$ .02 volts. If "VREF" is not within spec, replace M8580 or M8581 in slot 7 of CPU. (If new module is not available, swap modules in slots 7 and 8, to obtain same results. "Check VREF" on pin F5D1. If it is bad, replace the M8576 board.

18. Check the X-BUS sync with a scope using the following quick check.

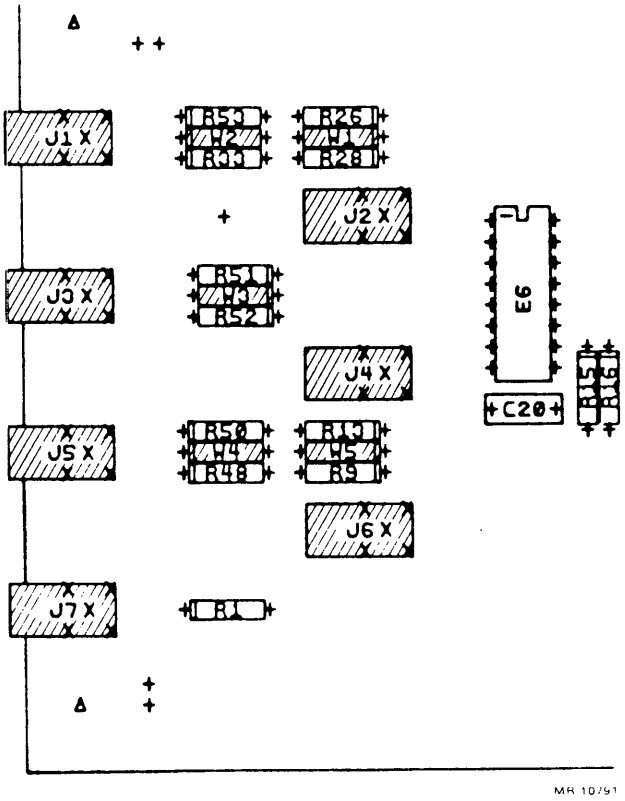


Figure 8-4 KW20 Clock Oscillator

- Bootstrap load KLDCP.
- Set X-BUS sync (the nominal setting of the two switches on the M8576 for XBUS deskew adjustment) as follows:
  - 13 foot XBUS cable: Turn each switch all the way clockwise and back off 1 click.
  - 3 foot, 8 inch XBUS cable: Turn each switch all the way clockwise and back off 2 clicks.

19. Set scope to 1 V/DIV for both chan 1 and chan 2. Set sweep rate to 20 ns/DIV. Load the KL10 microcode. Select full clock rate and source the clock from the master oscillator at 30 MHz. Start the microcode, (CR0, FW72/3, CS2, SM).

20. Place probe 1 on pin E22F2 in the CPU bay (signal "A change coming L") and place the clip on a ground pin. Place probe 2 on pin D5D1 in MF20 (signal "CTL2 A PHS COM FREE L") and place the clip on a ground pin. Synchronize internal on chan 1. These signals must be identical.

21. Move probe 1 from CPU bay to MF20 at pin C5M2 (signal "CTL2 B PHS COM FREE L") and place the clip on a ground pin. Set the scope to add channel 1 with channel 2. The waveform observed must be perfectly symmetrical with respect to on/off time.

22. If the sync does not check, the XBUS must be synchronized by following drawing A-SP-MF20-SYNC. If the sync does check, then proceed.

23. Run diagnostics DHKBA.A11 and DHKBF.A11.

8.10 VANE SWITCHES

To check the function of the vane switches, do the following.

1. Disconnect the ac wires from one fan (with system powered down) on the MF20 cooling assembly.

2. Power up the system.

3. The system should shut down within 30 seconds with an "airflow MEM 1" fault for external MF20F and an "airflow MEM 2" fault for internal memory.

4. Repeat steps 12.6.3.1 - 12.6.3.3 in Section 12 of the MF20F Installation Procedure for each of the MF20 cooling assembly fans.

5. Be sure to replace the ac wires when you are through.

6. Be sure battery backups are on.

8.11 TEST AND ACCEPTANCE

1. Run "B" string to completion.

2. Run one pass of DFMMH in reliability mode.

3. Boot KLAD monitor and run UETP for 4 hours.

4. Acceptance is complete.

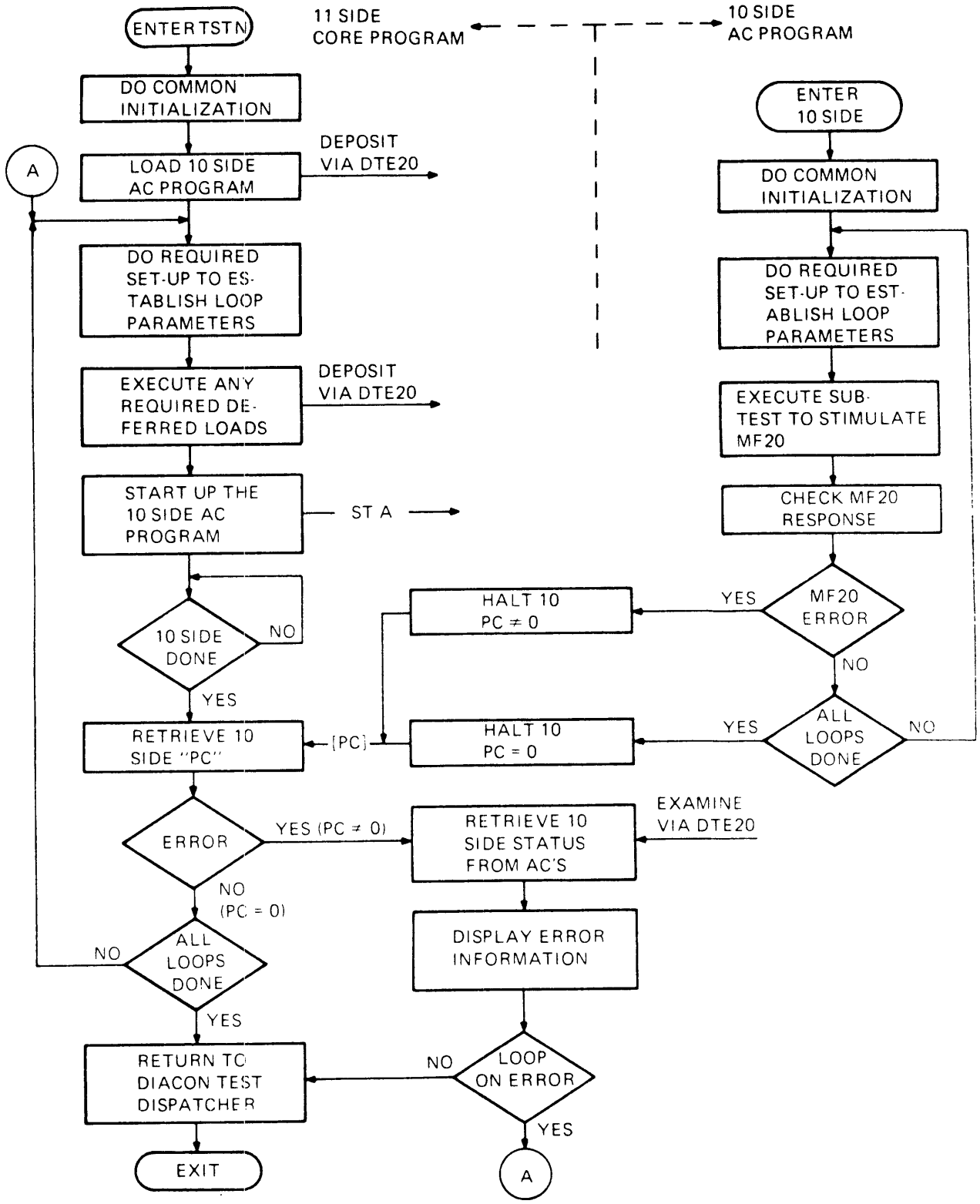


Figure 7-2 MF20 11-Based Diagnostic Test Simplified Flowchart

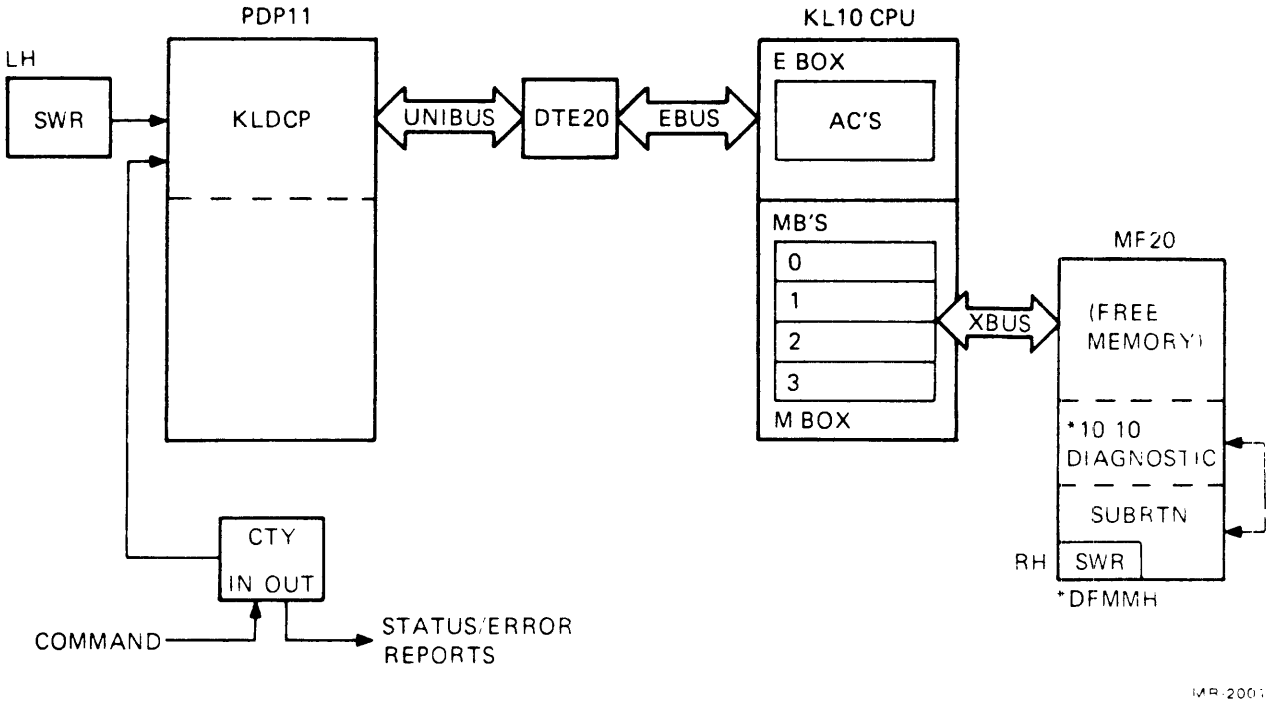


Figure 7-3 MF20 Diagnostic Overview
10/10 Operational Block Diagram

Some of the more elementary tests execute out of a single AC block, while more complex tests may use all six blocks. Any AC block not being used by the particular test being executed has all 16 of its locations initialized to contain 666666,,666666 pattern. This establishes a known reference in case the test AC block accesses an unused block.

Once the main body of the 10-based test code has been loaded into the KL10's AC blocks, the 11-based code executes the required set-up code to initialize the test parameters. When looping is invoked, this code establishes the proper test parameters for each pass through the loop. Finally, the 11-based code executes deferred loads to set up those KL10 ACs that may require initialization for each pass through the loop. An example of a deferred load would be modifiying the AC that contains the SBus diagnostic function word so that it points to the appropriate MF20.

At this time the 11-based control program loads the KL10's PC and sets the run flip-flop to start up the AC program on the 10-side. It then loops continuously, waiting for a timer to expire. The timer is initialized to allow sufficient time for the 10-side test code to complete its assigned task of testing the MF20 stimulus/response.

In parallel with the 11-based control program, the KL10 begins executing the MF20 tests specified by the AC program. After initializing the 10-side and setting its loop parameters, the AC program stimulates the MF20 and checks its response. If an error is detected by the KL10, the 10-based program halts with the PC equal to a non-0 value indicating the error number. If no errors are detected and all test iterations are complete, the KL10 halt clears the run flip-flop, a condition detectable by the 11-based control code that is looping.

Once the 11-side program senses that the KL10 halted, it retrieves the contents of the KL10's PC. A non-0 PC dispatches the 11-based control code to the appropriate routine that retrieves and displays the error information captured by the 10-based test and saved in an AC. A 0 PC indicates that the KL10 executed the tests error-free and signals the 11-based control program to proceed to the next iteration if required. This process continues until all possible 11-side and 10-side parameters have been tested, then the 11-based control program exits to DIACON's test dispatcher.

7.2.2.2 DHKBF (MF20 Diagnostic Part 1)- DHKBF consists of a set of 19 tests. Table 7-2 lists and briefly describes each test. Besides the ability to execute the tests listed in Table 7-2, the program contains a comprehensive set of console commands (Table 7-3) that permits the user to selectively call routines within the diagnostic to modify operation or obtain MF20 status information.

The following examples illustrate several ways of using DHKBF to test the MF20. Table 7-4 lists the PDP-11 console switch settings

Table 7-2 DHKBF Tests

Test No.	Description
1	Master oscillator test
2	Function 1 acknowledge test
3	Port loopback, SBus reset of port loopback
4	Control flag test (does more in QV mode)
5	SBus diagnostic data path and interference test
6	Voltage margin test (QV mode only)
7	Control RAM Galpat test
8	Refresh interval and address counters test
9	Single-step cycle tests
10	Single step mode incomplete cycle and clear 0-5 test
11	Address transmit/receive, parity, and hold-on-error tests
12	Control RAM parity error detection and clear 0-5 test
13	Address response/no response (NXM) test
14	Write path ECC generation test
15	ECC complement register function test
16	Write bad parity, detect, and clear 0-5 test
17	Write path spare-bit out test
18	SYN bit fixer select, ECC and syndrome register test using forced 0s.
19	Correction disabled test using forced 0s

Table 7-3 DHKBF Command Summary

Command	Function
Normal Use Commands	
NOTES:	1. To either command mode, set SW1 = 1 on the PDP-11 console prior to starting DHKBF. 2. Precede all DHKBF commands by typing a \ (backslash). 3. The absence of the \ will steer the command to DIACON. 4. Preceding the command with a period (.) steers the command to KLDCP.
/DR R N	Dump the contents of control RAM specified by R in controller N. R = F Fixed Value RAM R = B Bit Substitution RAM R = T Timing RAM R = A Address Response RAM (See /TC command)
/IC N	Initialize MF20 no. N
/MO N	Select master oscillator clock source N N = 3 Nominal 30 MHz N = 2 Fast 25 MHz N = 1 Slow 31 MHz N = 0 External NOTE After selecting MO source, type .CS2 to select KL10 clock source (external). Use care in selecting N = 0 as absence of external clocks will require powering down KL10 system and powering back up again to clear hung condition.
/QV	Toggle quick verify mode switch. This switch must be set to a 1 to execute voltage margin test (test 6) which requires operator intervention.
/RI	Reinitialize DHKBF.
/SD W	Execute SBus diagnostic function specified by W and display results.
/SM	Initialization for SHMOOOOO margin testing.

Table 7-3 DHKBF Command Summary (Cont)

Command	Function
/SR	Do an SBus reset.
/TC N	Test MF20 no. N only and set N as default for /DR, /IC, /SR, and /TR commands (default N = all).
/TN	Toggle test number typeout switch.
/VM V	Set voltage marign V (V is the integer part of the voltage).
	Commands Not for Normal Use
/CD S	Core dump-generated diagnostic as specified by string S.
/ER	Erase single-step recorded diagnostic function data.
Commands Residing in DBGOVL.All Overlay	
/DA	Dump KL10 state and AC blocks.
/PD	Patch PDP-11 diagnostic code.
/RA	.SM and restore saved ACs (/SA) to AC block for later restore (/RA).
/TI	Allows timing of PDP-11 code loop.
/TR	Define MF20 single-step cycle and trace its execution (see /TC).

Table 7-4 PDP-11 Console Diagnostic Switches

Switch No.	Function
SW15	Abort at completion of program pass
14	Print totals, restart test
13	Print totals, continue
12	Inhibit typeout
11	Print on PDP-10 line printer
10	Ring TTY bell on error
9	Loop on error
8	Halt on error
7	Print all errors
6	Reliability RUN mode
5	Inhibit error text
4	Inhibit paging
3	Modify device code
2	Inhibit cache
1	Operator selection of test
0	Special chain mode control switch

(left-hand switches) used to control program execution. The commands typed in by the user are indicated by underlining.

Example 1 - This example illustrates the loading and starting procedure that may be used to test the MF20. It uses the /TN switch to trace program progress.

```
.R DHKBF
DHKBF.A11  VER 0.2   20-SEP-78
>. SED
DIACON
*/TN
 ENABLED
*TS1

MAINDEC-10  DHKBF
MF20 DIAGNOSTIC, PART 1  VER 0.2  JAB29857
SWITCHES = 000002
F DIAGB.RAM
DIAGB.RAM   VER 156   "DIAGNOSTIC MICROCODE  15-FEBRUARY-1977"
UCODE VERSION 156, CLOCK RATE 0, PROCESSOR ID #1030.
1.
MEMORY RESOURCES:
CONTROLLER ADDRESS  TYPE  MODULES/GROUPS
                          7 6 5 4 3 2 1 0

        10          MF20

2. 3. 4. 5. 7. 8. 9. 11. 12. 13. 14. 15. 16. 17. 18. 19. END PASS 1.
1. 2. 3. 4. 5. 7. 8. 9. 11. 12. 13. 14. 15. 16. 17. 18. 19. END PASS 2.
1. 2. 3. 4. 5. 7. 8. 9. 11. 12. 13. 14. 15. 16. 17. 18. 19. $   PROM INTERRUPTED
 AT TEST 19.
```

Example 2 - This example illustrates how to use DHKBF to loop on tests 11 through 19. Note that typing an altmode $ interrupts the program to return control to DIACON. The /TN switch disables typing test numbers to trace progress.

```
>.SED
DIACON
*TS11

DHKBF
11. 12. 13. 14. 15. 16. 17. 18. 19. END PASS 1.
11. 12. 13. 14. 15. 16. 17. 18. 19. END PASS 2.
11. 12. 13. 14. 15. 16. 17. 18. $       PRGM INTERRUPTED AT TEST 18.
>./TN
?
>.SED
DIACON
*/TN
 DISABLED
*
```

Example 3 - This example illustrates use of the /TC and /TR switches to trace the progress of an idle refresh cycle in single-step mode.

NOTE

The checksum is a running sum of all the state flip-flops sampled at each tick. If the same operation is traced again, the same checksum should occur if no faults exist.

```
CMD:
>. SEL
DIACON
*/TC10
TEST CONTROLLER: 10
*/TR
RD, WR, R-F-W, OR IDLE REFRESH CYCLE (R,W,F,I)? I
ONE TICK AT A TIME (Y,N)? Y

CYCLE DESCRIPTION: IDLE REFRESH
>>> TICK 1.
TRUE:    ADT5 REFRESH NOW H,  CTL6 SUB RAM ADR 1 H,  CTL3 F WR ADR 35 H,
         CTL2 PHS COMING 1 L,  ADT4 TIM RAM RAS H
<CR> FOR NEXT TICK.
>>> TICK 2.
TRUE:    CTL1 REFRESH GO B H,  ADT5 REFRESH SEL H
FALSE:   CTL2 PHS COMING 1 L
<CR> FOR NEXT TICK.
>>> TICK 3.
TRUE:    CTL2 PHS COMING 1 L
<CR> FOR NEXT TICK.
>>> TICK 4.
FALSE:   CTL2 PHS COMING 1 L
<CR> FOR NEXT TICK.
>>> TICK 5.
*** "CTL2 PHS COMING 1 L" TYPOUT NOW INHIBITED
TRUE:    CTL1 RAS GO L,  ADT5 REFRESH GO+3 H,  ADT6 MOS RAS 0 H,
         ADT6 MOS RAS 1 H,  ADT6 MOS RAS 2 H,  ADT6 MOS RAS 3 H
<CR> FOR NEXT TICK.
>>> TICK 6.
<CR> FOR NEXT TICK.
>>> TICK 7.
TRUE:    ADT4 TIM RAM WE L,  ADT4 ADR 2ND HALF SEL H
<CR> FOR NEXT TICK.
>>> TICK 8.
<CR> FOR NEXT TICK.
>>> TICK 9.
<CR> FOR NEXT TICK.
>>> TICK 10.
TRUE:    ADT4 TIM RAM CAS H,  ADT4 TIM RAM PAR H
<CR> FOR NEXT TICK.
>>> TICK 11.
```

```
<CR> FOR NEXT TICK.
>>> TICK 12.
<CR> FOR NEXT TICK.
>>> TICK 13.
<CR> FOR NEXT TICK.
>>> TICK 14.
TRUE:   ADT4 TIM DATA RDY L
FALSE:  ADT5 REFRESH SEL H,  ADT4 TIM RAM PAR H
<CR> FOR NEXT TICK.
>>> TICK 15.
TRUE:   ADT4 TIM RAM PAR H
FALSE:  ADT5 REFRESH NOW H,  ADT4 TIM DATA RDY L
<CR> FOR NEXT TICK.
>>> TICK 16.
<CR> FOR NEXT TICK.
>>> TICK 17.
FALSE:  ADT6 MOS RAS 0 H,  ADT6 MOS RAS 1 H,  ADT6 MOS RAS 2 H,
        ADT6 MOS RAS 3 H,  ADT4 TIM RAM RAS H,  ADT4 TIM RAM CAS H,
        ADT4 TIM RAM WE L,  ADT4 ADR 2ND HALF SEL H
<CR> FOR NEXT TICK.
>>> TICK 18.
<CR> FOR NEXT TICK.
>>> TICK 19.
<CR> FOR NEXT TICK.
>>> TICK 20.
TRUE:   ADT4 TIM RAM BUSY CLR H
FALSE:  ADT4 TIM RAM PAR H
*** EVENT
<CR> FOR NEXT TICK.
>>> TICK 21.
TRUE:   ADT4 TIM RAM PAR H
FALSE:  CTL1 RAS GO L,  ADT5 REFRESH GO+3 H,  CTL1 REFRESH GO B H,
        ADT4 TIM RAM BUSY CLR H
<CR> FOR NEXT TICK.
>>> TICK 22.
TRUE:   ADT4 TIM RAM RAS H
FALSE:  ADT4 TIM RAM PAR H
<CR> FOR NEXT TICK.
>>> TICK 23.
<CR> FOR NEXT TICK.
>>> TICK 24.
*** ACTION 204

*** CHKSM 005574
**********************************************************************
```

7.2.2.3 DHKBG (MF20 Diagnostic Part 2)

DHKBG consists of a set of 21 tests. Table 7-5 lists and briefly describes each test. Like DHKBF, DHKBG includes a set of console commands (Table 7-6) that permits the user to selectively call routines within the diagnostic to modify operation or obtain MF20 status information.

The following examples illustrate several typical uses of DHKBG to test operation of the MF20. Like DHKBF, this program uses the PDP-11 console switches described in Table 7-4. Commands typed by the user are underlined.

Example 1 - This example illustrates a typical program load and run. The PROM data and bit swap data are typed only on the initial pass of the program. Note how typing the ALTMODE caused an exit to DIACON.

```
CMD:
>. F DHKBG
DHKBG.A11  VER 0.2  6-SEP-78
>. SED
DIACON
*TS1

MAINDEC-10  DHKBG
MF20 DIAGNOSTIC, PART 2  VER 0.2  JAB29857
SWITCHES = 000002
UCODE VERSION 156, CLOCK RATE 0, PROCESSOR ID #1030.

MEMORY RESOURCES:
CONTROLLER ADDRESS  TYPE  MODULES/GROUPS
                          7 6 5 4 3 2 1 0

        10          MF20

SM PROM DATA FOR
 CNTRL 10 GRP 0 FLD 0  DATA 01 111 011 100 001 100 110 111 000 110 010
 CNTRL 10 GRP 0 FLD 1  DATA 01 111 011 100 000 110 010 111 110 110 000
 CNTRL 10 GRP 0 FLD 2  DATA 01 111 011 100 001 101 011 111 100 110 010
 CNTRL 10 GRP 0 FLD 3  DATA 01 111 011 100 001 101 011 011 000 110 010

C# G B  SUBBLOCK 0      SUBBLOCK 1      SUBBLOCK 2      SUBBLOCK 3
10 0 0  OK, NO SWAPS    OK, NO SWAPS    OK, NO SWAPS    OK, NO SWAPS
10 0 1  OK, NO SWAPS    OK, NO SWAPS    OK, NO SWAPS    OK, NO SWAPS
10 0 2  OK, NO SWAPS    OK, NO SWAPS    OK, NO SWAPS    OK, NO SWAPS
10 0 3  OK, NO SWAPS    OK, NO SWAPS    OK, NO SWAPS    OK, NO SWAPS
END PASS 1.
$       PRGM INTERRUPTED AT TEST 14, SUBTEST 16659.
```

Table 7-5 DHKBG Tests

Test No.	Description
1	Master oscillator test
2	Function 1 acknowledge test
3	Storage module PROM read and verification
4	Group loopback group select test
5	Group loopback data and ECC paths
6	Syndrome network test using group loopback
7	Final read path correction and clear 0-5 test via group loopback
8	Storage module control signal generation test
9	Storage double control signal propagation test
10	Group, block, sub-block uniqueness test
11	MOS address generation test
12	MOS address propagation test
13	Data path to RAM test
14	SYN board spare bit in test using RAM
15	Double-bit error detect and clear 0-5
16	Full speed read-pause-write test
17	Refresh function verification test for each RAS address
18	March pattern through RAMs
19	Page refill cycle
20	Cache multiword read/write (only if enabled by /CU switch)
21	Multiword read/write with bit substitution

Table 7-6 DHKBG Command Summary

Command	Function
Normal Use Commands	
	Note: (Refer to Table 7-3)
/CU	Toggle cache use switch
/DR R N	Same as DHKBF (Table 7-3)
/IC N	Same as DHKBF (Table 7-3)
/MO N	Same as DHKBF (Table 7-3)
/QV	Same as DHKBF (Table 7-3)
/RI	Same as DHKBF (Table 7-3)
/SD W	Same as DHKBF (Table 7-3)
/SM	Same as DHKBF (Table 7-3)
/SR	Same as DHKBF (Table 7-3)
/TC N	Same as DHKBF (Table 7-3)
/TN	Same as DHKBF (Table 7-3)
/VM V	Same as DHKBF (Table 7-3)
Commands Not for Normal Use	
/CD S	Same as DHKBF (Table 7-3)
Commands Residing in DBGOVL.All Overlay	
/DA	Same as DHKBF (Table 7-3)
/PD	Same as DHKBF (Table 7-3)
/RA	Same as DHKBF (Table 7-3)
/SA	Same as DHKBF (Table 7-3)
/TI	Same as DHKBF (Table 7-3)

Example 2 - This example illustrates use of the CU, TN, and VM switches when running DHKBG.

```
>.SED
DIACON
*/CU
TESTS USING CACHE ENABLED
*/TN
 ENABLED
*/TN
 DISABLED
*/VM
VOLTAGE MARGINS: NONE
*/VM5
VOLTAGE MARGINS: 5.25
*/VM
VOLTAGE MARGINS: 5.25
*/VM0
VOLTAGE MARGINS: NONE
*/VM4
VOLTAGE MARGINS: 4.75
*/VM0
VOLTAGE MARGINS: NONE
```

Example 3 - This example illustrates use of the DR command to dump the contents of the address response RAM.

```
*/DR
FIXVAL, BITSUR, TIMING, OR ADRESP RAM (F,B,T,A)? A
CONTROLLER? 10
000/ 100 002 004 106 111 013 015 117 121 023 025 127 031 133 135 037
020/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
040/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
060/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
100/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
120/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
140/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
160/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
200/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
220/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
240/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
260/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
300/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
320/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
340/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
360/ 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001 001
```

Example 4 - This example illustrates use of the TC and SD commands to execute and display the results of SBus diagnostic functions (function 01). Note that selecting MF20 11 causes a response of all 0s since only MF20 10 exists.

NOTE

If the TC command has selected a specific MF20, it is used as a default in the left half of the SD command word.

```
*/TC 10
TEST CONTROLLER: 10

*/SD 200000 1
000500 002000

*/SD 0 1
000500 002000

*/SD 1
000500 002000


*/TC 11
TEST CONTROLLER: 11
*/SD 1
000000 000000
```

Example 5 - This example illustrates use of the MO command to change the KW20 clock rate.

```
*/MO
MASTER OSC SRC (3=NORMAL, 2=SLOW, 1=FAST, 0=EXTERNAL)? 3

*/MO
MASTER OSC SRC (3=NORMAL, 2=SLOW, 1=FAST, 0=EXTERNAL)? 0
ARE YOU SURE (Y,N)? N
MASTER OSC SRC (3=NORMAL, 2=SLOW, 1=FAST, 0=EXTERNAL)? 3

*/MO
MASTER OSC SRC (3=NORMAL, 2=SLOW, 1=FAST, 0=EXTERNAL)? 3

*/MO2

*/MO
MASTER OSC SRC (3=NORMAL, 2=SLOW, 1=FAST, 0=EXTERNAL)? 3
```

Example 6 - This example illustrates how the user would dump the contents of the KL10's ACs to analyze the 10-based 10 test code.

```
CMD:
>. SED
DIACON
*/DA

KL10 IS HALTED, VMA=000000
PC=000000, PREV BLOCK=0, CURRENT BLOCK=0
ERA: 600000 140023     APR: 000000 000000
ACBLK 0
PREV AC BLK 0, SEL AC BLK 0
EM0,17
000000/700500 000016
000001/700500 000014
000002/666666 666666
000003/104500 000012
000004/241540 777777
000005/647700 000050
000006/254000 000000
000007/571540 777767
000010/667600 003770
000011/254000 000000
000012/254200 000000
000013/666666 666666
000014/666666 666666
000015/666666 666666
000016/210000 000000
000017/006140 140023

ACBLK 1
PREV AC BLK 0, SEL AC BLK 1
EM0,17
000000/201740 000077
000001/201717 000000
000002/242700 777777
000003/431717 000000
000004/256200 000013
000005/641640 000010
000006/137700 000014
000007/700500 000015
000010/104000 000020
000011/365740 000001
000012/666666 666666
000013/212700 000015
000014/270600 000015
000015/666666 666666
000016/666666 666666
000017/666666 666666
```

```
ACBLK 2
PREV AC BLK 1, SEL AC BLK 2
EMO,17
000000/205700 400000
000001/246715 000000
000002/241740 000010
000003/405740 000376
000004/256200 000012
000005/344740 000010
000006/422640 000016
000007/644740 000376
000010/436640 000016
000011/104000 000032
000012/331640 000013
000013/666666 666666
000014/666666 666666
000015/666666 666666
000016/666666 666666
000017/666666 666666

ACBLK 3
PREV AC BLK 2, SEL AC BLK 3
EMO,17
000000/256200 000011
000001/666666 666666
000002/700500 000012
000003/242540 777753
000004/316640 000016
000005/312540 000017
000006/254200 000001
000007/700500 000014
000010/104000 001112
000011/120700 000016
000012/666666 666666
000013/666666 666666
000014/666666 666666
000015/666666 666666
000016/666666 666666
000017/666666 666666

ACBLK 4
PREV AC BLK 3, SEL AC BLK 4
EMO,17
000000/666666 666666
000001/666666 666666
000002/666666 666666
000003/666666 666666
000004/666666 666666
000005/666666 666666
000006/666666 666666
000007/666666 666666
000010/666666 666666
000011/666666 666666
000012/666666 666666
000013/666666 666666
000014/666666 666666
000015/666666 666666
000016/666666 666666
000017/666666 666666
```

```
ACBLK 5
PREV AC BLK 4, SEL AC BLK 5
EMO,17
000000/666666 666666
000001/666666 666666
000002/666666 666666
000003/666666 666666
000004/666666 666666
000005/666666 666666
000006/666666 666666
000007/666666 666666
000010/666666 666666
000011/666666 666666
000012/666666 666666
000013/666666 666666
000014/666666 666666
000015/666666 666666
000016/666666 666666
000017/666666 666666

ACBLK 6
PREV AC BLK 5, SEL AC BLK 6
EMO,17
000000/400600 000000
000001/322600 000001
000002/200700 000014
000003/700500 000016
000004/430600 000017
000005/420600 000015
000006/322600 000001
000007/254200 000001
000010/000000 101144
000011/000000 000000
000012/011000 000111
000013/000000 000000
000014/000000 000000
000015/760004 077407
000016/202174 000007
000017/002170 000000
```

7.2.3 10-Based Diagnostics

Unlike the 11-based diagnostics, the 10-based program executes from the 10-side memory itself. Before any 10-based program can be loaded, the 11-based configurator, MEMCON, must be run to configure the MF20 into the available address space and load its control RAMs. The 10-based diagnostics consist of DFMMH, an executive mode diagnostic program, and two user mode programs SYSERR and THGA.

7.2.3.2 DFMMH (Figure 7-3) - This is the KL10 4096K memory diagnostic. Prior to executing DFMMH, the common diagnostic subroutines, SUBRTN, must be loaded into the 10-side memory. These 10-side programs are loaded via the PDP-11 under control of KLDCP as directed by commands typed on the operator's console. The load path is into the MF20 via the SBus, MBox, EBox, DTE20, and PDP-11 core. Once loaded, DFMMH tests the MF20 as directed by:

1. Commands typed on the console
2. The setting of the PDP-11 console switches (left-hand switches),
3. The setting of the right-hand switches (as typed by the user).

Once started, the program displays status and error information on the console via KLDCP, the DTE20, and SUBRTN.

DFMMH is a 10-based, executive mode-only memory diagnostic that tests all of memory unoccupied by the program itself (approximately 17K at the bottom of the address space 0-17K). The program consists of the following five tests.

Test 1	Physical Address Test
Test 2	Address Test
Test 3	Worst-Case Patterns Test
Test 4	Floating 1s/0s Test
Test 5	Data Patterns Test

Tables 7-7 and 7-8 list the control switch functions that permit user control of the diagnostic. The following examples illustrate several variations of how the user might operate the system when running DFMMH. In each of the examples the user's typed inputs are underlined.

Example 1 - This example illustrates the standard procedures for configuring the memory, loading SUBRTN, and loading and running DFMMH. The switch settings SWITCHES = 100000 407000 result in printing totals after each test, printing correctable errors, and executing all tests.

Table 7-7 DFMMH Left-Hand Switches (PDP-11)

Switch No.	Function
15	Abort program on pass complete
14	Print operating totals, restart test
13	Print totals after each test
12	Inhibit all printout except forced printout
11	Print all data on line printer
10	Ring bell on error
9	Loop on error
8	Halt on error
7	Print all errors
6	Reliablity run
5	Inhibit error text
4	Inhibit paging
3	(Not used)
2	Inhibit cache
1	Operator selection of test parameters

Table 7-8 DFMMH Right-Hand Switches (KL10)

Switch No.	Function
18	Type correctable errors as they occur.
19	Ask about margins.
20	Ask about fast memory addressing bit.
21-23	= 0 All patterns = 1 All 1s = 2 All 0s = 3 Alternate bit pattern = 4 Floating 1s = 5 Floating 0s = 6 Floating 1s and 0s = 7 Pseudorandom, parity bit check
24-26	= 0 Physical address, patterns - With reliability switch = 1 (Table 7-8), runs all the following tests 1-5. = 1 Data patterns = 2 Address = 3 Worst-case patterns = 4 Float 1s/0s = 5 Physical address = 6 Data patterns and worst-case = 7 Runs tests 1-5
27	Enable looking at 28 through 35
28-29	Specify which million words to test: 00 1st million 01 2nd million 10 3rd million 11 4th million
30-32	Specify which 128K of selected million.
33-35	Test 32K segment of selected 128K.

```
*^C
CMD:
>. F MEMCON
MEMCON.A11   VER 0.6   05-SEP-78
>. SED
DIAGNOSTIC MEMORY BOOT VER 0.6
>       CM
STARTING MF20 DBE SCAN.  WAIT 25 SEC/256K.

LOGICAL MEMORY CONFIGURATION.
  ADDRESS  SIZE  INT  TYPE CONTROLLER
   000000  256K   4   MF20  10

CMD:
>. F SUBRTN
SUBRTN.A10   VER 0.13   27-APR-77
>. STD
>.
DECSYSTEM DIAGNOSTIC SUBROUTINE'S
VERSION 0.13, SV=0.13, CPU#=1030, MCV=156, MCO=100, HO=36, 60HZ

SWITCHES = 000000 000000
CLK SOURCE = EXTERN, CLK RATE = FULL, AC BLK 6 , CACHE: 0

MEMORY MAP =
FROM     TO          SIZE/K
00000000 00777777       256

END PASS 1.
>. SW 407000
>. SW
100000 407000
>. F DFMMH$
DFMMH.A10   VER 0.3   06-SEP-78
>.
PDP-10 KL10 4096K MEMORY DIAGNOSTIC (DFMMH)
VERSION 0.3, SV=0.13, CPU#=1030, MCV=156, MCO=100, HO=36, 60HZ
```

```
SWITCHES = 100000 407000
CLK SOURCE = EXTERN, CLK RATE = FULL, AC BLK 6 , CACHE: 0

MEMORY MAP =
FROM      TO          SIZE/K
00000000 00777777        256


PHYSICAL ADDRESS TEST, PASS COUNT  0
ERROR TOTALS : NONE


ADDRESS TEST, PASS COUNT  0
ERROR TOTALS : NONE


WORST CASE PATTERNS TEST, PASS COUNT  0
ERROR TOTALS : NONE


FLOATING ONES/ZEROS TEST, PASS COUNT  0
ERROR TOTALS : NONE

DATA PATTERNS TEST, PASS COUNT  0
ERROR TOTALS :

TEST COMPLETION, PASS COUNT  0
ERROR TOTALS : NONE

END PASS 1.
^C
```

Example 2 - This example illustrates the use of switch 1 on the PDP-11 console to invoke the dialogue between user and program that permits modifying the testing parameters.

```
CMD:
>. STM
>.
PDP-10 KL10 4096K MEMORY DIAGNOSTIC (DFMMH)
VERSION 0.3, SV=0.13, CPU#=1030, MCV=156, MC0=100, H0=36, 60HZ

SWITCHES = 100010 407000
CLK SOURCE = EXTERN, CLK RATE = FULL, AC BLK 6 , CACHE: 0

MEMORY MAP =
FROM      TO          SIZE/K
00000000 00777777        256

DO YOU WANT THE FOLLOWING SWITCHES SET?

PRINT PASS COMPL OPERATING TOTALS?
 Y OR N <CR> - Y

INHIBIT DATA COMPLEMENT, FAST RATE ADDRESSING?
 Y OR N <CR> - N

INHIBIT ERROR CHECKING ON MASK?
 Y OR N <CR> - Y

INHIBIT BLT EXERCISING CYCLE?
 Y OR N <CR> - N

INHIBIT FAST RATE ADDRESSING?
 Y OR N <CR> - N

INHIBIT READ-RESTORE CYCLES - WCW,RCW CYCLES?
 Y OR N <CR> - N

INHIBIT PARITY ERROR TYPEOUT?
 Y OR N <CR> - N

INHIBIT DATA ERROR TYPEOUT?
 Y OR N <CR> - N

TYPE CORRECTABLE ERRORS AS THEY OCCUR?
 Y OR N <CR> - Y
```

```
TEST SELECTION
PHYSICAL ADDRESS, Y OR N <CR> - Y
DATA PATTERNS - 1 TO 7, Y OR N <CR> - Y
ADDRESS, Y OR N <CR> - Y
WCP - 1 TO 4, Y OR N <CR> - Y
ME10 PRESENT?, Y OR N <CR> - N
FLOATING ONES/ZEROS, Y OR N <CR> - Y

SPECIFY FAST RATE MEMORY ADDRESSING BIT (0 (FOR ALL) OR 18 TO 35) - 0

VOLTAGE MARGINS ARE NOW:
MF20:  NOMINAL FOR ALL MOS CONTROLLERS

Y TO ASK, N FOR NOMINAL, D FOR DEFAULT LIST, ELSE FOR NO CHANGE - N

SELECTION COMPLETED, IS IT OK, Y OR N <CR> - Y


PHYSICAL ADDRESS TEST, PASS COUNT  0
ERROR TOTALS : NONE

^C
```

Example 3 - This example illustrates a typical error printout that occurred during a normal program run. Note that the printout begins with a summary of the state of the voltage margins. In this example no voltage margins were enabled.

```
VOLTAGE MARGINS ARE NOW:
MF20:  NOMINAL FOR ALL MOS CONTROLLERS

CORRECTABLE MEMORY ERROR
TN    PROG     PC     AS    PAT     ADDRESS        DATA      BIT #
5    032320 000006   SEQ   MWC2   00627526   043075 523056    40
5    032322 000004   SEQ   MWC2   00627526   043075 523056    40
5    032327 000005   BLT   MWC2   00627526   043075 523056    40
5    032331 000004   BLT   MWC2   00627526   043075 523056    40
5    032320 000006   SEQ   MWC4   00627526   420442 104124    40
5    032322 000006   SEQ   MWC4   00627526   420442 104124    40
5    032327 000005   BLT   MWC4   00627526   420442 104124    40
5    032331 000004   BLT   MWC4   00627526   420442 104124    40
TYPEOUT LIMIT REACHED IN CONTROLLER # 10  , BIT SUB RAM LOAD ADDRESS: 0360

DATA BIT FAILURES
BIT          PICKUP    DROPOUT   SINGLE BIT
40              0          0         24

ERRORS PER RAM
CONTROLLER     SLOT  SERIAL        E#/ERRORS

10             16    7460215       8/24
```

Example 4 - This example illustrates how to select the switch options such that only the address test is run and only on one specific 32K segment of memory.

```
>. SW402037
>. STM
>.
PDP-10 KL10 4096K MEMORY DIAGNOSTIC (DFMMH)
VERSION 0.3, SV=0.13, CPU#=1030, MCV=156, MCO=100, HO=36, 60HZ

SWITCHES = 100000 402037
CLK SOURCE = EXTERN, CLK RATE = FULL, AC BLK 6 , CACHE: 0

MEMORY MAP =
FROM      TO          SIZE/K
00000000 00777777        256


ADDRESS TEST, PASS COUNT  0
ERROR TOTALS : NONE


TEST COMPLETION, PASS COUNT  0
ERROR TOTALS : NONE

END PASS 1.

ADDRESS TEST, PASS COUNT  1
ERROR TOTALS : NONE


TEST COMPLETION, PASS COUNT  1
ERROR TOTALS : NONE


ADDRESS TEST, PASS COUNT  2
ERROR TOTALS : NONE


TEST COMPLETION, PASS COUNT  2
ERROR TOTALS : NONE


ADDRESS TEST, PASS COUNT  3
ERROR TOTALS : NONE


TEST COMPLETION, PASS COUNT  3
ERROR TOTALS : NONE

^C
```

7.2.3.3 SYSERR

All the diagnostics described to this point are executive mode programs that require using the system in stand-alone mode. When this is not possible, the service engineer can gather maintenance data relative to the MF20's performance by running one of two user mode programs: SYSERR and TGHA. These two programs should prove to be the most often used and most helpful tools in the service engineer's toolbox.

During any scheduled PM the service engineer should run SYSERR to obtain a summary of all the memory errors that have accumulated in the system's error file, ERROR.SYS. Refer to the TOPS-10 and TOPS-20 SYSERR Manual (AA-D533A-TK) for detailed information on how to run SYSERR and how to interpret the printouts related to the MF20 subsystem. The following examples illustrate the typical use of SYSERR and give samples of the error information supplied.

Example 1 - This example illustrates a read parity error detected in MF20 controller no. 11. The failing address was 2777000_8.

```
PROCESSOR PARITY INTERRUPT
 LOGGED ON Fri 13 Oct 78 15:23:56     MONITOR UPTIME WAS    0:09:15
        DETECTED ON SYSTEM # 2136.
        RECORD SEQUENCE NUMBER: 102.

        CONI APR:          7740,4413 = SBUS ERR,MB PAR ERR,
        ERA:               2002,777000 = WD #0 MEMORY READ
        BASE PHY. MEM ADDR.
         AT FAILURE:       2777000

        PC FLAGS AT INTERRUPT:      304000000000
        PC AT INTERRUPT:            1037554
        # ERRORS ON THIS SWEEP      1.
        LOGICAL AND OF
        BAD ADDRESSES:    2,777000
        LOGICAL OR OF
        BAD ADDRESSES:    2,777000
        LOGICAL AND OF
        BAD DATA:         0,0
        LOGICAL OR OF
        BAD DATA:         0,0
SYSTEM MEMORY CONFIGURATION:

CONTROLLER:  #10  MF20
F0:     7740,4160          F1:          500,1000
LAST WORD REQUEST:         RQ0RQ1RQ2RQ3- READ
LAST ADDRESS HELD:         4160
CONTROLLER STATUS:          SF2 & SF1= 2
        ERRORS DETECTED:               NONE
CONTROLLER:  #11  MF20
F0:     47042,777000       F1:          500,1000
LAST WORD REQUEST:         RQ0- READ
LAST ADDRESS HELD:         2777000
CONTROLLER STATUS:          SF2 & SF1= 2
```

```
        ERRORS DETECTED:            READ PARITY
ERRORS DETECTED DURING SWEEP:
ADDRESS BAD DATA          GOOD DATA     DIFFERENCE
2777000 0,0  GOOD DATA NOT FOUND
```

Example 2 - This example shows the error response when SYSERR detected a control error in MF20 no. 10.

NOTE
SYSERR does not execute a diagnostic function 2 to record the specific RAM that caused the control error.

```
PROCESSOR PARITY INTERRUPT
 LOGGED ON Thu 24 Aug 78 16:26:46      MONITOR UPTIME WAS  0:08:58
        DETECTED ON SYSTEM # 2136.
        RECORD SEQUENCE NUMBER: 4414.

        CONI APR:        7740,4013 = SBUS ERR,
        ERA:             602000,3644 = WD #3 MEMORY READ
        BASE PHY. MEM ADDR.
         AT FAILURE:     3644

        PC FLAGS AT INTERRUPT:     14000000000
        PC AT INTERRUPT:           2006
        # ERRORS ON THIS SWEEP     0.
        LOGICAL AND OF
        BAD ADDRESSES:  777777,777777
        LOGICAL OR OF
        BAD ADDRESSES:  0,0
        LOGICAL AND OF
        BAD DATA:       777777,777777
        LOGICAL OR OF
        BAD DATA:       0,0
SYSTEM MEMORY CONFIGURATION:

CONTROLLER:  #0  MA20   64 K
F0:     6000,0  F1:      36100,16012
        INTERLEAVE MODE:           4-WAY
        REG ENABLED:    0 2
        LOWER ADDRESS BOUNDARY:    0
        UPPER ADDRESS BOUNDARY:    377777
        ERRORS DETECTED:           NONE
CONTROLLER:  #1  MA20   64 K
F0:     6000,0  F1:      36100,16005
        INTERLEAVE MODE:           4-WAY
        REG ENABLED:    1 3
        LOWER ADDRESS BOUNDARY:    0
        UPPER ADDRESS BOUNDARY:    377777
        ERRORS DETECTED:           NONE
CONTROLLER:  #10  MF20
F0:     407740,736010   F1:        500,1000
LAST WORD REQUEST:      RQ0RQ1RQ2RQ3- READ
LAST ADDRESS HELD:      736010
CONTROLLER STATUS:       SF2 & SF1= 2
        ERRORS DETECTED:           CONTR ERR
CONTROLLER:  #14  MF20
F0:     7742,777000     F1:        500,1000
LAST WORD REQUEST:      RQ0RQ1RQ2RQ3- READ
```

```
LAST ADDRESS HELD:       2777000
CONTROLLER STATUS:        SF2 & SF1= 2
        ERRORS DETECTED:           NONE
ERRORS DETECTED DURING SWEEP:
ADDRESS BAD DATA         GOOD DATA      DIFFERENCE
```

7.2.3.4 TGHA

Overview - TGHA is a user mode program that runs under control of the TOPS-20 monitor during normal timesharing. Its primary functions are twofold:

1. To enhance the reliablity of the MF20 subsystem by dynamically taking corrective action to remove a failing component (bit substitution or patching)

2. To provide maintenance information to Field Service and Manufacturing necessary for the replacement and repair of the M8579 MOS storage array modules.

To perform these functions, TGHA creates and maintains an MF20 history file (TGHA.DAT) that contains historical records for each MOS array board as well as a cumulative MF20 error history file (TGHA.TRA) and configuration map. TGHA also has the capability of writing various types of ASCII error messages into the ERROR.SYS file. These messages will warn the user of impending MF20 failures and give instructions on running TGHA to obtain detailed reports.

CAUTION

Because TGHA has some privileges which are rarely found outside of the monitor, any attempt to stop TGHA during one of its operations could cause a system crash.

TGHA Operation Modes - TGHA has two modes of operation, automatic and manual.

Initially these two modes are distinguished internally by the job number under which TGHA is running. TGHA will enter automatic mode if it is job 0. This will only occur at monitor start-up and on the occurrence of a predetermined number of MF20 data read errors. This number is currently set to 1, but it may be increased at some future time when more data is available. TGHA may be run manually only by a privileged user.

Trace File - The trace file, called TGHA.TRA, contains dated entries indicating what corrective action has been taken by TGHA. If the TGHA software becomes confused or a data base becomes full, entries made in the trace file can be used to analyze the problem. Each trace file entry contains the following information.

1. Date
2. Time
3. Brief Description of TGHA action

The following example illustrates some typical trace file entries.

```
**********
9-Nov-78 16:18:25
```

```
*PARITY ERROR AT ADDRESS 1603276, BLOCK 7
*STORAGE MODULE SERIAL NUMBERS BY FIELD:
0 = 8320050 1 = 8320011 2 = 8320022 3 = 8320035
**********
```

11-Oct-78 15:50:12 - IN CONTROLLER 10, GROUP 0, VECTOR 3, THE SPARE BIT IS GOING TO BE CHANGED TO 124 (OCTAL).

11-Oct-78 15:50:12 - BIT SUBSTITUTION - EVERTHING COMPLETED NORMALLY

11-Oct-78 15:50:13 - IN CONTROLLER 10, GROUP, VECTOR 3, THE ICE BIT IS GOING TO BE LEFT SET.

11-Oct-78 15:50:13 - IN CONTROLLER 10, GROUP 0, VECTOR 3, THE ICE BIT IS GOING TO BE LEFT SET.

11-Oct-78 15:55:38 - IN CONTROLLER 11, GROUP 0, VECTOR 30, THE SPARE BIT IS GOING TO BE CHANGED TO 220 (OCTAL).

11-Oct-78 15:55:39 - BIT SUBSTITUTION - EVERYTHING COMPLETED NORMALLY

The following four examples illustrate the information printed on sample pages from an actual TGHA history file.

Example 1 - This is the first entry and the first page of the file. It shows that there were two MF20s with 256K of MOS on each (two groups). It also lists the serial numbers of all the M8579 storage modules.

```
TGHA HISTORY FILE DUMP AS OF: 11-Oct-78 15:56:03 FOR CPU #2136
HISTORY FILE DATA BASE VERSION #1

CONTROLLER    GROUP NUMBER    STORAGE MODULE SERIAL NUMBERS
10            0               7460209 7460108 7460117 7460200
              1               7460205 7460101 7460215 7460202
              2

11            0               7460118 7460105 7460116 7460113
              1               7460119 7460115 7460104 7460213
              2
```

Example 2 - This shows the page for the M8579 module with the serial number 7460209. No errors were detected that pointed to this module.

```
STORAGE MODULE SERIAL NUMBER : 7460209
THE LAST TIME THIS STORAGE MODULE WAS USED WAS 11-Oct-78 15:56:03
NO ERRORS LOGGED IN THE HISTORY FILE FOR THIS MODULE
```

Example 3 - This example indicates that the spare bit was

substituted for bit 21 (124 in location 3). This is the upper 8K segment of block 1.

```
GROUP DATA
CONTROLLER : 10 GROUP : 0

THE LAST TIME THIS GROUP WAS USED WAS 11-Oct-78 15:55:38
THE LAST TIME A CHANGE WAS MADE TO THIS GROUP WAS 11-Oct-78
15:50:13
BIT SUBSTITUTION MAP:

000/ 255 255 255 124 255 255 255 256 255 255 255 255 255 255 255
255
020/ 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255

PAGES THAT ARE BAD:
```

Example 4 - This example shows that the MOS chip in location E76 on the M8579 with serial no. 7460200 caused two errors on October 11, 1978. The error type indicates whether it was repeatable or not. Soft is not repeatable; hard is repeatable.

```
STORAGE MODULE SERIAL NUMBER : 7460200
THE LAST TIME THIS STORAGE MODULE WAS USED WAS 11-Oct-78 15:56:03
SEQUENTIAL ERRORS:
BLOCK   SUBBLOCK   BIFN   ROW   COLUMN   EX   TYPE   TIME
0       3          3      142   10       76   HARD   11-Oct-78 15:50:12
0       3          3      143   17       76   HARD   11-Oct-78 15:50:13
```

NOTE

It is important that the service engineer return this page of the printout along with the faulty M8579 when shipping modules back to a module repair facility.

Operating Procedures - The initial release of TGHA with TOPS-20 Version 3A or 4 supports a minimum subset of possible commands that will permit the service engineer to obtain listings of the history or trace files to analyze the performance of the MF20 subsystem. To run TGHA manually, the user must have the minimum of maintenance capabilities and after logging on, he must enable these privileges. When run in manual mode as opposed to being started as a job by the monitor, TGHA does not automatically start looking for MF20 errors.

After logging onto the system as a privileged user, the service engineer enables the privileges by typing:

```
@ ENABLE<CR>
```

Where: @ is the monitor prompt, and <CR> is carriage return.

The next step is to run TGHA by typing:

```
@R TGHA<CR>
```

After TGHA is started it types a prompt of:

```
TGHA>
```

indicating it is ready to accept a command. If HELP is typed, the following text will be typed on the user's terminal.

```
TGHA HELP - COMMAND      PURPOSE

            EXIT         EXIT FROM TGHA
            HELP         TYPE THIS TEXT
            HISTORY      DUMP HISTORY FILE
            TRACE        DUMP TRACE FILE
```

Dumping the history or trace files will invoke TGHA to process TGHA.DAT or TGHA.TRA and create two new files called HISTORY.LST or TRACE.LST in the user's disk space. Once these files have been created, they can be dumped on the line printer as follows. The example shows the complete dialogue.

```
@ENABLE
@R TGHA
TGHA>HISTORY<CR>
WRITING HISTORY.LST
TGHA>TRACE<CR>
WRITING TRACE.LST
TGHA>EXIT<CR>
@PRI HISTORY.LST,TRACE.LST<CR>
@LOGOUT<CR>
```

At this point the service engineer can retrieve the listings, analyze the error data, if any, and make the appropriate decision relative to scheduling corrective maintenance.

MF20 History Files - Almost everything that TGHA does is dependent on the error history file. The data in the file is collected over a long period of time in the normal working environment of the system; therefore, the error data contained in it should accurately reflect the actual hardware failures. In essence, monitor and user jobs are the ultimate test pattern generators. TGHA relies on the accuracy of this data to perform its corrective actions. The file will be named TGHA.DAT and will be kept in the same directory as TGHA. The updating processes can occur whenever TGHA is run in automatic mode.

When TGHA is run it will create the error history file if the file is not found. There is one exception: if TGHA fails to find the file when run at monitor start-up, it will not create the file at that time because of the possibly incomplete initialization of the file system. If this condition arises, it will treat memory as

though there are no known errors.

The file has one page (512 words) for CPU identification(s), controller-related error history, and array board directory. Since the MF20 control boards do not have serial numbers, TGHA will clear the controller-related error history whenever the CPU identification(s) change.

The file has one page for each of the MOS storage array boards. These pages each contain the board's serial number, multiplexer chip-related error data, and MOS RAM-related error data. Since these boards do have serial numbers, TGHA can track their swapping, removal, and replacement. When a new board appears, TGHA will create a new page for it and assume that the board is error-free. When a board is removed, TGHA will wait 10 days before deleting that board's page.

7.3 TROUBLESHOOTING PROCEDURES

7.3.1 Introduction

The following paragraphs discuss troubleshooting procedures to aid the service engineer in testing and repairing the MF20 MOS storage subsystem. The discussions assume that a single fault exists within the MF20 and the repair philosophy calls for isolating the fault to the field replaceable unit (module, oscillator, cable, etc.). The procedures are general and designed to provide the service engineer with suggestions for parts to check; they are not intended as rigid or guaranteed formulas.

7.3.2 Memory Fault Symptoms

This paragraph discusses the most common symptoms observed if the MF20 fails to respond in accordance with its design specifications. Numerous logic checks have been designed into the MF20 to detect most faults that could occur. Since the memory has no visible error indicators, the user must retrieve the error information using KLDCP, the 11-based diagnostic control program.

> **NOTE**
>
> **The service engineer must constantly consider the hardware/software interdependence in the MF20 and recognize that the control RAMs in the MF20 must be properly configured to ensure proper response. (See ICN command in MEMCON, Table 7-1).**

In most cases the basic diagnostics, DHKBF and DHKBG, will detect the malfunction and report the specific error conditions detected. Refer to Table 7-9 for a list of the primary error indicators.

To retrieve the APR status and display it on the console, using KLDCP, type:

```
>. EX  700240 0 <CR>
>. EM  0 <CR>
```

(KLDCP will type out status in octal.)

To retrieve the ERA status from the EBox and display it on the console, type:

```
>. EX  700400 0 <CR>
>. EM  0 <CR>
```

(KLDCP will type out error status in octal.)

To retrieve the MF20 error flags via SBDIAGs 00, 02, and 10, type:

```
>. DM  0:  NN0000  XX <CR>
>. EX  700500 0 <CR>
>. EM  1 <CR>
```

(KLDCP will type out SBDIAG word E + 1.)

NOTE

NN = MF20 controller no.; XX = SBDIAG no. 00, 02, or 10

To retrieve the MF20 error flags via an SBDIAG FCN06, type:

```
>. DM  0:  NN0000  4XX <CR>
>. EX  700500 0 <CR>
>. EM  1 <CR>
```

(KLDCP will type out SBDIAG error status E + 1.)

XBUS Error - This flag indicates the MF20 detected an error during a memory reference. The next step is to execute SBDIAG functions 00, 02, 10, and 06 to determine the specific error type.

NXM - This indicates that the MF20 failed to assert ACKN in response to a memory read/write request. Three signals to key on if this occurs are as follows.

```
CTL1 RAS START 0 A H
SYN7 BOX SELECT L
ADT7 BOX SELECT EN IN L
```

Possible causes for the failure are as follows.

1. Improperly seated XBus cable connector
2. Address response RAM - M8575
3. Failure of software to enable MF20 and properly configure address response RAM

Table 7-9 MF20 Primary Error Flags

Instruction	Bit	Error Indication
CONI APR	24	XBus Error
700240	25	NXM
	27	MB PAR
	29	ADR PAR ERR
RDERA	0-1	Word No.
700400	3	Channel Reference
	4-5	00 Channel Store Status 01 Channel Store Data 10 EBox Store 11 Cache Writeback
	6	= 0 Read = 1 Write
	14-35	Physical address of first word of transfer
SBDIAG 00	0	Control RAM Error
	1	Correctable Error
	2	Incomplete Cycle
	3	Read Parity Error
	4	Write Parity Error
	5	Address Parity Error
SBDIAG 02	28	Timing RAM Parity Error
	29	Bit Substitution RAM Parity Error
	30	Address Response RAM Parity Error
DBDIAG 10	27	DC Bad
SBDIAG 06	14	Double Bit Error

4. Improperly set up address jumpers on the backplane (Refer to Table 3-1).

MB PAR - The MBox sets this flag if it detects incorrect parity in its memory buffer register when reading or writing memory. The next step is to execute a RDERA to gather more information about the error. If SBUS ERROR is asserted, the fault is probably in the MF20 data path prior to the bus drivers. If SBUS ERROR is not asserted, the fault could be an XBus driver or a faulty XBus cable connection.

Control RAM - This flag indicates parity.

Parity Error - An error was detected during a read/write access when the MF20 tried to read one of its control RAMs. The next step is to execute an SBus diagnostic function 2 to determine the specific RAM causing the error. If the error is in the timing RAM, replace M8577; if in the spare bit substitution RAM, replace M8574; and if in the address response RAM, replace M8575.

Correctable Error - This flag indicates that the data read from the MOS had a single bit in error and was correctable. This error was probably caused by a faulty storage module if local to a specific group, block, and bit position number. If the error is not local to a range of specific addresses, replace the M8574 and M8575 data path modules.

Incomplete Cycle - This flag indicates a failure to complete the proper timing RAM sequence. The problem is probably in the M8577 module.

Parity Errors - Read, write, and address parity errors are the three most common errors detected. Corrective action is as follows.

Address Parity - Replace the M8576 and M8577 modules.

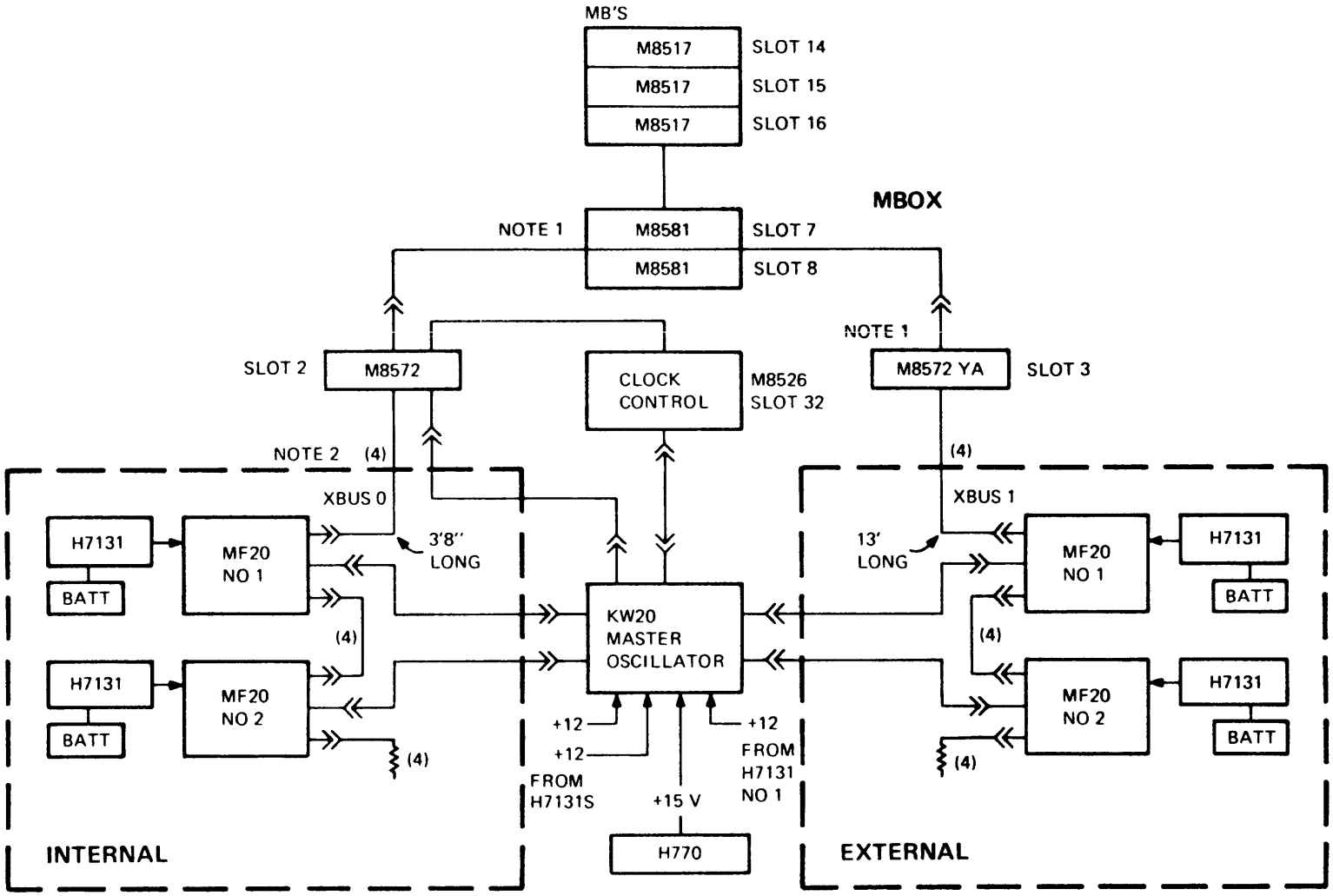
Read/Write Parity - If the errors affect all of memory, replace the M8574 and M8575 modules. If they affect only one group or block of addresses, replace the specified set of M8579 modules.

DC Bad - Check the power supply

Double-Bit Errors - Use the same approach as for parity errors.

7.3.3 General Troubleshooting Approach

This paragraph outlines a shotgun approach to MF20 corrective maintenance. It includes a troubleshooting block diagram that shows all the field replaceable components that comprise the MF20 MOS storage subsystem and how they are interconnected (Figure 7-4). A simplified troubleshooting flowchart is also included to guide the beginning technician through a set of module replacement procedures for repairing MF20 faults (Figure 7-5).



NOTES:

1. FOR CORE/MOS MIX CONFIGURATION THE M8581 MODULES BECOME M8580 MODULES AND SLOT 3 RECEIVES AN M9006 SBUS CONNECTOR.
2. WHEREVER THE NOTATION (4) IS USED IT INDICATES A SET OF FOUR XBUS CABLES OR TERMINATOR MODULES.

MR-3467

Figure 7-4 Troubleshooting Block Diagram

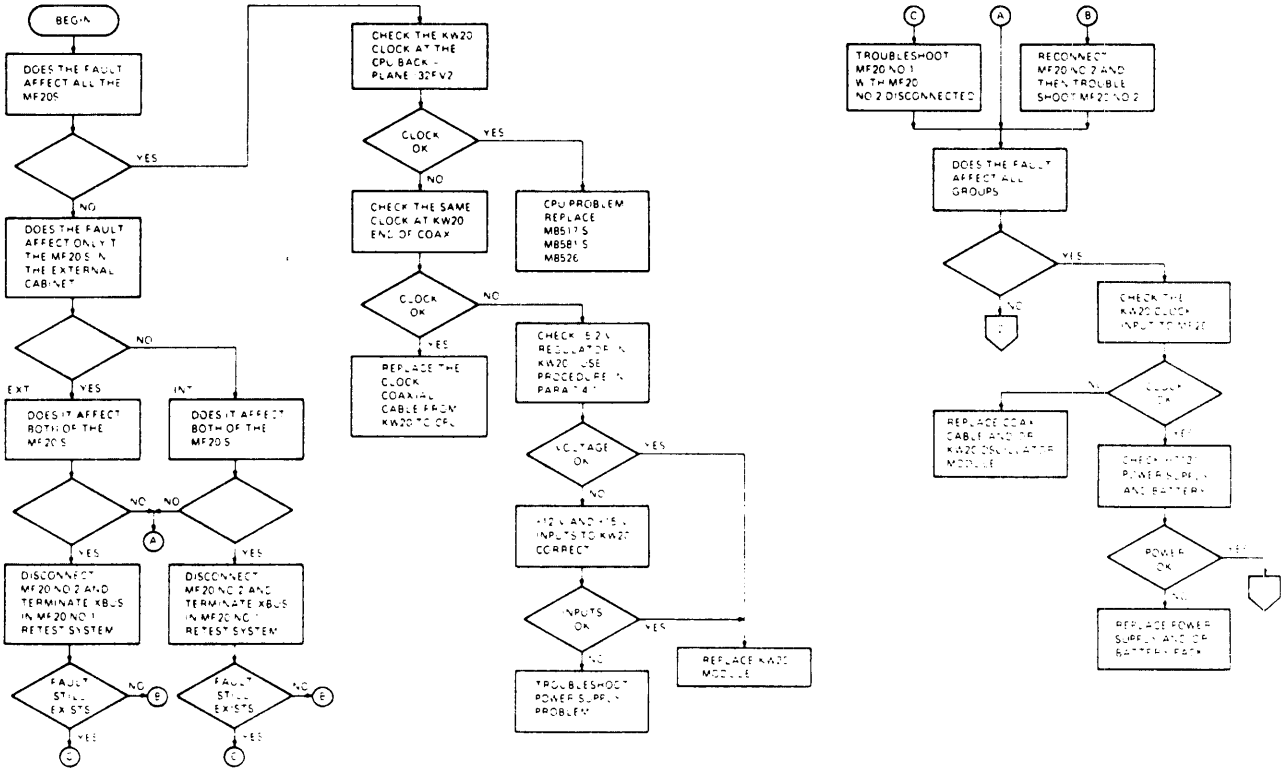


Figure 7-5 MF20 Troubleshooting Flowchart
(Sheet 1 of 2)

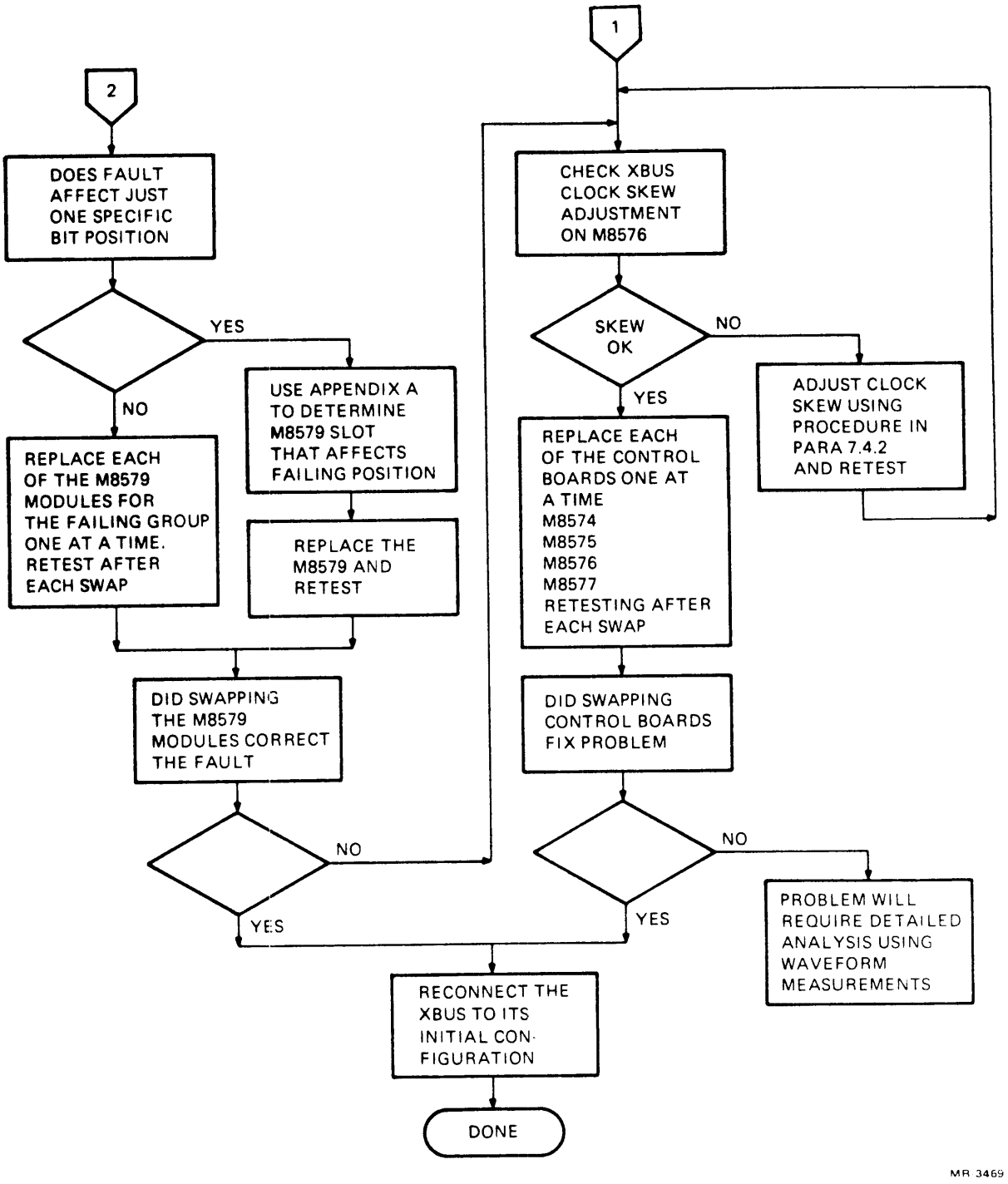


Figure 7-5 MF20 Troubleshooting Flowchart (Sheet 2 of 2)

The technique recommended assumes that the problem is being caused by a single fault and that the fault is contained within the boundaries of a field replaceable unit (FRU); i.e., module, power supply, cable, terminator, etc.

The following discussion outlines a basic approach to isolate MF20 problems. It assumes that the service engineer has run or attempted to run DHKBF, DHKBG, MEMCON, and DFMMH and has gathered several error responses. Isolating the fault depends on the ability to analyze and relate these responses to specific functional areas within the MF20 subsystem. The following question and answer session summarizes the approach to fault isolation.

Q1. Does the fault affect both MF20s in a single cabinet?

A1. If it does, then the fault must be either the KW20 master oscillator or the XBus cabling or connector to the KL10 CPU. Check for oscillator clocks at the coaxial cable connectors on both MF20 backplanes. Check seating of the M8572 module in the CPU and the XBus Berg connectors on the MF20 backplanes. If it affects only one MF20, check the separate coaxial cable that distributes the clock to the MF20's backplane. Also check all the power supply voltages to the faulty MF20's backplane and ensure that the dc power LED is lit on the backplane.

Q2. Does the fault affect all groups (set of four M8579s) in a single MF20?

A2. If it does and the oscillator clock and power supply are functioning correctly, then the fault is probably in one of the four control boards.

Q3. Does the fault point to port loopback failing?

A3. If yes, replace the M8574 module in slot 4.

Q4. Does the fault point to a particular control RAM failing?

A4. If yes, replace the appropriate module as follows.

Timing RAM	M8577
Spare Bit Substitution RAM	M8574
Address Response RAM	M8575
Fixed Value RAM	M8576

Q5. Does the fault point to a particular bit position in group loopback mode?

A5. If it does, the problem is probably in one of the data path modules, the M8574 or the M8575 in slots 4 and 6 respectively.

Q6. Does the fault point to a particular bit position within a particular block when using the actual RAM chips on the M8579 boards?

A6. If so, the fault is probably a malfunctioning M8579. Use DFMMH, data patterns test, to obtain chip callouts to isolate the faulty M8579 module.

Q7. Does fault indicate it affects the spare bit substitution logic during a write?

A7. If yes, problem is probably on the M8574 module.

Q8. Does fault indicate it affects the spare bit substitution logic during a read?

A8. If it does, then replace the M8575 module.

Q9. Does the fault point to a defective ECC generator?

A9. If yes, then replace the M8574 module.

Q10. Does the fault point to the ECC checking logic?

A10. If yes, then replace the M8575 module.

Q11. Does the fault appear to be a MOS address generation problem that affects all groups?

A11. If so, then replace the M8577 module.

Q12. Does the fault affect the single-cycle tests in DHKBF (test 9)?

A12. If so, the fault is probably the M8576 board, but could also be on the M8577.

Q13. Does the fault point to a problem with the refresh logic?

A13. If yes, replace the M8577 module.

There are many other possible questions that one could ask depending on the actual symptom. This list does not pretend to be all-inclusive and is intended only to give the service engineer who is new to the MF20 some help in getting started with troubleshooting.

7.3.4 Using the Diagnostics

Since the MF20 is so heavily dependent on the software for its identity and control, it is extremely important that the service engineer learn how to use the available diagnostic software to check out and repair the hardware. Depending on individual preference and specific circumstances surrounding the failure, the service engineer could employ one of two troubleshooting techniques: a "tops-down" approach or a "bottoms-up" approach. The following paragraphs discuss how to use the MF20 diagnostic software for both approaches. Refer to Figures 7-6 and 7-7 for a summary of the two approaches. The numbers in parentheses in the following discussion correlate with the circled numbers in the figures.

7.3.4.1 Tops-Down Approach - The tops-down approach will probably be the most common technique. To isolate the problem to a MOS storage subsystem (1), begin by running SYSERR to gather error summary information from the ERROR.SYS file using the /ALLSUM switch. This summary should flag serious or impending degradation of the (core-MOS mix) memory system and point to the MF20 subsystem as the problem.

Next, isolate the problem to a particular MF20 or storage board (M8579) (2). Log onto the system as a privileged user and run TGHA using the HISTORY and TRACE commands to obtain a listing of the pertinent MF20 error history. Proper analysis of this data should provide the following information.

1. Which MF20 is faulty
2. Which group or groups of storage boards are faulty
3. Whether it is a control or storage board
4. If it is a storage board (M8579), which one and which MOS chips on that board

If the fault has not been isolated by this time, it should be traced to the module or functional logic level (3, 4). The system is put into stand-alone mode, and MEMCON is loaded from the KLAD-pack and used to configure the memory. Then the 10-based 10 diagnostic, DFMMH, is run to perform rigorous data and address pattern testing of the MF20 to aggravate the fault to the point where solid error reports may be generated. Voltage margining could be invoked prior to running DFMMH to further aggravate the problem. The error information furnished by DFMMH should identify the failing assembly if the fault is due to a faulty storage module, or at least point to some major functional area on the control boards. If the fault has still not been detected or identified at this point, the basic 11-based 10 diagnostics, DHKBG and DHKBF, should be run to isolate the defective subassembly (5). These two programs will furnish detailed logic signal callouts which should permit identifying which modules to replace. Error

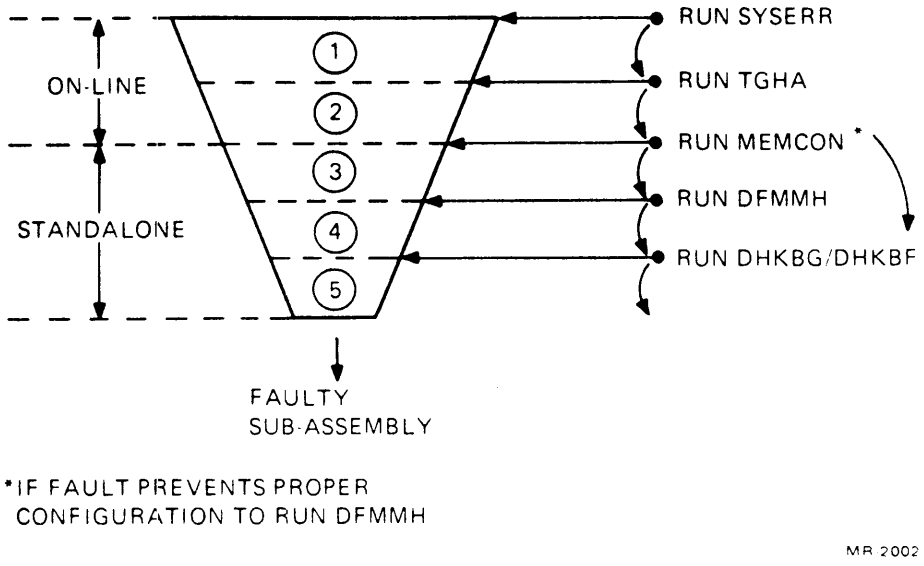


Figure 7-6 MF20 Troubleshooting Tops-Down Approach

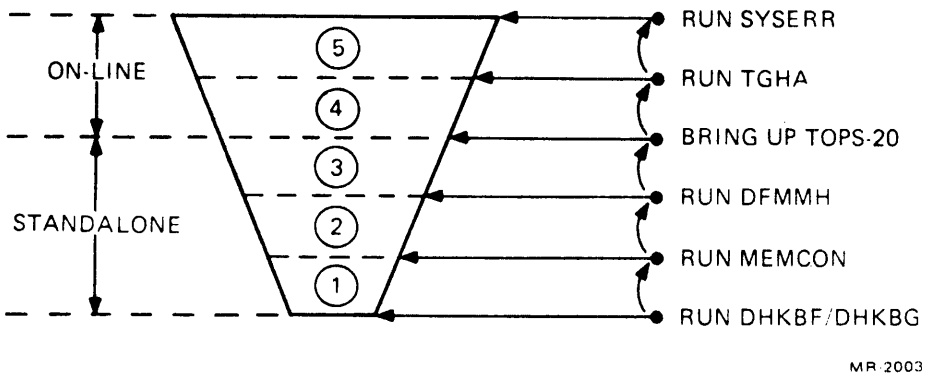


Figure 7-7 MF20 Troubleshooting Bottoms-Up Approach

messages furnish a failing test number and program counter (PC) location that points to the failing point in the program listing. The listings include comments that relate the test failure to specific areas of logic within the MF20. The comments include instructions for modifying the test code to establish tight scope loops for isolating a fault to the circuit level. Normal field maintenance should seldom require such detailed fault isolation.

7.3.4.2 Bottoms-Up Approach - The bottoms-up approach reverses the process and starts by running DHKBF followed by DHKBG to verify the MF20 control logic and some of the storage board logic.

After verifying the control, MEMCON is run to configure the memory prior to running DFMMH.

If at this point no faults are detected, TOPS-20 is loaded and started and TGHA is run to prepare the MF20 for on-line operation. Once on-line TGHA and SYSERR could be run to gather error statistics accumulated as the system runs timesharing.

Either of these approaches is effective and in most cases the service engineer will use a combination of the two. The tops-down approach is preferred because it is possible to identify and isolate the fault on-line and schedule deferred maintenance to replace the faulty module during the next PM. This, of course, depends on how seriously the failure is degrading system performance. The bottoms-up approach has the advantage of identifying the fault at the simplest operational level, although it requires taking down the system. For more detailed information on the 11-based diagnostics, DHKBF and DHKBG, refer to Appendix C.

7.4 SUBASSEMBLY REMOVAL AND REPLACEMENT

Most MF20 faults will be confined within the boundaries of a field-replaceable subassembly. This paragraph describes how to remove and replace the major field replaceable units in the MF20 subsystem.

7.4.1 Power Supply

NOTE

The MF20 power supply assembly weighs over 60 pounds. Two persons are required to remove and replace the unit.

7.4.1.1 Removal Procedure

1. Remove all power from the system and unplug the ac power cable to the MF20 power supply.

2. Remove all access and protective covers from the front of the power supply to permit access to the terminal boards.

3. Disconnect the margin cable connector from J1.

4. Place the battery pack circuit breakers in the OFF position and disconnect the battery cable from J4.

5. Disconnect the remote turn-on cables from J2 and J3 (if used).

NOTE

If a jumper connector is used in J2 or J3, be sure to note where and install it in exactly the same position in the new supply.

6. Remove the bakelite covers from TB1 and TB2 and disconnect the main power and fan cable connections.

7. Remove all the margin cable connections to TB3, 4, 5, and 6.

8. Remove all the main cable harness connections from the terminal posts in the middle of the unit.

NOTE

Replace the washer and nuts on the terminal posts after removal of the wires.

At this point the power supply should be free of all wire connections to the system. To remove the unit, simply remove the eight Phillips-head screws that mount the unit to the system frame.

CAUTION

Before removing the eight Phillips screws, insert some form of support under the unit to prevent if from falling when the screws are removed (a stack of used printer paper would serve the purpose).

9. Remove the four Phillips-head screws on the bottom on each side of the unit first.

10. Next, remove the two Phillips-head screws on the top on each side of the unit.

11. Remove the power supply unit.

7.4.1.2 Replacement Procedure

1. Remove the new power supply from its shipping container. Save the container for returning the defective power supply.

2. Mount the new power supply in the system using the eight

Phillips screws removed in steps 8 and 9 of the removal procedure.

3. Tighten all eight screws to prevent vibration of the unit.

4. Reconnect all the cables, connectors, and wires removed in steps 3 through 7 of the removal procedures.

CAUTION

Use a torque wrench to tighten down the nuts on the power output terminal posts. If this is not done correctly the threads on the terminal posts could be damaged (brass on brass) resulting in faulty power connections to the backplane. The correct torque is 28 in/lb.

5. Replace all safety and access covers.

6. Plug in the power connector to the KL10 system.

7. Refer to Chapter 3. Follow the prepower checkout procedures before attempting to place the MF20 on-line.

8. Run DHKBF using the /QV switch to verify that the power supply margins operate properly.

9. Finally, run DHKBF, DHKBG, MEMCON, and DFMMH to verify proper operation of the MF20.

7.4.2 Battery Pack

7.4.2.1 Removal Procedure

1. Remove all power from the system.

2. Place the battery pack circuit breakers in the OFF position.

3. Place the MF20 power supply breaker in the OFF position.

4. Disconnect the cable connector from J4.

5. Remove the six Phillips-head screws (from the rear) that hold the battery pack to the system frame.

6. Remove the pack.

7.4.2.2 Replacement Procedure

1. Remove the new battery from its shipping container. Save the container for returning the defective unit.

2. Mount the new pack in the system frame using the six Phillips-head screws removed in step 5 of the removal procedure.

3. Place the battery pack circuit breakers in the OFF position.

4. Connect the battery cable to J4 on the MF20 power supply.

5. Place the battery pack circuit breakers to the ON position.

6. Power up and check the MF20 subsystem by running the following diagnostics.

   DHKBF
   DHKBG
   MEMCON
   DFMMH

7. If no errors are reported, the system should be ready for on-line operation.

7.4.3 KW20 Master Oscillator

7.4.3.1 KW20 Fan Removal Procedure

1. Remove all power from the system.

2. Disconnect the input connector at the top of the fan (red/white wires).

3. Using a small offset Phillips screwdriver, remove the four Phillips-head screws used to mount the fan.

4. Remove the fan assembly.

7.4.3.2 KW20 Fan Replacement Procedure

1. Remove the new fan from its shipping container.

2. Mount the fan with the power connector in the upper right corner using the four Phillips-head screws.

   NOTE
   The airflow decal should indicate that the air flow is toward the oscillator module.

3. Reconnect the power connector.

4. Return the system to the user.

7.4.3.3 KW20 Logic Module Removal Procedure

1. Power down the system.

2. Remove the plastic shield from the front of the KW20 by removing two small Phillips-head screws.

3. Disconnect the following cables.

   a. Power cable to J8

   b. Control cable to J10.

   c. Clock coaxial cables from J1 to J7.

   NOTE
   Mark coaxial clock cables so that they will connect to the same connectors on the new module.

4. Loosen the four mounting screws (knurled silver heads) that hold the unit to the system frame, and remove the entire unit from the system.

5. Now release the black metal enclosure core by turning the six black latching screws one-half turn counter-clockwise.

6. Remove access cover.

7. Remove the six Phillips-head screws that hold the module to the box.

8. Remove the module.

7.4.3.4 Logic Module Replacement Procedure

1. Remove the new module from its shipping container. Save the container to return the defective module.

2. Mount the module in the box (power connector at the opposite end from the fan) using the six Phillips-head screws.

3. Replace the black metal cover and latch it down by turning the six screws one-half turn clockwise.

4. Mount the KW20 back into the system. Tighten down the four knurled silver screws.

5. Reconnect all the cables removed in step 3 of the removal procedure.

6. Replace the plastic shield removed in step 2 of the

removal procedure.

7. Power up the system and run all the diagnostics to check out the MF20.

7.4.4 MF20 Logic Assembly

7.4.4.1 Logic Module Removal Procedure

CAUTION

The modules used in the MF20 are extended-width hex-height boards. They should be handled with extreme care to prevent damage. Dropping one of these modules could cause expensive damage.

Before handling the M8579 storage modules, discharge any static charge on your body by touching the system frame. The MOS chips could be damaged by the voltages of static discharge.

1. Remove all power to the MF20 backplane.

2. Open the access door by turning the knob counterclockwise until the door releases.

3. Pull out on the module locking levers (up on the top ones. Down on the bottom ones) to release the module form its socket.

4. Remove the module, being sure to have a firm grip to prevent dropping it on the floor.

5. Place the module in a safe place to prevent its being damaged.

7.4.4.2 Logic Module Replacement Procedure

1. Remove the new module from its container. Save the container for returning the defective modules.

2. Mount the new module in the logic assembly by sliding it into the appropriate slot guides.

3. Lock the module into the backplane using the module locking levers (top lever-press down, bottom lever-press up). Close the logic assembly door and tighten down door knob clockwise.

4. Power up the system and run the appropriate diagnostic to verify that the fault was repaired.

7.4.4.3 Fan Removal Procedure - Each MF20 logic assembly has two

fans mounted at the bottom of the unit. Removal procedure is as folows.

1. Remove all system power.

2. Remove the ac power connector from the fan to be removed.

3. Using a small offset Phillips screw-driver, remove the four screws that hold the fan to the bottom of the MF20.

4. Remove the fan.

7.4.4.4 Fan Replacement Procedure

1. Remove the fan from its shipping container.

2. Mount it in the logic assembly using the four Phillips-head screws removed in step 3 of the removal procedure.

3. Reconnect the power connector to the fan (red/white wires).

4. Power up the system and run the diagnostics to verify proper operation.

7.4.4.5 Air Filter Removal/Replacement Procedure

1. Grasp the plastic tab on the end of the filter and pull out at a 45 degree angle to remove the filter. [Filter is accessable from the rear of the logic assembly (module side)].

2. Inspect the filter and clean if required using a warm solution of soap and water.

3. After drying the filter completely, replace it by reversing the procedure described in step 1 above.

7.5 MF20 ADJUSTMENTS

There are no mechanical and only two electrical adjustments in the MF20 MOS memory subsystem. This paragraph describes how to perform the KW20 regulator adjustment and the MF20 clock deskew adjustment (one on each M8576 module).

7.5.1 KW20 Regulator Adjustment (Figure 7-8)

Equipment required for this adjustment is as listed below.

1. Digital voltmeter with + or -20 millivolt accuracy
2. Phillips screwdriver
3. Alignment tool

The following procedure adjustment should be performed with the system powered up. The letters in parentheses correspond to points in Figure 7-6.

1. Remove the plastic shield by loosening the two Phillips-head holding screws (A).
2. Connect the voltmeter between points B and C.
3. Adjust the potentimeter D for a -5.2 reading on the voltmeter.
4. Remove the voltmeter leads.
5. Replace the plastic cover and tighten the Phillips-head screws.

> NOTE
> If -5.2 volt supply cannot be adjusted properly, replace the KW20. Refer to Paragraph 7.4.1.

7.5.2 M8576 Clock Deskew Procedure (Figure 7-9)

The following equipment is required.

1. Oscilloscope - Tektronix 475 or equivalent
2. W9025, 12 inch extender module
3. Alignment tool

7.5.2.1 Clock Check Procedure

> NOTE
> This is not a deskew procedure. If the requirements of this check are not satisfied, the deskew must be performed as described in Paragraph 7.5.2.2.

1. Set oscilloscope to 1V/div. for both channel 1 and

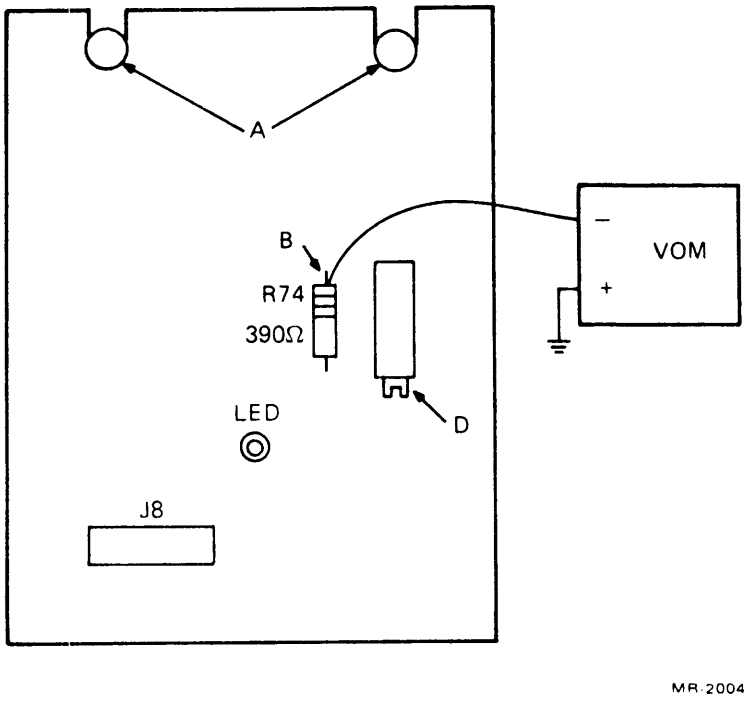


Figure 7-8 KW20 Clock Adjustments

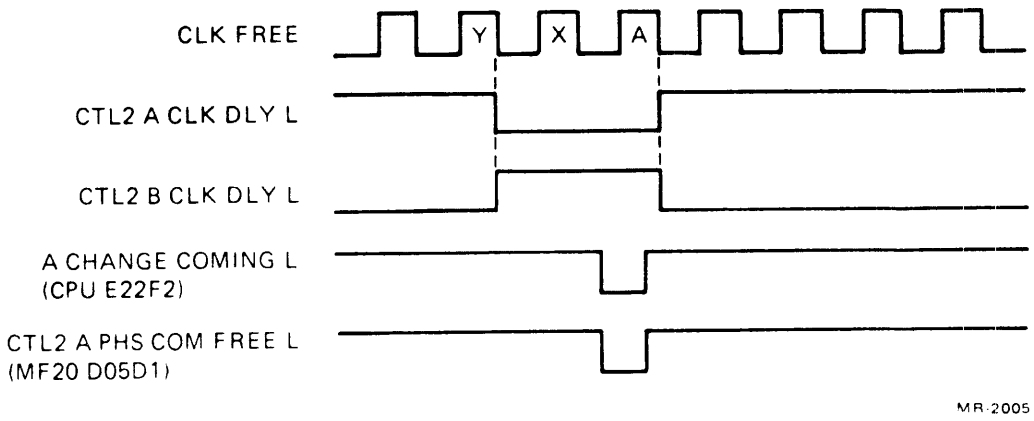


Figure 7-9 MF20 Clock Deskew

channel 2. Set sweep rate to 20 ns/div. Load the KL10 microcode. Select full clock rate and source the clock from the master oscillator at 30 MHz. Start the microcode (CR0, FW72/3, CS2, SM).

2. Place probe 1 on pin E22F2 in the CPU bay (signal A CHANGE COMING L) and place the clip on a ground pin. Place probe 2 on pin D5D1 in MF20 (signal CTL2 A PHS COM FREE L) and place the clip on a ground pin. Sync internal on channel 1. These signals must be identical.

3. Move probe 1 from CPU bay to MF20 at pin C5M2 (signal CTL2 B PHS COM FREE L) and place the clip on a ground pin. Set the oscilloscope to add channel 1 with channel 2. The waveform observed must be perfectly symmetrical with respect to ON/OFF time.

7.5.2.2 Clock Deskew Procedure

1. Power down the MF20 and place the M8576 on the extender. Rotate the two switches all the way clockwise. Restore power and load the KL10 microcode, master reset the machine (MR), select full clock rate (CR0), and source the clock from the master oscillator at 30 MHz. Now start the microcode (SM).

2. Place the probe for the viewable external sync on pin E22F2 in CPU bay (signal A CHANGE COMING L) and place the ground clip on a ground pin.

3. Select negative external sync as a trigger on oscilloscope.

4. Put the dip clip on E86 of the M8576 and place probe 2 on E86 pin 5 (CT A CLK DLY L). Place the ground clip on pin 1.

5. Place probe 1 on E86 pin 9 of the M8576 (CLK FREE) and ground clip on pin 16.

6. View external sync and locate A phase tick of CLK FREE (probe 1). It is the first positive-going pulse after the CPU signal A CHANGE COMING L goes low.

7. Having located the A tick of CLK FREE, define the positive pulse before A tick to be X tick. Define the positive pulse before X tick to be Y tick (see Figure 7-7).

8. Rotate the bottom switch to align the negative-going edge of CT A CLK DLY L (probe 2) with the negative-going edge of Y tick (probe 1). If this optimal setting is not possible, the negative-going edge of CT A CLK DLY L must be aligned within the following range: after the

CHAPTER 9
MG20 SPECIFIC INFORMATION

9.1 INTRODUCTION

The MG20 memory option uses the new M8570 memory modules. These modules use 64K RAMs and each provides a storage capacity of 1024K X 11 bits. The minimum configuration of an MG20 is one group of four modules, providing a storage capacity of 1024K words (one megaword). An MG20 can have up to three groups of modules, thus providing a storage capacity of 3072K words. Version 5.1 of the TOPS-20 operating system supports two megawords, and Version 6.0 supports three megawords. Version 7.02 of the TOPS-10 operating system supports up to four megawords.

To provide the entire four megawords of storage with MG20s, use two MG20s with two groups of memory modules each. Thus, the entire four megawords can be stored internally, eliminating the need for an expansion cabinet.

The MG20-DA/DB includes an expansion cabinet required for the KL10-D system. It comes with 2048K words of memory. To achieve the maximum memory of 4096K words, an additional MG20 backplane and power supply must be installed, as shown in Figure 9-1. The KW20 master oscillator housed in the MG20 cabinet provides the system clock. The floorspace requirement of the KL10-D can be greatly reduced by replacing the core memory cabinets with a single MG20 expansion cabinet. The MG20 can, of course, be used in combination with the core memory.

NOTE

Because the MG20-DA/DB controller is single-ported, there is no support for SMP or any external channel (DMA) configurations.

9.2 CONFIGURATIONS

MG20-LA - MG20 backplane, one group of modules (1024K words of memory), power supply, battery box, and KW20 master oscillator, 60 Hz.

MG20-LB - MG20 backplane, one group of modules (1024K words of memory), power supply, battery box, and KW20 master oscillator, 50 Hz.

MG20-E - One group of four M8570 modules (1024K words).

MG20-LC - MG20 backplane, one group of modules (1024K words of memory), power supply, and battery box, 60 Hz for expansion beyond 2048K words.

MG20-LD - MG20 backplane, one group of modules (1024K words of memory), power supply, and battery box, 50 Hz for expansion beyond 2048K words.

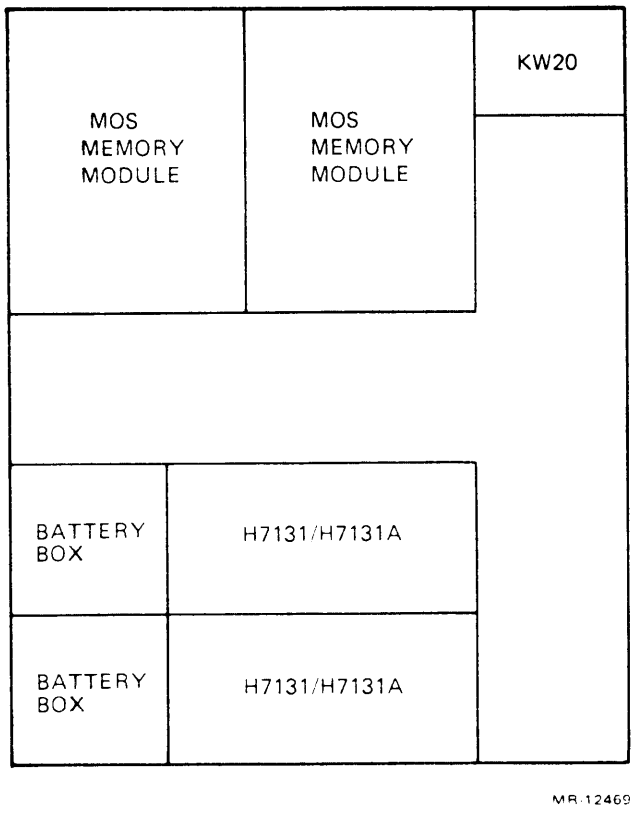


Figure 9-1 MG20-LE/LF Physical Layout

MG20-DA - Expansion cabinet containing an MG20 backplane, two groups of modules (2048K words), power supply, battery box, and KW20 master oscillator, 60 Hz.

MG20-DB - Expansion cabinet containing an MG20 backplane, two groups of modules (2048K words), power supply, battery box, and KW20 master oscillator, 50 Hz.

9.3 REFERENCE DOCUMENTS

MG20 Field Maintenance Print Set (MP01904-01)
DECsystem-10 Site Preparation Addendum (EK-DEC10-SP-008)
DECSYSTEM-20 Site Preparation Addendum (EK-DEC20-SP-005)
KL10-Based Site Preparation, Power System Installation Manual (EK-OKL10-SP-002)
KL10 Maintenance Guide (EK-OKL10-MG)

9.4 SITE PREPARATION OF THE MG20-DA/DB FOR THE KL10-D SYSTEM

NOTE

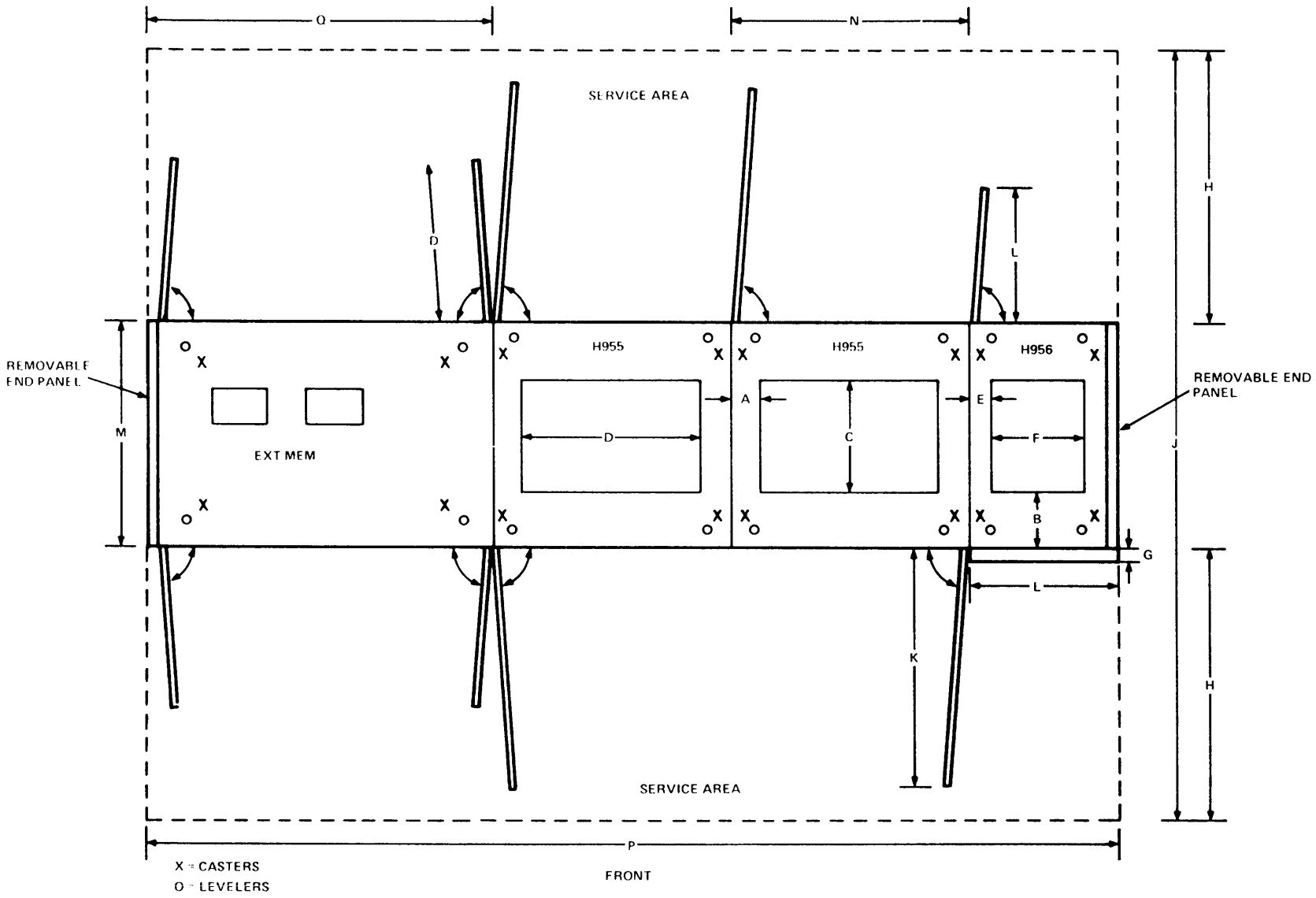
This equipment generates, uses, and may emit radio frequency. The equipment has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may 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.

The MG20 external cabinet is an addition to the existing KL10-D system. The MG20 cabinet is bolted to the I/O bay using the hardware and panels provided. Cables are supplied to interface the MG20 to the KL10-D. Refer to Figure 9-1 for the physical layout of the MG20. Figure 9-2 reflects the service area requirements for the KL10-D/MG20 configuration.

9.5 UNPACKING THE MG20-DA/DB

Digital has implemented strict shipping regulations to ensure that all hardware and software is received as stated in the customer's order. All boxes and skids should be left intact to avoid damaging or losing equipment, accessories, parts, or manuals. Before removing the MG20-DA/DB from its shipping container, verify the integrity of the shipment by checking the following.

1. Verify that the correct number of boxes and skids was received. (These numbers will appear on one of the gummed labels affixed to the shipment.)



DIMENSIONS	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q
METERS	0.09	0.19	0.38	0.61	0.08	0.36	0.04	0.91	2.59	0.79	0.50	0.76	0.80	3.33	1.14
INCHES	3.7	7.5	15	24	3.2	14	1.5	36	102	31	20	30	31.5	131	45

MR 1/470

Figure 9-2 KL10-D and MC30 Service Area Requirements

2. Inspect the box and skid housing the MG20-DA/DB for visible damage.

3. Check that all containers are sealed, with labels placed across the tape/band.

If you notice any damage, or if any items are missing, report the problem to the shipper and the local DIGITAL field service branch supervisor responsible for the installation.

When you are ready to begin the installation, move the equipment to a convenient work area close to the DECsystem-10. Remove the plywood panels and polyethylene bag used to protect the MG20 during shipment.

NOTE

Save all shipping containers and mounting hardware. These may be needed if the MG20 must be returned.

9.6 INSPECTING THE MG20-DA/DB

Once the outer wrapping has been removed, carefully inspect the hardware. If any discrepancies are found, report them to the shipper and the local DIGITAL field service branch supervisor. A report should be made on the Marlboro Manufacturing Feedback Form so that shipping problems may be corrected at the source. Inspection should include the following steps.

1. Take a complete inventory of all items as shown on the shipping list.

NOTE

If any items are missing (that is, cables, modules, etc.), do not proceed with the installation. Taking the customer's system down to install the MG20 and then finding you are unable to complete the job on schedule because of a missing part could cause a serious customer relations problem.

2. Inspect the logic assembly, power supply, and battery box for damage (dents, bent pins, broken wires, missing screws, etc.).

3. Verify that all the logic modules are plugged into the correct slots, as shown in the module utilization list (MUL) (Figure 9-3).

9.7 MG20 MODULE UTILIZATIONS (FIGURE 9-3)

Four M8570 memory modules make up one MG20-E group, providing 1024K words of storage. Up to two groups (eight M8570 modules) can reside in each MF20 controller, providing up to 2048K words of storage.

Each MG20 backplane has the following logic modules. Figure 9-3 shows the module utilization list (MUL).

- One M8574 write path module
- One M8575 syndrome module
- One M8576 control and timing module
- One M8577 address and timing module
- Four or eight M8570 MOS memory modules. Each group of four M8570 modules provides a capacity of 1024K words; up to two groups (2048K words) can mount in a single MG20.

9.8 SOFTWARE AND DIAGNOSTIC REQUIREMENTS

To support an MG20, the DECsystem-10 or DECSYSTEM-20 must have software with version numbers equal to or greater than the following.

RSX20F Version 15-12 (supervisor supporting MG20)
TOPS-10 Release 7.02 (minimum release version)
TOPS-20 Release 5.1 (minimum release version) with autopatch for TGHA support
TGHA Version 4.0 (supporting MF20/MG20)

To support the MG20, you must have diagnostics with version numbers equal to or greater than the following.

DHKBF.A11	Version 0.4 - MF20/MG20 Controller Test I
DHKBG.A11	Version 0.4 - MF20/MG20 Controller Test II
MEMCON.A11	Version 0.10 - Memory Configurator
DFMMH.A10	Version 0.10 - MF20/MG20 4096K Memory Reliability

9.9 UPGRADING THE MF20 AND MG20 MEMORIES

9.9.1 Configuration Restrictions

1. No more than two groups (MG20-Es) or eight M8570 memory modules can be installed in one MG20-L cabinet (groups 0 and 1).

2. Old and new arrays cannot be combined in the same MG20-L cabinet.

3. The total storage capacity of combined core and MOS cannot exceed four megawords.

9.9.2 Power Supply

When you install two groups of M8570 memory modules in one MF20 backplane, check that you have an H7131 power supply made by ACDC, rather than the H7131A made by Digital in Burlington. Two groups of memory modules require 21 A from the +5.0 V power supply output. The H7131A power supply is rated for only 20 A and thus will fail. Use the H7131 power supply, rated at 35 A.

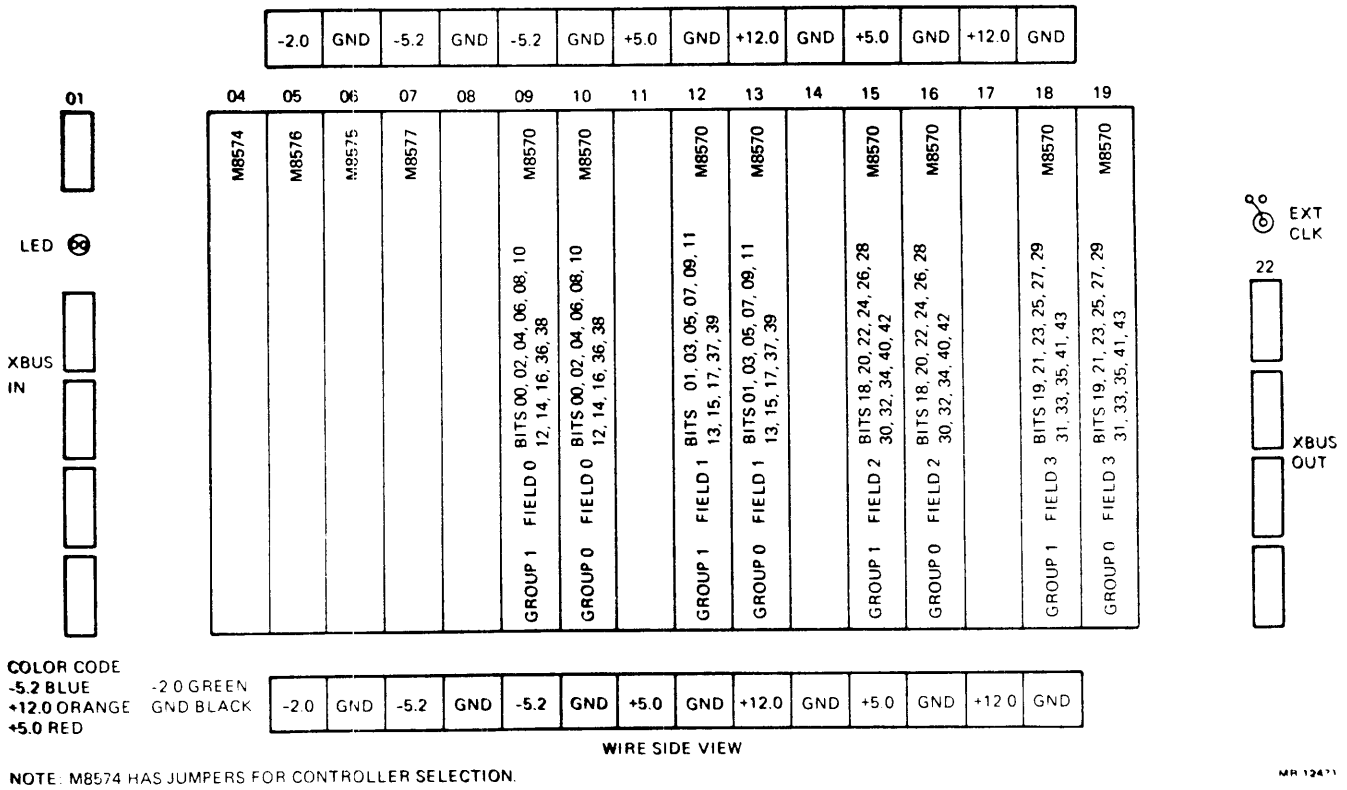


Figure 9-3 MG20 Simplified Module Utilization List

9.9.3 MF20 Upgrade Procedure

1. Remove all of the old M8579 memory modules (up to 12). Carefully place them into antistatic module jackets for storage or repackaging for return. No more than two groups (eight M8570 memory modules) can be installed into a backpanel. The first group is group 0 and the second is group 1. The appropriate backplane slots for each group are as follows.

Group Number	Backplane Slots
0	10, 13, 16, 19
1	9, 12, 15, 18 (optional)

2. Install the MG20 Module Revision History label (Part Number 3615747-01) over the old MF20 label on the rear cooling door.

3. Proceed to Paragraph 9.10 (Verification and Checkout Procedure).

9.9.4 MG20 Upgrade Installation - Adding a second group of M8570 storage modules to increase the capacity of an existing MG20 subsystem is a simple procedure. Follow these steps.

1. Power down the system.

2. Insert the four M8570 modules into the MG20 backplane slots shown above.

3. Power-up the system and conduct the verification and checkout procedures described in Paragraph 9.10.

9.10 VERIFICATION AND CHECKOUT PROCEDURE

9.10.1 DC Voltage Power Check

1. After installation is complete, power up the system.

2. Check the dc voltages on the MG20 backpanel.

Voltage	Tolerance	Color
-2.0 V	+/- 2%	Green
-5.2 V	+/- 2%	Blue
+5.2 V	+/- 2%	Red
+12.0 V	+/- 2%	Orange
Ground		Black

NOTE
The +12.0 V is not used on the new array; it is only required for the KW20 to sample the power from each MG20 controller.

9.10.2 Diagnostic Verification

9.10.2.1 DHKBF - MF20/MG20 Controller Test I -

1. Under KLDCP, load and run DHKBF Version 0.4. This diagnostic will verify the control logic. Set console switches to 0.

2. All memories should be configured and tested.

3. Run five passes. No errors are acceptable.

9.10.2.2 DHKBG - MF20/MG20 Controller Test II -

1. Load and run DHKBG Version 0.4. This diagnostic will verify the control logic and storage modules. Set all console switches to 0.

2. Run five passes. No errors or swaps are acceptable.

9.10.2.3 DFMMH - MF20/MG20 4096K Memory Reliability -

1. Under KLDCP, perform a BT command. This will load the latest KL microcode, configure the on-line MF20/MG20 controllers, and load the SUBKL subroutine package. Verify that the new memory configuration is correct.

2. Load and run DFMMH.A10 Version 0.10. With RH switches set at 407000 and console switch 13 set at 1 (totals), run at least one complete pass on the entire memory. This will take approximately 15 minutes per megaword for a quick verify.

3. No hard errors are acceptable.

9.11 STORAGE ARRAY ORGANIZATION

Before continuing with detailed descriptions of the MG20, several terms need to be understood. These terms relate to how the MOS storage array is organized; therefore, they are used in discussing any part of the MG20 that relates to accessing data within the MOS array.

9.11.1 Storage Units

A single MG20 can store up to 1024K (1,048,576) 44-bit words (36 data bits + 8 control bits). Each one of these bits must be assigned to a single storage element, and this assignment must uniquely identify each bit in the memory module. Once the

assignments have been made, the control section can be designed to permit the program to access any one of these millions of storage elements. To aid the design, these storage elements are arranged into more manageable units. The following definitions describe the units.

Group - A set of four M8570 storage modules. A single MG20 can control up to two groups, a total of eight storage modules.

Block - A set of 176 MOS storage elements (chips). A group consists of four blocks.

Sub-block - A set of 44 MOS storage elements. A block consists of 4 sub-blocks.

Field - An 11-bit slice of the MG20's internal data path. Each M8570 storage module delivers 11 bits (1 field) when accessed. Four M8570 modules are required to provide storage for a 44-bit word.

9.11.2 Memory Module Access

When the MBox requires data from the MG20, it places a 22-bit address on the XBus and initiates a memory cycle. The MG20 stores the address in its control sections and utilizes it as decribed in Figure 9-4. It is important to recognize that software has the ability to load the address response RAM, which makes the relationship between storage unit and physical MOS chip a program-controlled variable.

9.12 FUNCTIONAL BLOCK DIAGRAMS

This paragraph examines more closely the functional areas of logic within each module. It emphasizes how the logic within the boundaries of each module is partitioned into specific functional areas and how these areas interact to contribute to the overall MG20 operation. The location and function of key signals entering and leaving each module are shown in Figure 9-5.

9.12.1 M8570 Memory Module

The M8570 memory module is partitioned into the following functional areas.

1. MOS RAM array
2. Data multiplexers
3. Address, data, and timing control
4. Personality PROM

The MOS array consists of 176 64K MOS chips. There are eleven identical prints, SM00-SM10, each of which describes a 1-bit slice of the data path. All data entering or leaving the MOS array is buffered by a set of DC008 data multiplexer chips. Forty-four multiplexer chips are required (11 per M8570) to perform the dual functions of:

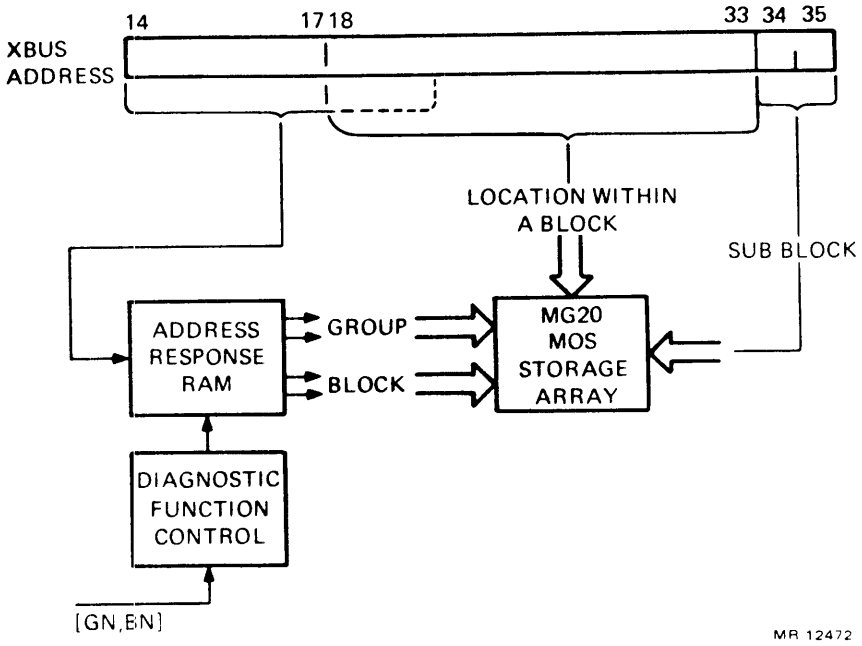


Figure 9-4 Memory Module Access

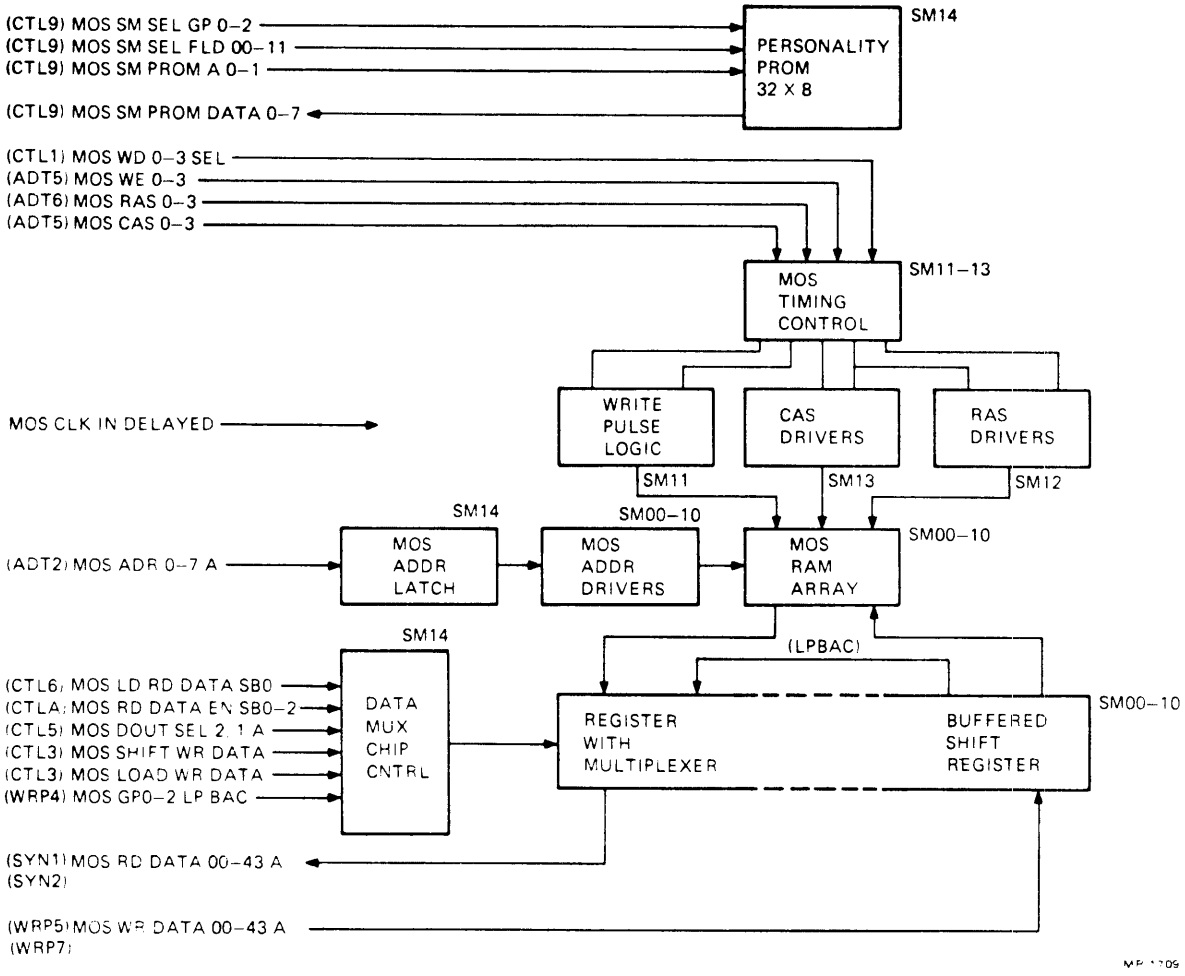


Figure 9-5 M8570 Functional Block Diagram

1. Quadword buffering during read and write

2. ECL to TTL translation for a write, and TTL to ECL translation for a read.

To permit the transfer of data between the MOS storage array and the data multiplexer chips, the following signals must be activated in a definite sequence.

1. RAM address - 16 bits to specify one of 64K locations to be read or written. (This is applied in two 8-bit units.)

2. Row address strobes (RAS)

3. Column address strobes (CAS)

4. Write enable signals during write only (WE)

These functions are handled by the address and timing logic shown on SM11 through 14.

Getting data into and out of the storage module is the task of the data multiplexer chip control logic. In addition to normal read/write data control, the MOS WR DATA bus can be looped back via the multiplexer chips to test the integrity of the connections between the WRP and SYN modules. This is achieved via a diagnostic function operation. The SM14 is the personality PROM. The locations within this PROM are set up during manufacture to identify the M8570 and its unique characteristics. Software can read this information to track error statistics for each individual storage module manufactured. Refer to Paragraph 9.14 for more detail on the PROM content.

9.13 MEMORY MODULE OPERATION

This paragraph provides a detailed description of the operation of M8570 memory modules. It discusses the control signal sequences involved with storing and retrieving data to/from the MOS storage array. The organization of the storage array has been described in Paragraph 9.11. Paragraph 9.16 contains more detailed descriptions on the 64K (typical) MOS RAM chip and the DC008 data multiplexer chip.

9.13.1 Simplified Block Diagram (Figure 9-6)

All data to be stored in the MOS array passes through the M8574 module and appears at the inputs to the M8570 storage modules as 44 signals, MOS WR DATA D00-43A. All data retrieved from the M8570s appears at the outputs as MOS RD DATA D00-43A and is sent to the M8575 module for checking and ultimate delivery to the MBox via the M8574 and the XBus. The selected group of M8570 modules can store 1024K 44-bit words, and a single set of MG20 control boards can control up to two groups. Therefore, the ADT (M8577) and CTL (M8576) modules must generate the control signals required to select the proper location within the storage array and ensure passage of the 44 bits to or from the 64K MOS chips.

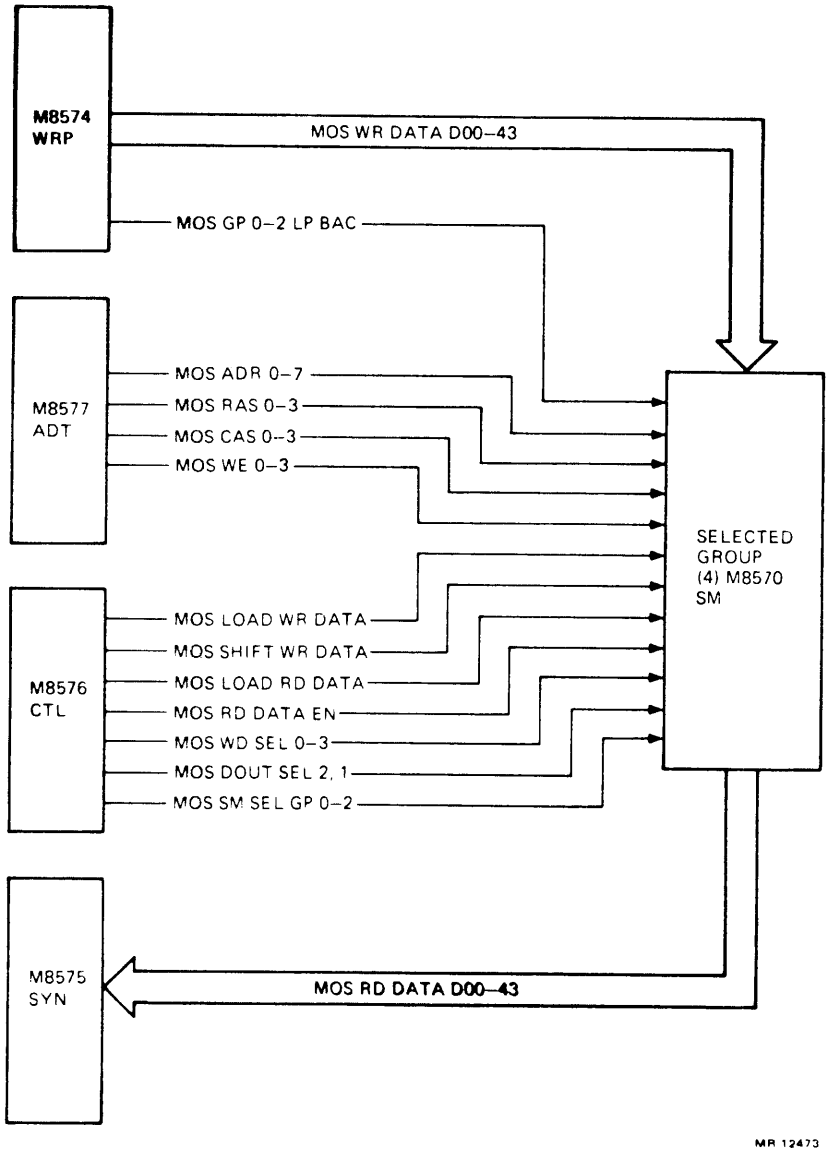


Figure 9-6 MOS Memory Storage Module Simplified Block Diagram

9.13.2 One-Bit Slice (CS-M8570-0-SM03)

This diagram shows a 1-bit slice of the data path within the M8570 module. Each module print set contains 11 diagrams similar to this one to describe a complete field. Since all 44 bits operate in the same manner, it is only necessary to describe one bit position. (SM03 was chosen as the example -- bit positions 2, 3, 20, or 21.) Each one-bit slice consists of the following:

1. Eight 2965 MOS driver chips to buffer the MOS address

2. Sixteen 4864 MOS RAM chips. Each chip stores 65,536 bits and is organized as follows:

	BLK 0	BLK 1	BLK 2	BLK 3
SB0	E75	E74	E73	E72
SB1	E71	E70	E69	E68
SB2	E67	E66	E65	E64
SB3	E63	E62	E61	E60

3. A single DC008 data multiplexer chip E58 that consists of:

   - an upper register multiplexer section used for read cycles
   - a lower buffered shift register section used for write cycles.

The following list outlines the two-digit number for RAS/CAS signals and algebraic expression to determine bit position.

1. Each RAS/CAS signal is followed by a two-digit number (XY) where:

   X = block number
   Y = sub-block number

2. To determine the bit position number, the algebraic expression [TX18+02+U] must be evaluated where:

   U T = Field number

   0 0 0
   0 1 1
   1 0 2
   1 1 3

   N and A specify backplane slots for each group of four M8570s.

FLD	0	1	2	3
N (GRP0)	10	13	16	19
A (GRP1)	9	12	15	18

Example: Field 2 for group 1 is implemented by the M8570 that plugs into slot 15.

9.13.3 Write Operation

The following paragraph describes the operation during a quadword write starting at an XBus address that begins on a quadword boundary. First, the data bits for each word, WD0, 1, 2, and 3, must be transmitted over the XBus from the MBox to the WRP module. Then they are shifted one bit at a time into the DC008 buffered shift register. At each SM CLK B, the data bit at the DATA IN input (pin 1) is shifted into QC; QC is shifted into QB; and QB is shifted into QA if the signal SM SHIFT WR DATA is asserted. After three bits have been shifted in, SM LOAD WR DATA is asserted along with the bit for WD3 being present at the DATA IN input. On the next SM CLK B, the following transfers occur with the DC008.

```
D0 <= QA
D1 <= QB
D2 <= QC
D3 <= DATA IN
```

The bit position being written for each of the four words appears at the output of the DC008. The data is latched and ready to be written into the proper MOS RAM chips as shown below.

```
SM D03 WD0 written into sub-block 0
SM D03 WD1 written into sub-block 1
SM D03 WD2 written into sub-block 2
SM D03 WD3 written into sub-block 3
```

The ADT module gates out the first half of the 16 bit MOD address, SM ADR 0-7, along with a row address strobe (RAS) as defined below.

```
RAS 00-03 if writing block 0
RAS 10-13 if writing block 1
RAS 20-23 if writing block 2
RAS 30-33 is writing block 3
```

This latches the row address within the four MOS chips in the addressed block. After RAS has latched the row address, the ADT module gates out the second half of the 16-bit address, also as SM ADR 0-7, along with a column address strobe (CAS) as defined below.

```
CAS 00-03 if writing block 0
CAS 10-13 if writing block 1
CAS 20-23 if writing block 2
CAS 30-33 if writing block 3
```

This latches the column address within the four MOS chips in the addressed block. The addressing scheme is as follows:

```
7 row addresses (128 rows)
9 column addresses (512 columns)
```

While CAS signals establish the column address, the appropriate write enable signals are activated to allow writing the data into the MOS storage elements as defined below.

SM WE 00-03 enables writing **sub-block** 0
SM WE 10-13 enables writing **sub-block** 1
SM WE 20-23 enables writing **sub-block** 2
SM WE 30-33 enables writing **sub-block** 3

During a quadword write, SM11 MOS WD SEL 0-3 would permit activating a complete set of write enable signals to enable writing four sub-blocks (4 MOS chips). For other than a quadword (1, 2, or 3), these signals would activate only the write enable signals specified by XBUS RQ 0-3.

9.13.4 Read Operation

During a read, the sequence of operations is reversed. RAS is asserted first along with the row address on SM ADR 0-7. Once the row address has been latched in the chip, CAS is asserted along with a column address on SM ADR 0-7. At this point, one MOS storage element in each sub-block is selected and four bits are available at the data inputs to the DC008 register chip (pins 13-16). The signal SM LOAD RD DATA is asserted by the CTL module to allow the next SM CLK B to parallel load the quadword into the data multiplexer chip. Now SM RD DATA EN is asserted by the CTL module to enable the signals SM DOUT SEL 2-1 to select the order in which the words will be sent to the SYN module on MOS RD DATA D00-43. These two signals are generated by the read data mover logic on the CTL module as a function of:

1. The state of XBUS RQ 0-3
2. The initial state of XBUS ADR 34-35

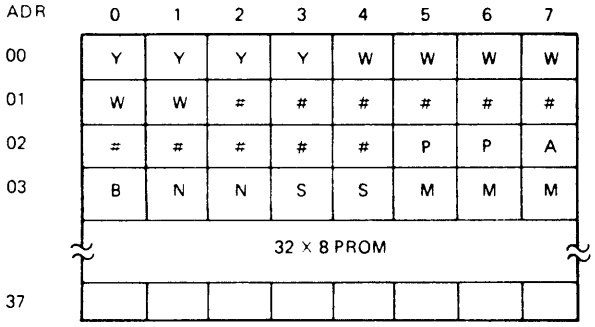
The following table defines their use.

SET 2	1	Word available at DC008 pin 11
0	0	WD0
0	1	WD1
1	0	WD2
1	1	WD3

As each word is sent to the SYN module, it is checked for errors and sent to the WRP module's port buffer to be delivered to the XBus.

9.14 PERSONALITY PROM

Each M8570 memory module contains a PROM that provides a description of that individual board's "personality." The factors considered when determining board personality are listed in Figure 9-7, which also describes the bit format used to encode each characteristic. During initial start-up, the software reads the contents of the PROMs to establish an association between specific physical hardware and a data base maintained and used by the software. This data base permits the software to gather and log



YYYY — 4-BIT YEAR NUMBER, BCD EQUIVALENT OF THE LAST DIGIT OF THE CALENDAR YEAR.

WWWWWW — 6-BIT WEEK NUMBER

— 11-BIT SERIAL NUMBER

PP — MOS STORAGE BOARD POPULATION CODE

- 00 BLOCK 0 ONLY EXISTS
- 01 BLOCK 0 AND 1 EXIST
- 10 BLOCK 0, 1, AND 2 EXIST
- 11 FULLY POPULATED (DEFAULT)

A — PARITY – CHOSEN TO GENERATE EVEN PARITY FOR LOCATION 0, 1, AND 2 IN TANDEM

B — ODD PARITY FOR LOCATION 3.

NN — 2-BIT TIMING NUMBER

SS — MOS CHIP SIZE

- 11 4K
- 10 16K (M8579)
- 01 32K
- 00 64K (M8570)

MMM — 3-BIT MOS CHIP MANUFACTURER CODE.

- 000 MOSTEK
- 001 FUJITSU
- 010 HITACHI
- 011 NEC
- 100 MOTOROLA
- 101 MITSUBISHI
- 110 OTHER (DEC, ETC.)
- 111 RSVD

MR 12474

Figure 9-7 Personality PROM - Bit Format

data errors and save this information in a disk file that associates an actual module serial number with its specific error statistics over a period of time. Service personnel can print out this file and use the information to isolate and repair faults. The timing information is used to establish MOS read/write timing sequences that may vary from one MOS manufacturer to another. The hardware to support the feature consists of:

1. A 32 X 8 PROM chip on each storage board that is written (first four locations) when the board is manufactured

2. Logic to permit software to select and read the first four locations within any PROM

9.14.1 Functional Block Diagram

Each M8570 board (SM14) uses a single 82S123 32 X 8 PROM chip. To enable reading the information from the PROM, the software executes Diagnostic Function 2 (Paragraph 4.3.3) to generate the following signals (CTL9).

MOS SM SEL GP 0-3	Selects which **group**
MOS SM SEL FLD 00-03	Selects which **field** (M8570)
MOS SM PROM A 0-1	Selects one of the first four PROM addresses

These signals are generated by decoding P BUF D11-14, with two 10162 chips on (CTL9) to generate single group- and field-select signals that are sent to the storage boards. The two-bit address signals A0 and A1 are a direct function of P BUF D 13-14 as coded by the programmer.

Once enabled, the selected PROM (SM14) sends back eight bits (MOS SM PROM DATA 0-7) with the contents of the location specified. This data is received by the M8576 (CTL9) and sent to eight 10164 mixers on (CTL7) where they generate CHK TO P D 07-14. From here, the PROM data is sent to the port buffer diagnostic function (M8574) via the (M8575). Once in the port buffer, it can be gated onto the XBus data lines. To retrieve all the PROM data, the software must execute four Diagnostic Function 2s for each storage board. During each PROM read, the software must perform the parity check since there is no hardware provided for this function.

9.15 M8570 MEMORY MODULE GEOGRAPHY

Figure 9-8 is a diagram of the physical partitioning of the 176 MOS and 11 data multiplexer chips on each M8570. Although not drawn to scale, it can be used to locate a chip or particular backplane test point for viewing the data entering (write) or leaving (read) an M8570 module. The information presented in the diagram permits the service engineer to select a test point once the following information has been determined.

- Failing **group** of M8570s
- Failing bit position (defines **field**)
- Failing **block** and **sub-block**

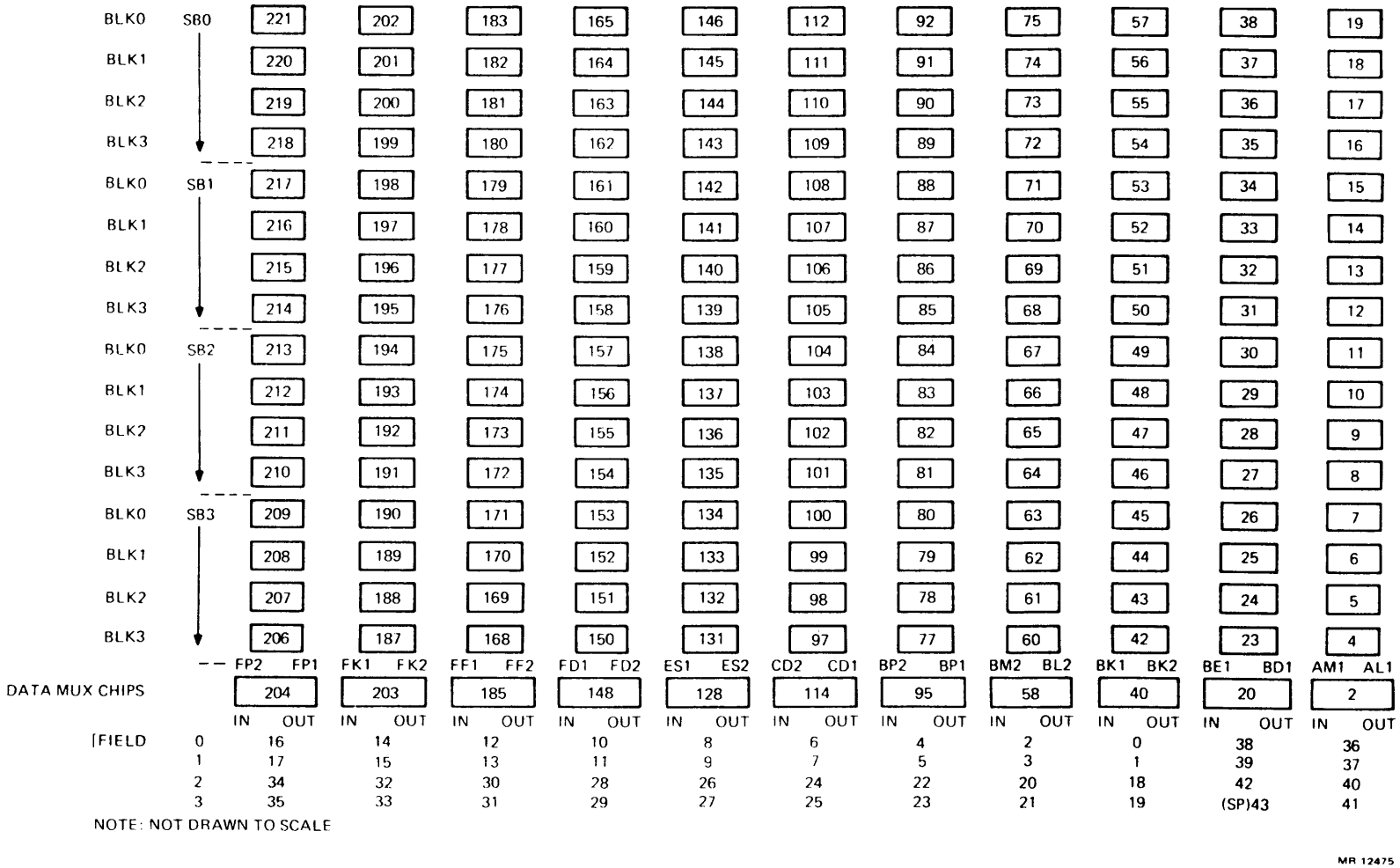


Figure 9-8 M8570 Storage Module Geography

This information is available on the error printouts furnished by the diagnostic. The failing **group** and **field** numbers are used to identify the MG20 slot as described below:

MG20 Slots

Field	Group 0	Group 1
0	10	9
1	13	12
2	16	15
3	19	18

Example: Bit 22 in **group** 1 is failing. From Figure 9-8, bit 22 is in **field** 2. The above table shows that field 2 with group 1 is slot 15.

To display the data input, examine pin B15P2 and the output at pin B15P1. If the printout only occurred for block 1, sub-block 2, E83, a MOS chip should be replaced. If the failure affected all blocks and all sub-blocks, the data multiplexer chip E95 might be suspected. This example assumes that the fault has already been isolated to the M8570 using directed module replacement.

9.16 COMPLEX IC DESCRIPTIONS

Most of the integrated circuits used to design the MG20 are standard KL10 devices. The service engineer should be able to understand their operation from the context in which they are shown in the engineering drawings. This section describes the operation of the more complex ICs.

9.16.1 DC008 Data Multiplexer Chip (Figure 9-9)

Each M8570 storage module contains 11 DC008 data multiplexer chips to buffer all information entering or leaving the MOS storage array. The DC008 is a complex 22-pin IC that consists of the following two independent sections.

1. A buffered shift register section used during a WRITE
2. A register with multiplexed outputs used during a READ

Each chip interfaces to four MOS RAM chips to permit quadword reads and writes. Block-level description of the operation of the DC008 can be divided into three modes of operation.

9.16.1.1 Read Operation - During a read, four MOS RAM chips are addressed and strobed for each bit position. Each RAM chip constitutes a sub-block as identified by XBUS ADR 34-35. The data bits at the outputs (pin 14) of the four MOS RAM chips feed pins 13 through 16 of the DC008 multiplexer chip. To load these four bits (bit n for WD0, 1, 2, 3) into the DC008, the signal SM LOAD RD DATA is activated which allows the next SM CLK to load the data into the data multiplexer chip. Once loaded, the data must be transferred serially to the M8575 module over a single data line MOS RD DATA Dn. To achieve this the mixer output section must be

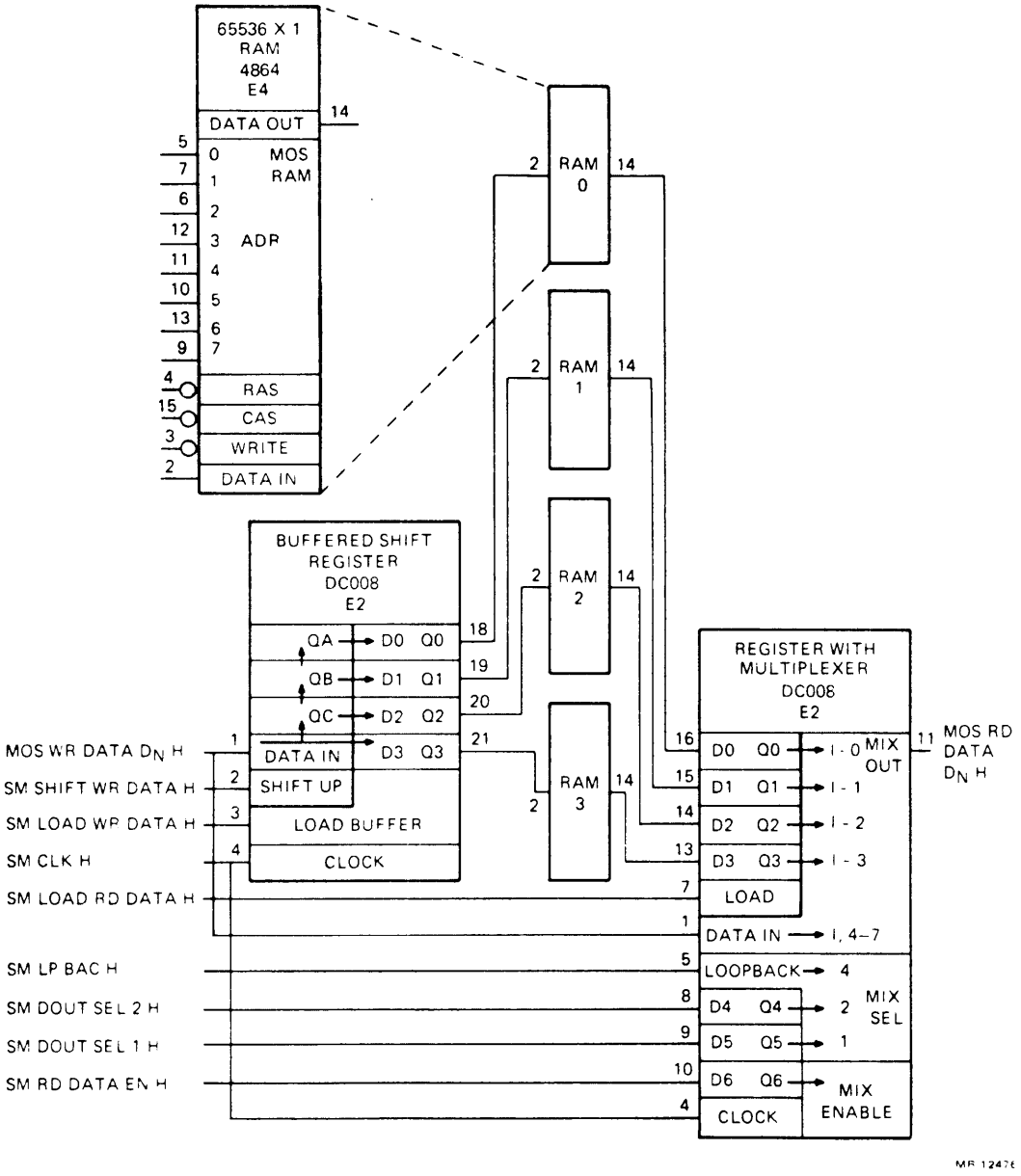


Figure 9-9 DC008 Logic Block Diagram

enabled and the output at pin 11 switched to reflect the state of one of four possible bits, Q0-Q3. The signal at pin 10, SM RD DATA EN, es asserted to enable the mixer, and the signals at pins 8 and 9, SM DOUT SEL 2, 1 specify which of the four words the M8575 is requesting.

9.16.1.2 Write Operation - During a write cycle, the information flows from the M8574 to the DC008 to the MOS RAMs. The bits are transmitted serially from the M8574 over a single data line, MOS WR DATA Dn, and shifted into the shift register section of the DC008 to position the bits properly with respect to the four RAM chips. The bits must always be sent in ascending order to ensure proper positioning.

To write a quadword the M8574 would place bit n for WD0 on the MOS WR DATA Dn line at pin 1. Following this, the signal at pin 2, SM SHIFT WR DATA H, would be asserted to permit the next SM CLK at pin 4 to shift the data up through the shift register section as follows.

```
QA <-- QB
QB <-- QC (WD0)
QC <-- DATA IN (pin 1 = WD1)
```

When word 2 is ready, the M8574 would place bit n for WD on the MOS WR DATA Dn line at pin 1, assert SM SHIFT WR DATA H, and shift the data up through the shift register as follows.

```
QA <-- QB WD0
QB <-- QC WD1
QC <-- DATA IN (pin 1 = WD2)
```

When the M8574 sends the last bit (bit n of WD3), it asserts SM LOAD WR DATA, which causes the following transfer to occur at the next SM CLK.

```
D0 <-- QA WD0
D1 <-- QB WD1
D2 <-- QC WD2
D3 <-- DATA IN (pin 1 = WD3)
```

At this point the four bits of the quadword are available at the output pins 18-21. From here the MOD RAM sequence writes them into the memory chips.

9.16.1.3 Loopback Operation - When group loopback is enabled, the signal SM LP BAC H is asserted which results in the data in at pin 1 appearing at the output pin 11. This feature provides the diagnostic program with the means of verifying the integrity of the data path from the XBus through the M8574 to the DC008s on the M8570 and from the M8570 through the M8575 and M8574 back to the XBus.

9.16.2 4864 MOS RAM Chip (Figure 9-10)

The basic storage element in the MG20 is a 64K MOS RAM chip. This paragraph describes the operation of this chip in terms of its general structure and input/output characteristics. it also includes a short discussion of MOS memory fundamentals for the reader unfamiliar with this technology.

9.16.2.1 General Description (Figures 9-10, 9-11)

The 4864 is a dynamic random access memory circuit organized at 65,536 words by 1 bit. It is packaged in a standard 16-pin configuration and contains all the necessary address, control and timing, and data circuits necessary to interface the MOS storage cells to the external control logic. The 65,536 memory cells are organized into a 128 words (row) X 512 bits (column) matrix. Figure 9-10 shows a functional block of the chip organization and Figure 9-11 shows the pin connections.

9.16.2.2 Basic Operation

To access a specific cell within the 128 row X 512 column matrix, a 16-bit address is supplied to the chip in two 8-bit slices (A0--A7). The first 8-bit slice uses addresses A0--A6 to select one of 128 rows and is latched into internal address buffers within the chip by the signal RAS which accompanies the row address. The most signification row address (A7) will form part of the column address. After allowing sufficient time to latch the row address, the second 8-bit slice and row address A7 select one of 512 columns and is also latched into internal address buffers. This time by the signal CAS which accompanies the column address.

At this point a MOS cell in the matrix is selected for sampling and its state is presented at the data out pin from where it can be strobed into an external data buffer register.

The time period during which all cells must be refreshed is called the refresh interval and is approximately 2 milliseconds for the MG20 MOS chips. Since there are 128 rows and each row must be refreshed once per 2 milliseconds, this means that the external control logic must initiate a refresh cycle every 15.6 microseconds to prevent loss of information.

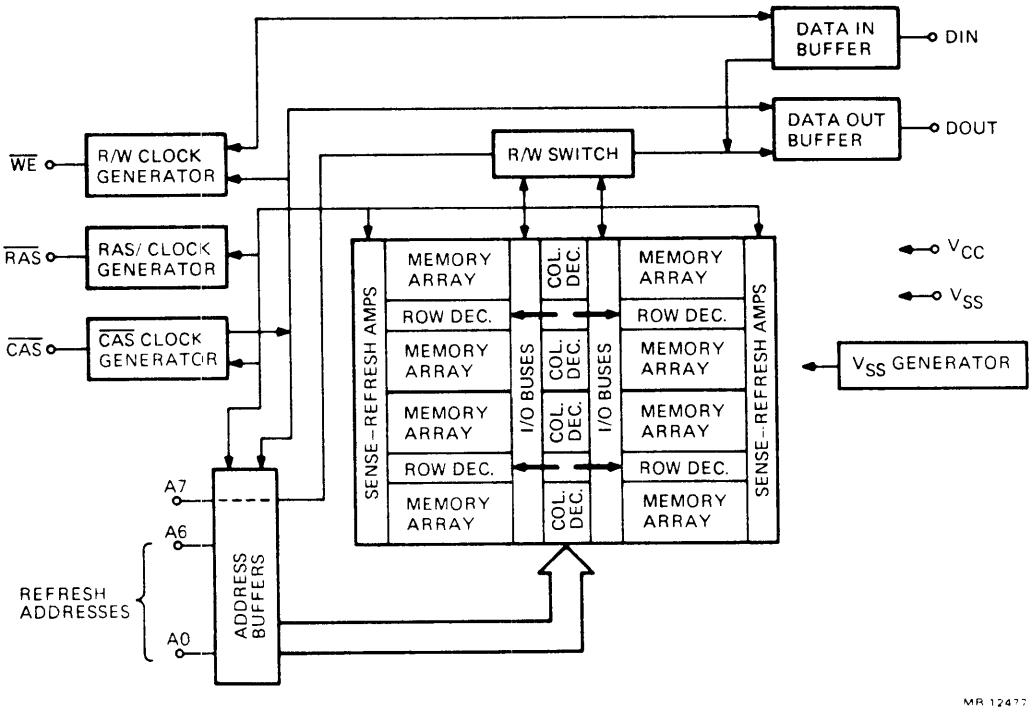
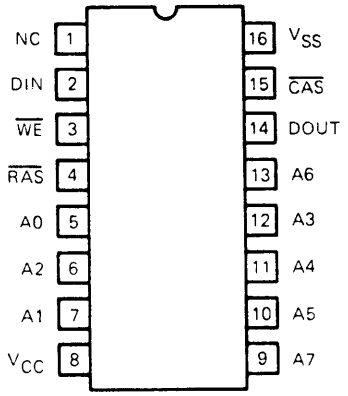


Figure 9-10 MOS RAM Block Diagram



(TOP VIEW)

A0-A7	ADDRESS INPUTS
$\overline{CAS}$	COLUMN ADDRESS STROBE
DIN	DATA IN
DOUT	DATA OUT
$\overline{RAS}$	ROW ADDRESS STROBE
$\overline{WE}$	READ/WRITE INPUT
V_{CC}	POWER (+5V)
V_{SS}	GROUND
A0-A6	REFRESH ADDRESS INPUT

MR 1247a

Figure 9-11 Pin Arrangements

CHAPTER 10
TGHA (THE GREAT HEURISTIC ALGORITHM)

10.1 INTRODUCTION

TGHA is the MOS memory analyzer program that is run by the monitor whenever a correctable error occurs in an MG20. It is responsible for keeping the MG20 memories in a viable condition using the MG20 maintenance features. TGHA obtains data about an MG20 error from the monitor. This program can be run in the following situations:

TGHA is used to create ASCII files (bit to text) from TGHA's data base. History .LST contains information about the hardware configuration, the state of the controller corrective logic, and a list of logged errors. TRACE.LST contains entries tracking the corrective action taken by TGHA. TGHA 4(0) can be run with either or both MF20 (16K) and MG20 (64K) memory modules in the memory system. However, both types of memory modules cannot be installed within the same controller.

1. TGHA sources will not be available to customers.

2. This document applies to both TOPS-10 and TOPS-20 unless otherwise specified.

3. If the memory boot portion of KLI does not configure a block of MG20 memory due to multiple bit errors, TGHA will indicate what memory and how much memory was not configured. The message will appear on the CTY, in the TGHA trace file, and in ERROR.SYS.

4. If the memory boot portion of KLI finds a parity error, KLI will use the spare bit and configure the block. TGHA, upon finding that a spare bit was set into use by KLI, will never use that spare bit for any other purpose. Upon finding such a condition, TGHA will output an appropriate message on the CTY, in the trace file, and in ERROR.SYS.

5. If a parity error occurs during swapping in of a spare bit, the spare bit will be flagged as if KLI had set the spare bit into use (see previous paragraph). Similar CTY, trace file, and SPEAR entries will be made.

10.2 UPDATE INSTRUCTIONS

TGHA data and trace file names are TGHA.DAT and TGHA.TRA respectively. The old files TGHAV2 or TGHAV3 DAT/TRA should be renamed to preserve the database.

10.2.1 Update of TOPS-10

For TGHA to run properly on TOPS-10, the following entry must be made in OPR.ATO after DAEMON is started:

```
1.  :SLOG
2.  :DEF TGHA =
3.  R TGHA
```

Since TGHA makes entries in ERROR.SYS via DAEMON, DAEMON must be running before TGHA is started. TGHA.EXE must reside in the [1,4] area:

TOPS-20: PS:<SYSTEM>

10.2.2 Update of TOPS-20

TGHA.EXE must reside in PS:<SYSTEM> area.

1. The monitor runs TGHA at initial system start-up.
2. The monitor runs TGHA in response to MG20 errors.
3. Field service can run TGHA to obtain list files.

10.3 MAKING LIST FILES WITH TGHA

NOTE

In order to run TGHA manually, the user must have at least minimal maintenance capabilities and be enabled. This is required because TGHA looks at the MG20 memory to determine the current configuration.

TGHA can also be run by field service personnel. On TOPS-10, TGHA recognizes that it is not the system TGHA. On TOPS-20, TGHA recognizes that it is not run by a job number of 0. Therefore, TGHA does not automatically look for MG20 errors.

TGHA will respond with a prompt of:

TGHA>

If "Help"<CR> is typed, the following test will be printed:

```
TGHA Help - Command     Purpose

Exit                    Exit from TGHA.
Help                    Type this text.
History                 Dump history file.
Trace                   Dump trace file.
TGHA>
```

The history and trace dump files will be created in the area from which TGHA has been run. They will be HISTRY.LST for TOPS-10 and HISTORY.LST for TOPS-20 and TRACE.LST respectively.

These files are listable versions of the TGHA database and the trace file. Although TGHA.TRA may look readable, you should know that it is the equivalent of a ring buffer -- that is, the oldest data is overwritten when the file becomes large enough. This prevents the trace file from becoming uselessly large. TRACE.LST is in chronological order, with the oldest entries first.

10.3.1 TGHA File Requirements

The TGHA file system for both TOPS-10 and TOPS-20 includes the following files:

File	Purpose	Creation Mechanism
TGHA.EXE	TGHA execution file	Monitor tape
TGHA.HLP	TGHA help file	Monitor tape
TGHA.DOC	TGHA overview	Monitor tape
TGHA.DAT	TGHA history/data file	TGHA.EXE when first run
TGHA.TRA	TGHA trace file	TGHA.EXE when first run
TGHA.BAD	Bad copy of TGHA.DAT	TGHA.EXE when the database gets confused

When the monitor first starts up on a system reload, it looks for TGHA in the appropriate area. If it is not there, TGHA cannot be run by the monitor. If TGHA.EXE is not found, the monitor will throw away the error data that might have been collected by TGHA. TGHA.DAT and TGHA.TRA will be created by TGHA when necessary.

TGHA.DAT will not grow as errors are incurred. TGHA will purge old data as its databases fill up. The only way that TGHA.DAT will grow is if new hardware is brought on-line -- that is, if a storage module is swapped or a new controller is added to the system. TGHA requires all of the history data that it has collected to remain intact so that it can make correct decisions about corrective action.

NOTE

TGHA.DAT should not be deleted. It contains important history information.

10.3.2 Initial System Start-Up

At initial system start-up, the monitor runs TGHA in start-up mode. In this mode, TGHA will first enable single-bit error reporting throughout MG20 memory. TGHA will then either build the history file if it does not already exist, or verify that it knows about all of the MG20 hardware that is on-line. If the history file exists and new MG20 hardware appears, TGHA will add this new hardware to its history file.

Once the start-up initialization is complete, TGHA then looks for any MG20 errors that have occurred since the initial system start-up.

10.3.3 MG20 Errors

For TOPS-10, when an MG20 correctable error occurs, the monitor wakes up the system TGHA. For TOPS-20, when an MG20 correctable error occurs, the monitor calls TGHA with a job number of 0. TGHA then gets the data for the MG20 error from the monitor. This is called a chronological error because the errors are stored in order of occurrence in the history database. There is a separate chronological error list for each module.

When the chronological error list is full, TGHA will then attempt to resolve known errors.

10.3.4 Known Error Determination

Known error determination is done statistically using the CHI squared goodness of fit formula. Starting with the first chronological error in the storage module, the formula is used to test the distribution of the other errors in the chronological error list for each type of possible hardware error.

For instance, the first hardware failure considered is a full MUX failure. A table is built by scanning all of the other errors in the storage module and tallying only those errors that are covered by the specific type of hardware error being considered -- in this case, a full MUX failure. The table for a full MUX failure is distributed by block and sub-block number. This translates into a distribution by chip since there are 16 chips involved in a full MUX failure. If the distribution is even enough, the CHI squared goodness of fit test will succeed, and a known error is declared. The known hardware determination continues until either the hardware error type is found or all of the error types have been tried.

The last type of hardware error considered is a cell error. If there are more than a minimum number of errors of the same cell (typically five), then a cell error is declared.

After an error type has been found, all of the chronological errors that are influenced by the known error type are eliminated from the chronological error list. This procedure continues for each remaining error in the chronological error list.

After the known error routine has been run, the corrective action routine is called.

10.3.5 Corrective Action

The goal of the corrective action routine is to determine the optimum corrective action given the current known errors. This is done on a per group basis.

First, the worst error in the group is found. This is the error that affects the most memory. Once the worst error has been found, the applicable spare bit is used to cover it. If there is more than one known error in the group, a scan is done to set all ICE (inhibit correctable error reporting) bits within the scope of the worst error. This procedure is repeated until all known errors for the group have been analyzed.

There is one exception to the previous procedure. If the memory boot portion of KLI has (during the DBE scan -- double bit error) used a spare bit to handle a double bit error, TGHA will not change the use of that specific spare bit. If TGHA were to attempt to use such a spare bit for another purpose, the possibility of parity errors resulting would be too great to risk.

Also, if a parity error occurs during the setting of a spare bit, TGHA will set the same considerations on that spare bit as if KLI had set it.

On the occurrence of any of the previous errors, the appropriate entries will be made in the ERROR.SYS file.

10.3.6 Parity Errors

Parity errors are handled in the usual manner. That is, if the monitor continues successfully after the parity error, it will attempt to take the page of memory with the parity error off-line. The monitor then runs TGHA. However, the monitor may not be able to take the page with the parity error off-line if, for instance, the page was part of the resident monitor. TGHA will enter parity errors in its trace file, not the history file database. TGHA will make an entry in the trace file and ERROR.SYS indicating physical address, block of memory, and the orientation of the four storage modules containing the error.

10.4 HISTORY FILE

The history file is made up of three types of pages.

1. Directory page
2. Group pages
3. Storage module pages

The sequence starts out with one directory page. This page contains configuration information about all of the MG20s on the system. Next is a group page, followed by four storage module pages. The pattern of a group page and four succeeding storage module pages is repeated for each group currently on-line in the MG20s. A storage module page for each storage module that goes off-line is then listed.

10.4.1 Directory Page

The first page of the history file contains the current configuration of the MG20 memory, including the location of each storage module by serial number. Note that the serial number in the history file is in the same format as the sticker on the storage module itself.

The location of the storage modules in MG20 memory is documented in the history list file for archival and field service reasons. The errors logged in the history file are dependent upon the orientation of the storage modules. If they are moved, the error corrections in the MG20 may have different characteristics. If field service wishes to replace a storage module, the exact location by serial number is therefore documented. The following example shows the difference between MF20 and MG20 controllers.

Example

```
TGHA HISTORY FILE DUMP AS OF 22-Mar-84 23:51:52
FOR CPU #2771
HISTORY FILE DATA BASE VERSION # 2

CONTROLLER  GROUP NUMBER  STORAGE MODULE SERIAL NUMBERS BY FIELD
NO. & TYPE                0       1       2       3
10   MF20   0             1030024 1030038 0520026 1030365
            1             1240062 1220060 1220056 1260031
            2             1220057 0270008 1290018 0420052

11   MG20   0             4080047 4080075 4080052 4080048
            1
            2
```

10.4.2 Group Page

The group page contains information specific to the configuration of the group. A record of the last time the group was used in this configuration may be useful if groups of storage modules are removed and put back into the memory later. Likewise, the time of the last change to the group is useful to know when determining where the most recent errors have occurred in the MG20.

The chronological list of errors in the history file and the entries in the trace file are tagged with the date and time. This is required when correlating corrective action with the MG20.

The group page also contains a map of the current state of the bit substitution RAM. The bit substitution RAM is used to direct bit replacement with the spare bit RAMs.

The value of the spare bit RAM contains four fields that have the following meanings:

Field Bits	Meaning
400	Either KLI used this spare bit to cover a parity error, or a parity occurred during a bit swap. The 400 bit is in the TGHA database only, and is not used in the actual bit swap process.
374	The octal value of the bit being replaced*
2	Set - correctable error reporting is disabled. Cleared - correctable error reporting is enabled.
1	Bit substitution RAM value parity bit.

* To find the value of the bit being replaced, shift this field two places to the right. The decimal equivalent of the octal digits is the word bit being replaced.

Bit Substitution RAM Examples:

Value	Meaning
256	- The spare bit is not in use. The value of the bit to be swapped points to the spare bit. The spare bit is in decimal bit position 43. - Correctable error reporting is disabled for this bit substitution RAM address.
255	- Spare bit is not in use (same as previous example). - Correctable error reporting is enabled.
652	- The 400 bit on indicates that this use of this spare bit prevents a parity error from occurring. TGHA will not change this value. The bit substitution RAM date requires odd parity. - Bit 42 of the MG20 data path is being swapped (the parity bit).

Example

```
THE LAST TIME THIS GROUP WAS USED WAS 14-FEB-84 09:00:05
THE LAST TIME A CHANGE WAS MADE TO THIS GROUP WAS 14-FEB-84 08:59:07
BIT SUBSTITUTION MAP:

000/ 040 255 255 255 040 255 255 255 255 255 255 255 255 255 255 255
020/ 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
```

The actual spare bit address in SBUS Diagnostic Function 7 is seven bits wide. The address in the group page accounts for only the lower five bits. The high order two bits is the group number within the controller. The group field is determined from the group position in the MG20 controller, not the group database. Therefore, the group portion of the field is not (and cannot be) kept within the spare bit RAM address table. This portion of the SBUS Diagnostic Function is filled in at execution time.

10.4.3 Storage Module Page

A storage module page contains information relating specifically to the storage module. The serial number appears in the same format as the sticker on the module itself. This is critical for field returns, which require that this page be attached to the storage module if it is to be returned to the factory for repairs. A record of the last time the module was used can be useful in tracking module swaps made in the MG20 and for field returns.

Example

```
STORAGE MODULE SERIAL NUMBER: 7460103
THE LAST TIME THIS STORAGE MODULE WAS USED WAS 14-FEB-84 09:00:28
NO KNOWN ERRORS DETERMINED YET FOR THIS MODULE
CHRONOLOGICAL ERRORS:
```

BLOCK	SUBBLOCK	BIFN	ROW	COLUMN	E #	TIME
0	0	6	123	142	149	13-Feb-84 16:31:48
0	0	6	122	142	149	13-Feb-84 15:40:58
0	0	6	130	142	149	13-Feb-84 15:37:00
0	0	6	71	10	149	13-Feb-84 15:31:34
0	0	6	123	142	149	13-Feb-84 15:30:06
0	0	6	2	144	149	13-Feb-84 15:30:06

The known error list contains the error information related to the known errors resolved from chronological errors by TGHA. Most of the information in each entry is used by TGHA to determine overlap of errors. This overlap is important when considering the optimum use of the spare bits.

The storage module chip location is included to facilitate module repair.

The time of resolution of the error may give an indication of the rate of deterioration of failure of the module.

The current chronological error list for the storage module will include any errors that have not been resolved into known errors yet. Any soft errors may also be included in this list. These soft errors will be removed from the list some time after a minimum default (about one week).

10.4.4 Unused Storage Modules

Any storage modules that are in the TGHA history file, but are not presently in use, follow in a similar manner. Inclusion of the unused storage modules in the history file is required when a storage module is replaced and the system is reloaded. Field service personnel need a copy of the history file for the bad storage module so that the error data can be included with the module when the module is returned for repairs.

10.5 TRACE FILE

The trace file contains dated entries indicating what corrective action has been taken by TGHA. If the TGHA software runs into confusion or a database becomes full, entries in the trace file will also reflect the difficulty. Serious MG20 hardware errors will also be entered in the trace file.

10.5.1 Spare Bit Swap

Example

11-Oct-82 15:50:12 - IN CONTROLLER 10, GROUP 0, VECTOR 3, THE SPARE BIT IS GOING ONCE THE SPARE BIT SWAP HAS COMPLETED, THE RESULTS OF THE BITSWAP ARE INDICATED.

11-Oct-82 15:50:12 - BIT SUBSTITUTION - EVERYTHING COMPLETED NORMALLY WHEN CORRECTABLE ERROR REPORTING IS DISABLED FOR A SPARE BIT SUBSTITUTION RAM ADDRESS.

11-Oct-82 15:50:13 - IN CONTROLLER 10, GROUP 0, VECTOR 3, THE ICE BIT IS GOING.

10.5.2 Parity Error

Example

```
11-OCT-82 19:20:02
* PARITY ERROR AT ADDRESS 1001475, BLOCK 0
* STORAGE MODULE SERIAL NUMBERS BY FIELD: 0 = 7460209 1 = 7460108
2 - 7460117 3 = 74
```

10.5.3 Serious MG20 Hardware Failures

Example

```
*11-OCT-82 19:20:02
* TGHA HAS TEMPORARILY CORRECTED A SERIOUS MOS MEMORY FAILURE.
*** CALL FIELD SERVICE TO REPORT THIS CONDITION ***
```

This message also appears on the CTY. Although the error has been corrected using the spare bits, the probability of a parity error due to further hardware degradation has risen to an uncomfortable state. The MG20 diagnostics should be run and the offending storage module replaced.

10.5.4 Use of Spare Bit by KLI

Example

```
5-Aug-79 13:20:29
THE MEMORY BOOT IN KLI HAS USED THE SPARE BIT TO PREVENT A PARITY
ERROR.
THIS CONDITION SHOULD BE CORRECTED AS SOON AS POSSIBLE.
CONTROLLER       GROUP   BLOCK   WORD (BITS 33-35)
10               0       1       5
```

10.5.5 KLI Detected Bad Block

If KLI has not configured a block of MG20 memory because of an error that cannot be repaired using the spare bit, the block will not be configured. This message also appears on the CTY. An entry will be made in the trace file as follows:

Example

```
*5-Aug-79 13:40:51
* THE FOLLOWING BLOCKS ARE MARKED AS BAD
* AND ARE NOT ON LINE:
*  CONTROLLER  GROUP   BLOCK
*       10     1       3
*       10     2       1
* THIS CONSISTS OF 128K OF MEMORY THAT IS OFF LINE.
*** CALL FIELD SERVICE TO REPORT THIS CONDITION ***
```

10.5.6 Missing Database

If TGHA finds that there is no database and that an MG20 controller was last initialized by TGHA, TGHA will output the following message:

Example

```
15-Aug-1979 15:10:20
* MG20 CONTROLLER 10 WAS LAST INITIALIZED BY TGHA AND
* TGHA HAS LOST ITS DATABASE. REFER TO TRACE OR SPARE
*  DOCUMENTATION IN TGHA.DOC FOR FURTHER DETAILS AND RECOVERY
* INSTRUCTIONS.
```

In this situation, TGHA cannot determine whether any spare bits in use were set by the memory boot in KLI or TGHA. This information was stored in the TGHA database that was lost. TGHA must therefore assume that all spare bits in use were set by KLI to prevent parity errors. This means that these spare bits will not be available for correction of further hardware failures in the MG20 memory. This is not an ideal situation for TGHA to be in, and will not rectify itself automatically. Although this situation is not

immediately critical, it degrades the performance of TGHA. The recovery procedure can be deferred to a convenient time, such as a field service PM or a scheduled system reload.

10.5.6.1 Recovery Procedure - The recovery procedure is as follows.

1. Delete TGHA.DAT from PS:<SYSTEM>.

2. Bring the system down.

3. Reconfigure the memory system by using the 'force' option in the memory configuration portion of KLI.

4. Reboot the system.

This is the ONLY time that the TGHA database file (TGHA.DAT) should be explicitly deleted. Although this destroys all of the known error history, the known errors will be resolved by TGHA in the same manner as before.

10.5.7 Software State Word
The software state of the memory indicates whether KLI or TGHA was the last program to make any changes in the BITSUB RAM. Although the software bits in SBUS function were designed for this purpose, conflicts made them impractical. The information is flagged in a word in the BITSUB RAM instead of using the software state bits. If, when TGHA goes to check the state of the software state word in the BITSUB RAM, TGHA finds that the data is not of the format that either KLI or TGHA uses, the following message will be put out:

Example

17-Aug-1979 13:02:59
IN CONTROLLER 10 THE BITSUB RAM STATE WORD WAS GARBAGE.
ASSUMING A DBE SCAN WAS DONE LAST. SEE TGHA.DOC FOR FURTHER DETAILS.

This is a bad state since the spare BITSUB RAM address used for the software state word is not in the address space used for swapping of memory bits. It is suggested that the MG20 diagnostics be run at the next convenient time.

10.6 SPEAR
ERROR.SYS entries made by TGHA have the following format:

10.6.1 New Known Error:
A new MG20 known error has been declared. Data: storage module serial number: 838009, block: 1, sub-block: 0, bit in field (10): 11, row: 71, column: 24, E number 192, error type: chip

10.6.2 Serious MG20 Hardware Failures

Example

```
*11-Oct-82 19:20:02
* TGHA HAS TEMPORARILY CORRECTED A SERIOUS MOS MEMORY FAILURE.
*** CALL FIELD SERVICE TO REPORT THIS CONDITION ***
```

Although the error has been corrected using the spare bits, the probability of a parity error due to further hardware degradation has risen to an uncomfortable state. The MG20 diagnostics should be run and the offending storage module should be replaced.

10.6.3 Use of Spare Bit by KLI

Example

```
5-Aug-83 13:20:29
THE MEMORY BOOT IN KLI HAS USED THE SPARE BIT TO PREVENT A PARITY
ERROR. THIS CONDITION SHOULD BE CORRECTED AS SOON AS POSSIBLE.
CONTROLLER      GROUP   BLOCK     WORD (BITS 33-35)
10              0       1         5
```

10.6.4 KLI Detected Bad Block

If KLI has not configured a block of MG20 memory because of an error that cannot be repaired using the spare bit, the block will not be configured. An entry will be made in ERROR.SYS as follows:

Example

```
*5-Aug-83 13:40:51
* THE FOLLOWING BLOCKS ARE MARKED AS BAD
* AND ARE NOT ON LINE:
*   CONTROLLER   GROUP  BLOCK
*        10      1      3
*        10      2      1
* THIS CONSISTS OF 128K OF MEMORY THAT IS OFF LINE.
*** CALL FIELD SERVICE TO REPORT THIS CONDITION ***
```

APPENDIX A
M8579 STORAGE MODULE GEOGRAPHY

Figure A-1 shows a rough sketch of the physical partitioning of the 176 MOS and 11 data multiplexer chips on each M8579. Although not drawn exactly to scale, it can be used to locate a chip or particular backplane test point for viewing the data entering (write) or leaving (read) an M8579 module. The information presented in the diagram permits the service engineer to select a test point once the following information has been determined.

Failing group of M8579s
Failing bit position (defines field)
Failing block and sub-block.

All this information is available on the error printouts furnished by the diagnostic. The failing group and field number are used to identify the MF20 slot as described below.

	Group		
Field	0	1	2
0	10	9	8
1	13	12	11
2	16	15	14
3	19	18	17

Example: Bit 22 in group 1 is failing. From Figure A-1 bit 22 is in field 2 - the above table shows we want slot 15.

To display the data input, examine pin B15P2 and the output at pin B15P1. If the printout only occurred for block 1, sub-block 2, E87, a MOS chip should be replaced. If the failure affected all blocks and all sub-blocks, the data multiplexer chip E79 might be suspected. This example assumes that the fault has already been isolated to the M8579 using directed module replacement.

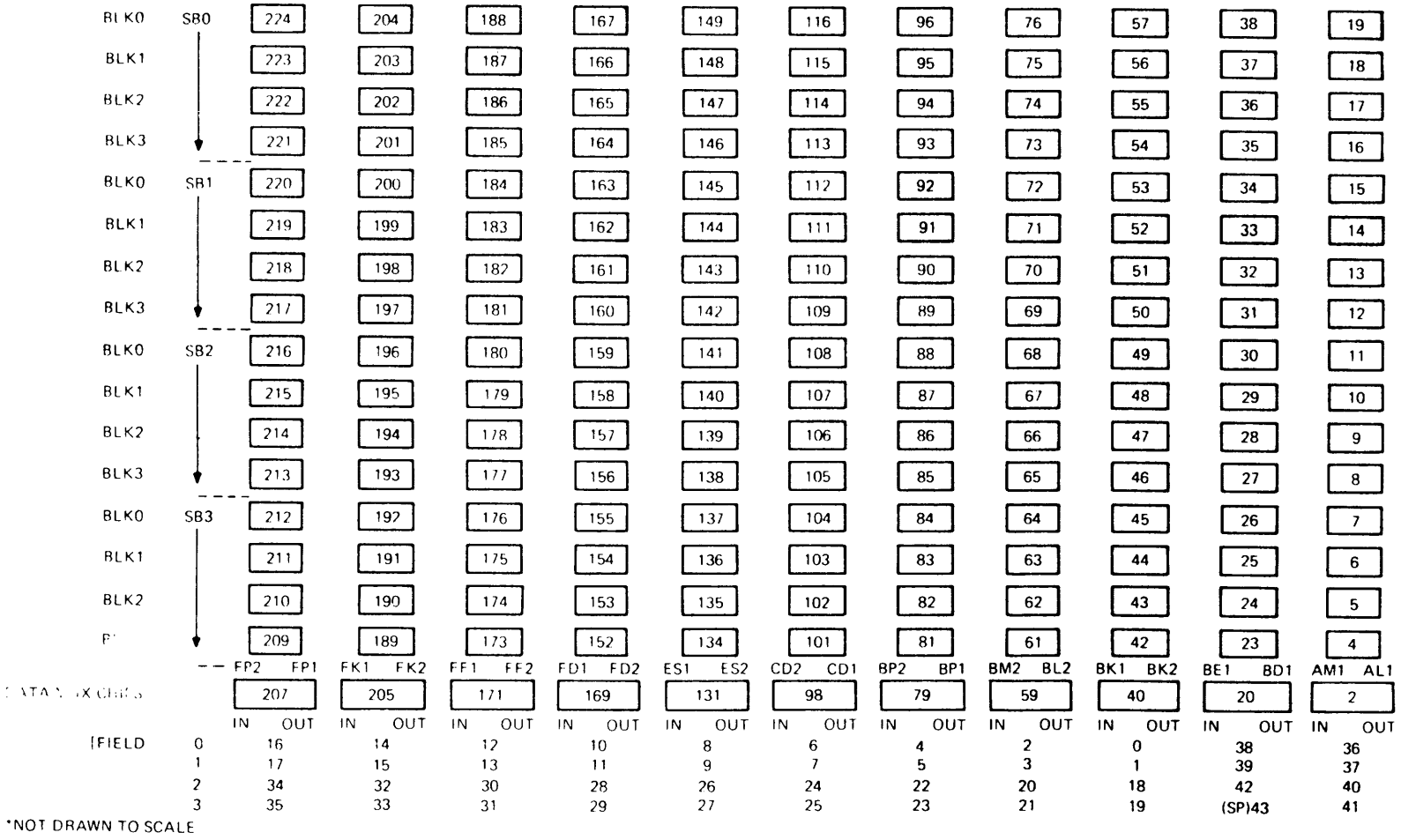


Figure A-1 M8579 Storage Module Geography

APPENDIX B
COMPLEX IC DESCRIPTIONS

B.1 INTRODUCTION

Most of the integrated circuits used to design the MF20 are standard KL10 devices. The service engineer should be able to understand their operation from the context in which they are shown in the engineering drawings. This appendix describes the operation of the more complex ICs.

B.2 DC008 DATA MULTIPLEXER CHIP (Figure B-1)

Each 8579 storage module contains 11 DC008 data multiplexer chips to buffer all information entering or leaving the MOS storage array. The DC008 is a complex 22-pin IC that consists of the following two independent sections.

1. A buffered shift register section used during a WRITE
2. A register with multiplexed outputs used during a READ

Each chip interfaces to four MOS RAM chips to permit quadword reads and writes. Block-level description of the operation of the DC008 can be divided into three modes of operation.

B.2.1 Read Operation

During a read, four MOS RAM chips are addressed and strobed for each bit position. Each RAM chip constitutes a sub-block as identified by XBUS ADR 34-35. The data bits at the outputs (pin 14) of the four MOS RAM chips feed pins 13 through 16 of the DC008 multiplexer chip. To load these four bits (bit n for WD0, 1, 2, 3) into the DC008, the signal SM LOAD RD DATA is activated which allows the next SM CLK to load the data into the data multiplexer chip. Once loaded, the data must be transferred serially to the M8575 module over a single data line MOS RD DATA Dn. To achieve this the mixer output section must be enabled and the output at pin 11 switched to reflect the state of one of four possible bits, Q0-Q3. The signal at pin 10, SM RD DATA EN, is asserted to enable the mixer, and the signals at pins 8 and 9, SM DOUT SEL 2, 1 specify which of the four words the M8575 is requesting.

B.2.2 Write Operation

During a write cycle, the information flows from the M8574 to the DC008 to the MOS RAMs. The bits are transmitted serially from the M8574 over a single data line, MOS WR DATA Dn, and shifted into the shift register section of the DC008 to position the bits properly with respect to the four RAM chips. The bits must always be sent in ascending order to ensure proper positioning.

To write a quadword the M8574 would place bit n for WD0 on the MOS WR DATA Dn line at pin 1. Following this, the signal at pin 2, SM SHIFT WR DATA H, would be asserted to permit the next SM CLK at pin 4 to shift the data up through the shift register section as follows.

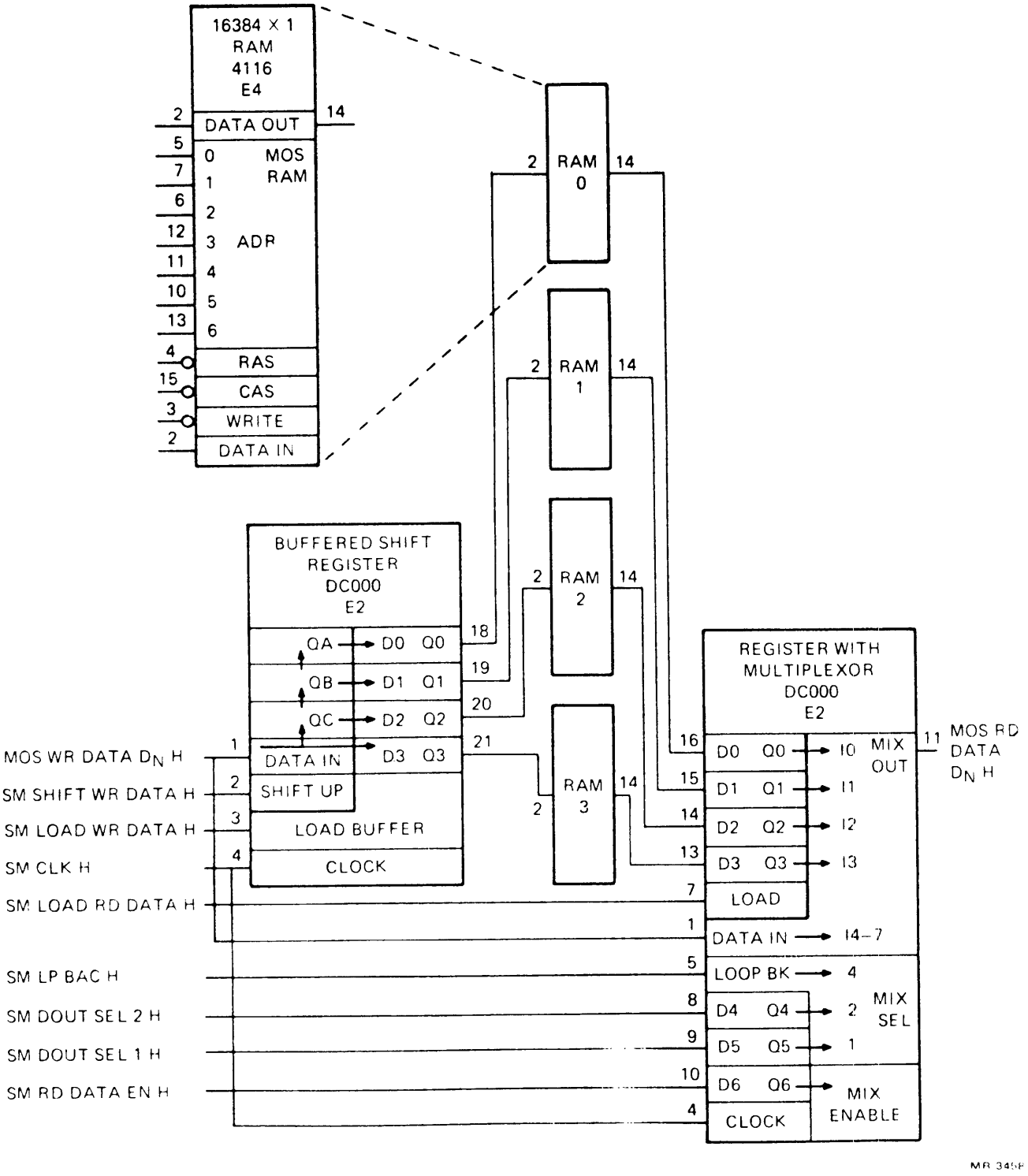


Figure B-1 DC008 Logic Block Diagram

```
QA <-- QB
QB <-- QC (WD0)
QC <-- DATA IN (pin 1 = WD1)
```

When word 2 is ready, the M8574 would place bit n for WD on the MOS WR DATA Dn line at pin 1, assert SM SHIFT WR DATA H, and shift the data up through the shift register as follows.

```
QA <-- QB WD0
QB <-- QC WD1
QC <-- DATA IN (pin 1 = WD2)
```

When the M8574 sends the last bit (bit n of WD3), it asserts SM LOAD WR DATA, which causes the following transfer to occur at the next SM CLK.

```
D0 <-- QA WD0
D1 <-- QB WD1
D2 <-- QC WD2
D3 <-- DATA IN (pin 1 = WD3)
```

At this point the four bits of the quadword are available at the output pins 18-21. From here the MOS RAM sequence writes them into the memory chips.

B.2.3 Loopback Operation

When group loopback is enabled, the signal SM LP BAC H is asserted which results in the data in at pin 1 appearing at the output pin 11. This feature provides the diagnostic program with the means of verifying the integrity of the data path from the XBus through the M8574 to the DC009s on the M8579 and from the M8579 through the M8575 and M8574 back to the XBus. Refer to Figure 4-4 for illustration of group loopback.

B.3 4116 MOS RAM CHIP (Figure B-2)

The basic storage element in the MF20 is a 16K MOS RAM chip. This paragraph describes the operation of this chip in terms of its general structure and input/output characteristics. It also includes a short discussion of MOS memory fundamentals for the reader unfamiliar with this technology.

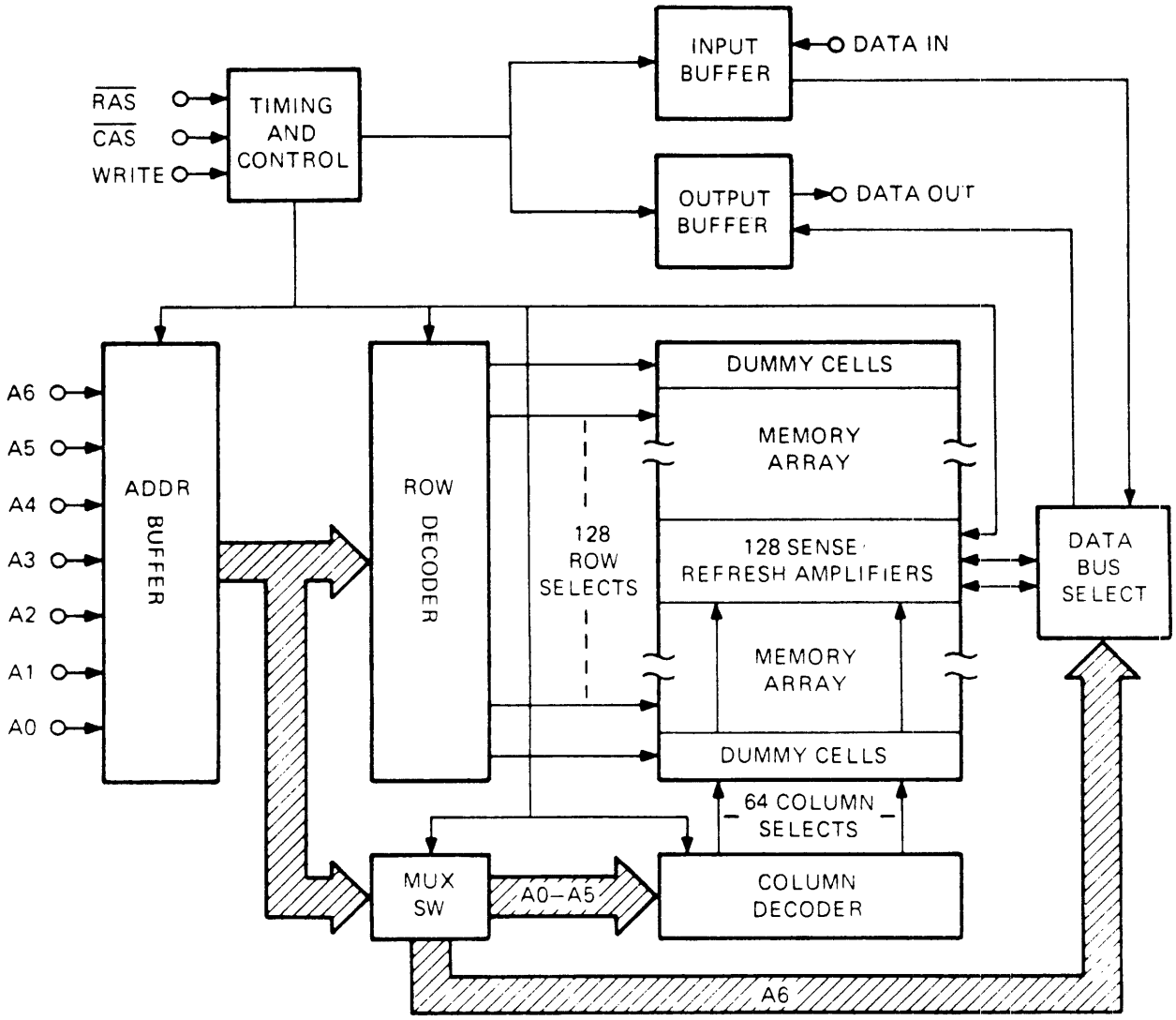
B.3.1 General Description (Figure B-2)

The 4116 is a dynamic random access memory circuit organized at 16,384 words by 1 bit. It is packaged in a standard 16-pin configuration and contains all the necessary address, control and timing, and data circuits necessary to interface the MOS storage cells to the external control logic. The 16,384 memory cells are organized into a 128 X 128 matrix. Figure B-2(A) shows a functional block of the chip organization and Figure B-2(B) shows the pin connections.

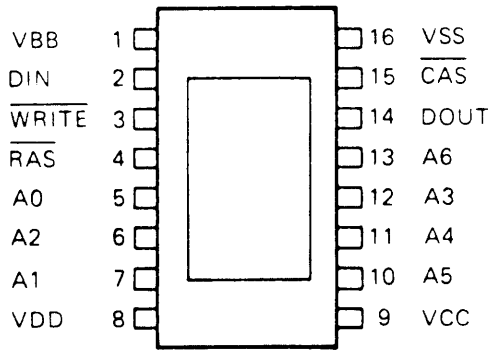
B.3.2 Basic Operation (Figure B-2, B-3)

To access a specific cell within the 128 X 128 matrix, a 14-bit address is supplied to the chip in two 7-bit slices (A0-A6). The

(A) FUNCTIONAL BLOCK DIAGRAM



(B) PIN CONNECTIONS



A0–A6	ADDRESS INPUTS
CAS	COLUMN ADDRESS STROBE
DIN	DATA IN
DOUT	DATA OUT
RAS	ROW ADDRESS STROBE
WRITE	R/W ENABLE
VBB	POWER (-5 V)
VCC	POWER (+5 V)
VDD	POWER (+12 V)
VSS	GROUND

MR-3459

Figure B-2 MOS RAM Block Diagrams

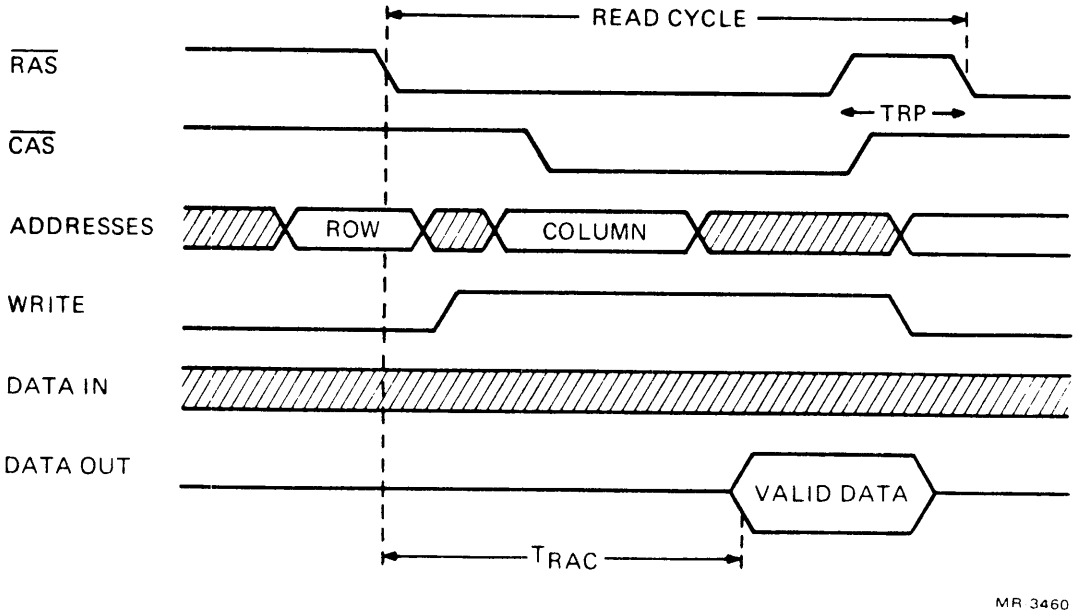


Figure B-3 RAM Timing

first 7-bit slice selects one of 128 rows and is latched into internal address buffers built into the chip by the signal RAS (Row Address Strobe) which accompanies the row address. After allowing sufficient time to latch the row address, the second 7-bit slice selects two of 128 columns (A0-A5) and is also latched into internal address buffers, this time by the signal CAS (Column Address Strobe) which accompanies the column address. At this point two of the 16,384 binary cells in the matrix are selected for sampling, and the A6 bit of the column address is used to switch a data bus selector to permit the state of a single MOS cell to appear at the data out pin from where it can be strobed into an external data buffer register. In the case of the MF20 this is the DC008 data multiplexer chip on the M8579 modules. If a write operation were to be performed, the WRITE signal would be turned on to allow writing the signal at the data in pin into the selected cell. Figure B-3 shows the typical timing sequence generated to access the MOS. T_{rac}, access time from RAS, is typically less than 150 ns in most MOS chips. T_{rp}, RAS precharge time, limits the maximum rate of access and is typically on the order of 100 ns minimum.

B.3.3 Simplified Circuit Description (Figure B-4)

Each binary cell within the 16,384-cell-square matrix consists of a single MOS FET connected to a capacitor. This is shown by TA, CA, and TB, CB in Figure B-4. When the charge on CA (cell C62) is to be sampled, the row decode signal would enable 128 FETs, one of which is TA. TA connects the charge or CA to the refresh amplifier from where it is fed through TX to the sense amplifier SA. From here the signal A would allow latching the signal into the data-out latch. Note how the FET TX is enabled by the column decoder. An alternate path also exists from CB (cell C126) through TB to a second refresh amplifier, through TY and a second sense amplifier up to an AND gate that must be enabled by the signal A being asserted. Since the signal A can only be in one state, only one sense amplifier can be selected.

Since the information within the MOS array is stored as the charge on a capacitor, it is possible that the information will be lost due to leakage discharge. To prevent this from happening each cell must be refreshed (recharged) periodically. This is achieved by addressing each row (128 cells) and activating a timing signal Tn that enables the FETs TC, TD, etc. (128 of them) and allows restoring the charge on the MOS capacitive cells. The time period during which all cells must be refreshed is called the refresh interval and is approximately 2 milliseconds for the MF20 MOS chips. Since there are 128 rows and each row must be refreshed once per 2 milliseconds, this means that the external control logic must initiate a refresh cycle every 15.6 microseconds to prevent loss of information.

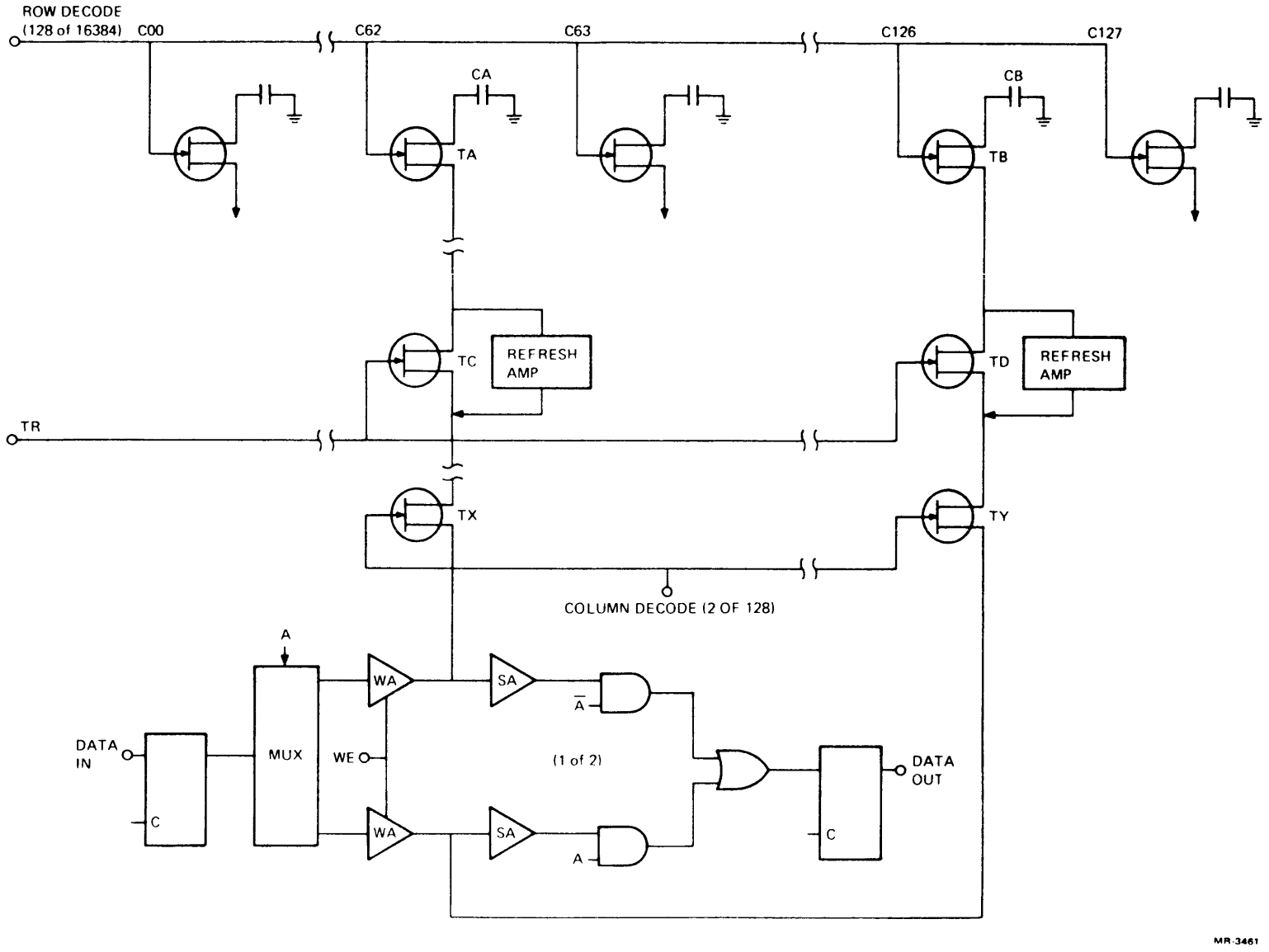


Figure B-4 16K MOS RAM Chip - Simplified Block Diagram

C.1 11-BASED 10 DIAGNOSTIC TEST STRUCTURE

The heart of the MF20 diagnostic test strategy is the 11-based diagnostic program. Three programs reside in the PDP-11 front-end console processor and are used to test and control the MF20 MOS storage subsystem on the KL10 side. The programs are DHKBF, DHKBG, and MEMCON. They all utilize sophisticated programming techniques involving interprocessor communication between the KL10 and PDP-11 via the DTE20. The purpose of this appendix is to describe the general test structure and testing sequence as the user would find in the program listings. Test 15 from DHKBF is used as an example to illustrate how to interpret the listing. Examples 1 and 2 are excerpts from the actual MACRO11 listings. They will be used to describe the two major parts of each test, the 11-based control routines and the 10-based MF20 test code.

C.2 11-BASED CONTROL ROUTINE (Example 1)

Each test begins with a brief description of the purpose of the test that includes an explanation of how it works. Part of the error message displayed when the program detects an error is the test number which points the user to this test in the listing. The first step in troubleshooting any problem is to read this information as part of analyzing the fault symptoms.

Following the abstract is a small piece of code tagged I.15. This is the initialization code that establishes the initial conditions required to run the test. First the program jumps to an initialization routine that sets up the required conditions. It then calls a routine that loads the 10-based test code from 11 core into the KL10's AC blocks. This call has the following format.

```
MABL <ECRLM0,ECRLM1,ECRLM2> ,ECRAC0,0400
```

where: MABL is a macro call to the routine designed to load the code into the KL10.

ECRM0 1 2 are the three bit masks that specify which ACs are to be loaded in AC blocks 0, 1, and 2 respectively.

ECRAC0 specifies the starting address in 11 core where the routine will find the binary image of the KL10 test code.

0400 indicates that the 10-based test code is to be started at AC4 of block 0 when the test is run.

After initialization and KL10 loading the test returns to the test dispatcher via an RTS instruction.

Next the dispatcher calls the actual test code that begins at the tag T.15 (ECRSTR:). The LOOP call sets up the specific test parameters for each iteration through the loop. Then, after initializing the error stack pointer, a JSR is executed that calls a routine to initialize any ACs not set up by the MABL call (deferred loads) and starts the test running on the 10-side. At this point the 11-based code waits for the KL10 to execute the test code out of the AC blocks. The KL10 will signify that it has completed the test by HALTing with an error code number in the KL10's PC. The 11-based control routine senses that the KL10 has halted and retrieves its PC. It then executes the conditional GOTO call as described below.

```
CGOTO   <5$,1$,2$,3$,4$>
```

where: 5$ is where the program goes if no error is detected. PC=0

1$ is where it goes for ERROR 1 halt.

2$ is where it goes for ERROR 2 halt.

3$ is where it goes for ERROR 3 halt.

4$ is where it goes for ERROR 4 halt.

At 1$, 2$, 3$, and 4$ there are DUMP and FAULT calls to subroutines that will retrieve and display the error status captured by the KL10 in its ACs. For example the call, MOSDMP AC15,AC16, would retrieve the contents of AC15 and AC16. At 5$ the program executes an ENDLOOP call that invokes a routine to one of the following.

1. Return control to the test dispatcher.

2. Return control to the beginning of test 15, at T.15, to execute the next iteration.

C.3 10-BASED TEST ROUTINES (Example 2)

The 10-based code begins with a brief abstract to describe the operation of the AC block programs. It generally includes directions that describe how to modify the AC block program to set up tight oscilloscope loops. Depending upon the test being analyzed, it may also suggest which logic modules to replace. A dump of each individual AC block follows the abstract. Although a MACRO11 listing is being read, the instructions are shown in MACRO10 format. Each line of code is commented to indicate the following information.

1. The AC that contains the instruction

2. A brief description of the functio

3. An error call number indication

4. Entry points when control is passed from one AC block to another

A special diagnostic instruction called the JAB is used to pass control between AC blocks. The format is:

```
JAB,,,0610      where the 61 indicates that control is to be
                transferred to AC6 in block 1.
```

The information shown in the listing can be compared with that which is actually in the AC blocks at the time of the error by using the DA command in any of the three 11-based programs that support the MF20. However, some of the ACs may have been altered by deferred loads and therefore not agree with the information shown in the listing.

Example 1 11-Based Control Routine

```
.SBTTL TEST 15. ECC COMPLEMENT REGISTER FUNCTION TEST.

. REM     %

THIS IS THE ECC COMPLEMENT REGISTER FUNCTION TEST.  ITS PURPOSE IS
TO TEST THE ABILITY TO READ AND WRITE THE ECC COMPLEMENT REGISTER
AND TO TEST THE ABILITY TO LATCH AND READ THE DATA BITS 36-43
MIXER.  PRIOR TO DOING A WRITE WE LOAD AND TEST THE ECC COMPLEMENT
EVEN BIT ON EVERY LOOP.  WE THEN ENABLE LATCHING OF THE D36-43
MIXER AND PERFORM A WRITE WITH A DATA PATTERN WHICH WOULD NORMALLY
GENERATE AN ECC OF ALL 0S.  THIS ALLOWS US TO READ THE ECC CHECK
DATA WHICH IS BEING WRITTEN OUT TO MEMORY.  WE THEN READ THE
LATCHED MIXER AND CHECK THAT THE ECHOED DATA MATCHES THE DATA WE
PLACED THE ECC COMPLEMENT REGISTER (GOOD PARITY) OR THE SAME DATA
WITH THE 2 LEFTMOST BITS COMPLEMENTED (BAD PARITY).  AFTER EACH
WRITE WE ALSO VERIFY THAT THE WRITE PARITY ERROR FLAG IS AT THE
CORRECT STATE.

          %

I.15:
ECRINI:   JSR     R5,INIT         ;DO COMMON INIT
          MABL    <ECRLM0,ECRLM1,ECRLM2>,ECRAC0,0400
          RTS     PC

I.15:
ECRSTR:
          LOOP    ALLMF,CMRG+CFVR+CTIM+CSPRD+C16KR+CLR05+CEN
;-----WE DUMP THE CONTROLLER NUMBER ONTO THE ERROR STACK IN CASE
;     WE DO GET AN ERROR.
          MOV     ERSI..,ERSP.. ;RESET WORKING ERROR STACK
          STKMEP  TCN           ;DUMP THE CONTROLLER NUMBER
          JSR     PC,DLDRUN     ;DO DEFERRED LOAD AND RUN AC PROG
          CGOTO   <5$,1$,2$,3$,4$>
1$:       MOSDMP  AC15,AC16
          FAULT   <ECC COMP REG LOAD ERR\>
2$:       MOSDMP  AC13,AC14
          FAULT   <D36-43 MIXER NOT CORRECT AFTER WRITE\>
3$:       COPYAC  BLK2,AC17,DBUF
          JSR     PC,CMDBUF     ;COMPLEMENT IT
          MOSDMP  DBUF,AC16,B03MSK
          FAULT   <WRITE PAR ERR FLAG IS WRONG\>
4$:       MOSDMP  ZERO...AC0,6$
          FAULT   <CANNOT CLR ERR FLGS\>

5$:       ENDLOOP
          NORMAL

6$:       WD36    0077,7777,7777  ;FCT 0 ERR MASK
```

Example 2 10-Based Test Routine

THIS IS THE 10 SIDE CODE FOR THE ECC COMPLEMENT REGISTER FUNCTION TEST. IT IS LOADED INTO AC BLOCKS 0-2.

THERE IS A DOUBLE PATH THROUGH THIS CODE WHICH IS DETERMINED BY WHETHER GOOD OR BAD PARITY IS WRITTEN INTO THE ECC COMPLEMENT REGISTER. WE SWITCH FROM PATH TO PATH ON EVERY PASS AND MODIFY THE ECC COMPLEMENT REGISTER DATA ON EVERY OTHER PASS (EXCEPT FOR ECC COMP, EVEN WHICH IS TOGGLED ON EVERY PASS). TO ENSURE THAT THE TEST STAYS IN STEP WITH THE DATA, WE ALWAYS MODIFY AN EVEN NUMBER OF BITS IN THE ECC COMP. REGISTER DATA.

SCOPE LOOP

ERROR 1 (ECC COMP REG LOAD ERR) - PLACE A JRST,,,4 IN AC 5 OF AC BLOCK 0 AND START THE LOOP AT AC 4.
ERROR 2 (D36-43 MIXER NOT CORRECT AFTER WRITE) - PLACE A JAB,,,1200 IN AC 3 OF AC BLOCK 1 AND START THE LOOP IN AC 12 OF AC BLOCK
ERROR 3 (WRITE PAR ERR FLAG IS WRONG) -
ERROR 4 (CANNOT CLR ERR FLGS) - PLACE A JRST,,,12 IN AC 13 OF AC BLOCK 2 AND START THE LOOP AT AC 12.

BLK10
IN AC BLOCK 0 WE INITIALIZE THE ECC COMPLEMENT REGISTER AND READ IT BACK. WE CHANGE THE ECC COMPLEMENT REGISTER DATA ON EVERY OTHER PASS, BUT ON EACH PASS WE COMPLEMENT THE ECC COMPLEMENT EVEN BIT. THIS ALTERNATELY GENERATES GOOD AND BAD PARITY IN THE ECC COMPLEMENT REGISTER.

AFTER SETTING THE ECC COPM. REG. WE ENABLE LATCHING OF THE D36-43 MIXER USING AN SBDIAG FUNCTION 6,7.

```
ECRAC0:
     WD36  0002,1000,0000      ;00 - TEST INCR
     I10   ADD,16,,0           ;01 <ENTRY> - INCR TEST
     I10   TLNE,16,,004000     ;02<ENTRY> - ARE ALL TESTS DONE
     I10   JRST,4,,0           ;03 <HALT-NORMAL> - ALL TESTS DONE
     I010  SBDIAG,,,16         ;04 <START> - LOAD THE ECC COMP REG
     I10   HLLZ,15,,16         ;05 - CALC THE EXPECTED ECHO
     I10   TLZ,15,,760004      ;06 - TURN OFF EXTRA BITS
     I10   CAME,17,,15         ;07 - TEST THE ECHO
     I10   JRST,4,,1           ;10 <ERROR 1> - FCT 06,4 ERR ON ECHO
     I10   TLC,16,,10          ;11 - SET UP FOR NEXT TEST
     IO10  SBDIAG,,,14         ;12 - EXECUTE THE 06.7
     I10   JAB,,,0610          ;13 -
     WD36  0000,0000,3406 SDW  ;14 - SBDIAG FCT 06.7
                               ;15 - SBDIAG ECHO/TEMP STORE
     WD36  0000,0400,2006 SDW  ;16 - SBDIAG FUNCT 06.4 (INIT)
                               ;17 - SBDIAG ECHO
ECRLM0=057777                  ;AC BLK LOAD MASK
```

Example 2 (Cont)

IN THIS AC BLOCK WE FIRST PERFORM A WRITE AND THEN WE READ THE D36-43 MIXER. IF THE PARITY OF THE ECC COMPLEMENT REGISTER WAS BAD, WE FIRST COMPLEMENT THE EXPECTED VALUE OF ECC CHECK 32 AND 16 BEFORE TESTING AGAINST THE ACTUAL.

```
ECRAC1:

                                ;00 [UNUSED]
     I10    TLC,13,,003000      ;01 - COMPLEMENT THE ECC 32 AND 16
     IO10   SBDIAG,,,14         ;02 - EXECUTE THE 06.6
     I10    CAME,15,,13         ;03 - DOES ECHO MATCH
     I10    JRST,4,,2           ;04 <ERROR 2> - D36-43 MIXER ERROR
     I10    JAB,,,0521          ;05 -
     I10    MOVEM,17,,20        ;06 <ENTRY> - DO A WRITE
     I10    TRC,12,,3           ;07 - SET PROPER JRST
     I10    PXCT,4,,16          ;10 - GET THE EXPECTED ECC
     I10    TLZ,13,,10          ;11 - WE DO NOT EXPECT THIS BIT
                                ECHOED
     I10    JRST,,,1,0 RELOAD   ;12 - GO TO THE PROPER TEST
                                ;13 - WILL GET EXPECTED ECC
     WD36   0000,0000,3006 SDW  ;14 - SBDIAG FCT 06.6
                                ;15 - SBDIAG ECHO
     I10    MOVE,13,,17         ;16 - INST TO GET EXPECTED ECC
     WD36   4000,0040,0100      ;17 - DATA TO PROD ECC OF ALL 0S
ECRLM1=153776                   ;AC BLK LOAD MASK

;-----IN AC BLOCK WE CHECK THE WRITE PARITY ERROR FLAG FOR THE
PROPER VALUE.

ECRAC2:
     WD36   0100,0000,0000 SDW      ;00 - SBDIAG FCT 0 CLR
                                    ;01 - SBDIAG ECHO
                                    ;02 - [UNUSED]
                                    ;03 - [UNUSED]
                                    ;04 - [UNUSED]
     I10    TRC,15,,300             ;05 <ENTRY> - SET PROPER EXIT
     IO10   SBDIAG,,,16             ;06 - DO A FCT 0
     I10    TLNE,17,,20000,0 RELOAD ;07 - IS ERR FLG CORRECT
     I10    JRST,4,,3               ;10 <ERROR 3> - ERR FLG WRONG
     I10    TLC,7,,4000             ;11 - SET TEST FOR NEXT LOOP
     IO10   SBDIAG,,,0              ;12 - TRY TO CLR THE ERR FLG
     I10    TLNE,1,,770000          ;13 - ARE ERR FLGS CLR
     I10    JRST,4,,4               ;14    ROR 4> - FLAGS NOT CLRD
     I10    JAB,,,0101,0 RELOAD     ;15
     WD36   0000,0000,0000 SDW      ;16     DIAG FCT 0
                                    ;17 - [illegible]DIAG ECHO
ECRLM2=077741                       ;AC BL[illegible]OAD MASK
     ENDBLK
.EVEN
```