

pdp11

**MF11S-K
MOS memory
user's guide**

digital

**MF11S-K
MOS memory
user's guide**

First Edition, October 1977

The material in this manual is for informational purposes and is subject to change without notice.

Copyright © 1977 by Digital Equipment Corporation

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC	DECtape	PDP
DECCOMM	DECUS	RSTS
DECsystem-10	DIGITAL	TYPESET-8
DECSYSTEM-20	MASSBUS	TYPESET-11
		UNIBUS

CONTENTS

	Page
CHAPTER 1 INTRODUCTION	
1.1 GENERAL FUNCTIONAL DESCRIPTION	1-1
1.2 PHYSICAL DESCRIPTION	1-1
1.2.1 The Control Module (M7893).....	1-1
1.2.2 The Array Module (M7984).....	1-1
1.2.3 MF11S-K Memory Size.....	1-1
1.2.4 Internal Memory Bus	1-1
1.3 MAINTENANCE AND RELIABILITY	1-1
1.4 BATTERY SUPPORT MODE.....	1-5
1.5 SPECIFICATIONS	1-5
1.5.1 Unibus Interface.....	1-5
1.5.2 Power Supply Requirements	1-5
1.5.3 Access and Cycle Times	1-9
1.5.3.1 Memory Storage.....	1-9
1.5.3.2 Refresh Cycle	1-9
1.5.3.3 ECC INIT Routine.....	1-9
1.5.3.4 Control and Status Register (CSR)	1-10
1.6 BACKPLANE UTILIZATION.....	1-10
CHAPTER 2 FUNCTIONAL DESCRIPTION OF THE ERROR CORRECTION LOGIC	
2.1 GENERAL.....	2-1
2.2 MODIFIED HAMMING CODE AND CHECK BITS.....	2-1
2.3 MEMORY WORD SIZE	2-2
2.4 USE OF THE 39-BIT MEMORY WORD.....	2-2
2.5 READ OPERATION (DATI AND DATIP).....	2-2
2.5.1 Accessing the Data	2-2
2.5.2 Generation of Syndrome Bits.....	2-2
2.5.3 Example A, Data Bit 0 Set	2-2
2.5.4 The Error Correction Process	2-7
2.5.5 Example B, Handling a Double Error	2-7
2.5.6 Handling a Check Bit Error, Example C	2-8
2.6 WRITE OPERATION IN GENERAL (DATO AND DATOB).....	2-8
2.7 WRITE OPERATION FOLLOWING A DATIP.....	2-9
2.7.1 No Single Error Detected.....	2-10
2.7.2 Single Error Detected	2-10
2.8 WRITE OPERATION NOT FOLLOWING A DATIP.....	2-10
2.8.1 Error Detection on a Write	2-13
2.8.2 Double Error Condition	2-13
2.8.3 Single Error Condition	2-13

CONTENTS (CONT)

	Page
CHAPTER 3	INSTALLATION OF THE MF11S-K MEMORY
3.1	GENERAL.....3-1
3.2	DD11-K BACKPLANE SLOT ALLOCATION.....3-1
3.3	MF11S-K OPTIONS3-2
3.3.1	MF11S-K (32K MOS System Complete)3-2
3.3.2	MS11-KA (32K × 16 MOS Array Module)3-2
3.3.3	Options Required Per Storage Capacity3-2
3.4	MIXING THE MF11S-K AND OTHER MEMORIES3-7
3.4.1	Address Space Above MF11S-K.....3-7
3.4.2	Address Space Below MF11S-K3-8
3.5	SUPPLY VOLTAGES.....3-8
3.6	FIRST TIME START-UP3-8
CHAPTER 4	CONDITION STATUS REGISTER FUNCTION AND LOGIC
4.1	GENERAL.....4-1
4.2	READING FROM AND WRITING TO THE CSR4-1
4.3	CSR BIT FUNCTIONS.....4-1
4.3.1	Double Error Indication Enable (CSR Bit 0).....4-2
4.3.2	Disable Error Correction Mode (CSR Bit 1)4-2
4.3.3	Diagnostic Check Mode (CSR Bit 2).....4-3
4.3.4	Inhibit Mode Pointer (CSR Bit 3)4-3
4.3.5	Single Error (CSR Bit 4)4-4
4.3.6	Address and Check Bit Storage (CSR Bits 11–5)4-4
4.3.7	Set Inhibit Mode (CSR Bit 13)4-4
4.3.8	Double Error (Bit 15)4-4

FIGURES

Figure No.	Title	Page
1-1	Backplane with Controller and Array Modules.....	1-2
1-2	Simplified MF11S-K System Block Diagram.....	1-2
1-3	MF11S-K Control Module.....	1-3
1-4	MF11S-KA Array Module.....	1-4
2-1	Modified Hamming Code Matrix.....	2-1
2-2	Read Cycle Flowchart.....	2-3
2-3	ECC Logic Used on a Read Cycle Block Diagram.....	2-5
2-4	Syndrome Bit Decoding.....	2-7

FIGURES (CONT)

Figure No	Title	Page
2-5	Write Cycle Flowchart	2-9
2-6	ECC Logic Used on a Write Cycle	2-11
2-7	Read-Modify-Write Cycle Flowchart	2-12
3-1	DD11-L Backplane Utilization.....	3-3
3-2	DD11-K Backplane Unibus Routing.....	3-5
3-3	Fully Configured DD11-K Backplane	3-6
4-1	CSR Bit Allocation	4-2

TABLES

Table No.	Title	Page
1-1	Backup Battery Power Requirements	1-5
1-2	Unibus Loading	1-5
1-3	Power Supply Requirements.....	1-6
1-4	System Power.....	1-7
1-5	Memory Storage – Access and Cycle Times	1-9
1-6	Refresh Repetition Rate.....	1-9
1-7	CSR Access	1-10
2-1	Syndrome Bit Generation.....	2-4
2-2	Error Correction, Example A	2-4
2-3	Double Error, Example B.....	2-8
2-4	Check Bit Error, Example C.....	2-8
3-1	Starting Address Selection.....	3-7
4-1	CSR Address Modification	4-1
4-2	Data Modification Through the Check Bits	4-3

CHAPTER 1

INTRODUCTION

1.1 GENERAL FUNCTIONAL DESCRIPTION

The MF11S-K memory is an MOS, semiconductor, random access memory (RAM) designed to interface with the PDP-11 Unibus. It utilizes an error correction scheme that makes the MF11S-K more reliable than standard memory systems. The memory storage matrix is arranged in 39-bit words: two 16-bit PDP-11 words plus seven check bits. The check bits are used to indicate the integrity of the two 16-bit words. The error correction algorithm allows any single bit error in the entire 39-bit configuration to be detected and corrected. Two simultaneous errors in the same 39-bit word are detected but not corrected and will be treated as a fatal error. A fatal error of this type causes a trap procedure identical to a parity trap. Single errors are transparent to the Unibus.

1.2 PHYSICAL DESCRIPTION

One control module (M7983) and from one to four array modules (M7984) mounted in a dedicated 9-slot backplane (DD11-K) make up the MF11S-K MOS memory system (Figure 1-1). Figure 1-2 is a simplified block diagram of the MF11S-K system.

All backplane slots not used by the memory are available as small peripheral controller slots.

1.2.1 The Control Module (M7893)

The control module (Figure 1-3) is a hex multilayer board that contains the Unibus interface, timing control, error correction logic, the Control and Status register (CSR), MOS RAM control circuits, and refresh circuitry.

1.2.2 The Array Module (M7984)

The storage array module is also a hex multilayer board (Figure 1-4). It contains 156 MOS RAM ICs (16 pin dip, 4096 bits) and their associated drive circuitry. Although the array module is organized internally in a $16K \times 39$ bit configuration, in order to maximize the efficiency of the error correction logic, the MF11S-K memory responds to the processor like a standard $32K \times 16$ bit memory.

1.2.3 MF11S-K Memory Size

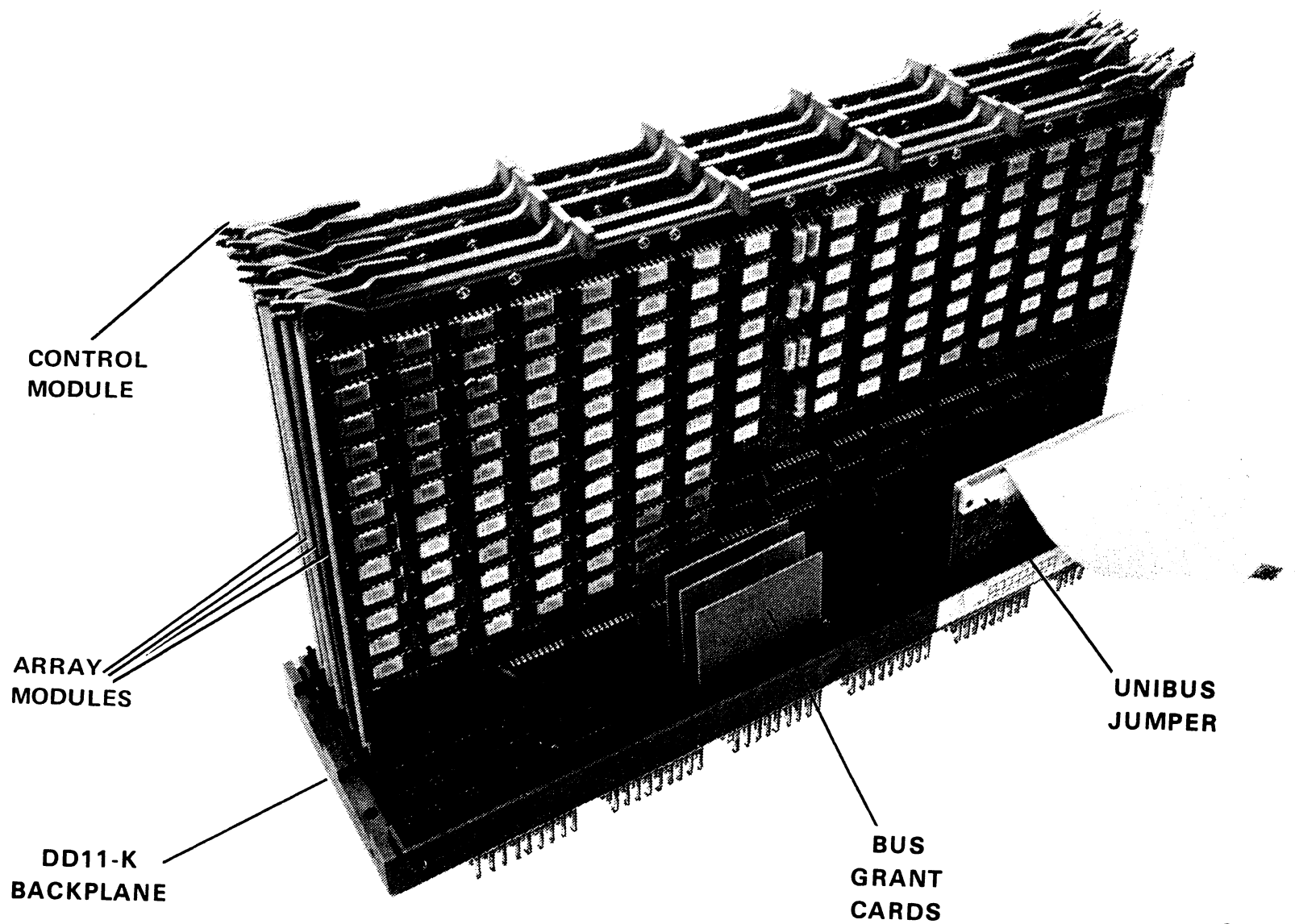
The memory can be increased in capacity from a minimum size of $32K \times 16$ bits (one control module and one array module) in 32K increments. The maximum MF11S-K MOS memory system capacity, $128K \times 16$ bits, can be obtained by using four 32K 16-bit array modules.

1.2.4 Internal Memory Bus

The MF11S-K memory utilizes an internal bus for communication between the control module and the array modules. Each module, therefore, includes drivers and receivers for the internal bus.

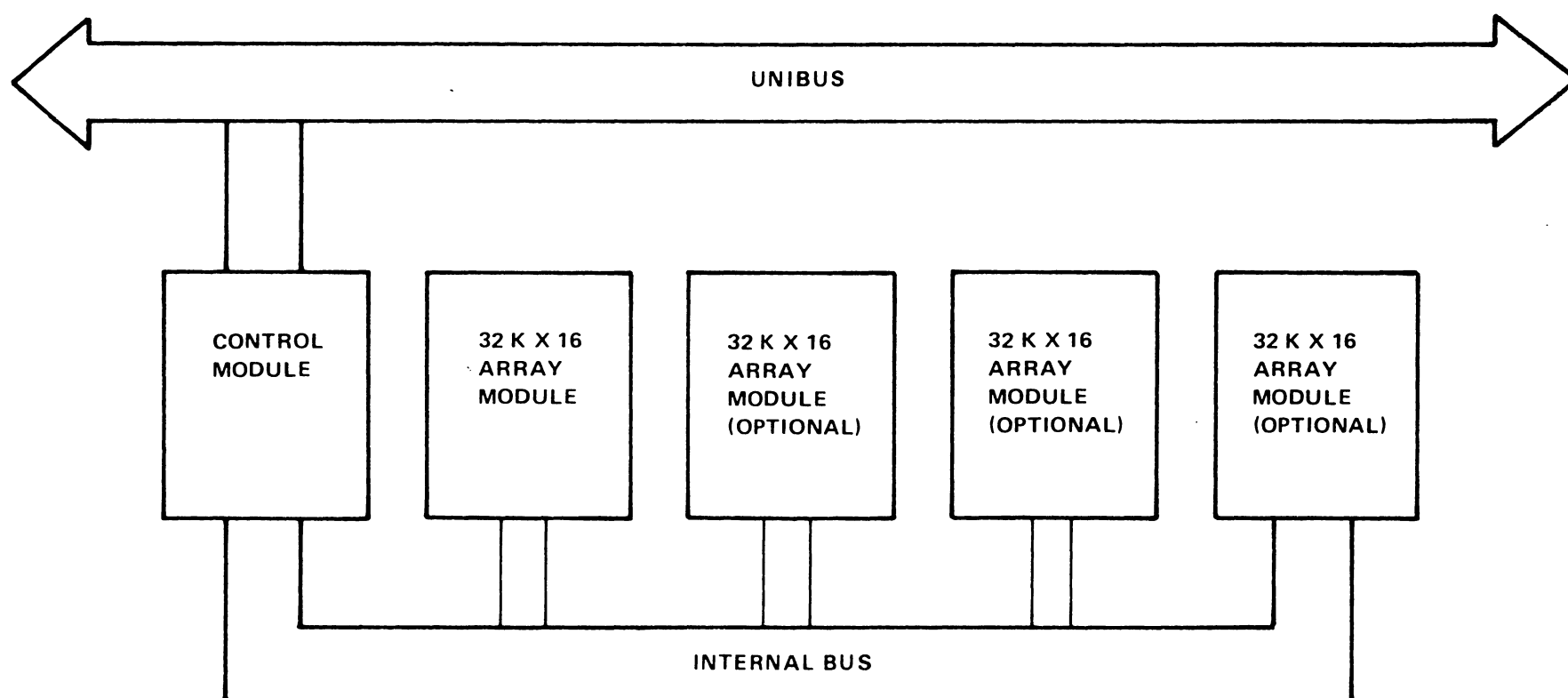
1.3 MAINTENANCE AND RELIABILITY

The error correction capability significantly increases the reliability of the MF11S-K over conventional memory systems. This is accomplished by allowing correctable faults to accumulate in the system. A service call is justified only if noncorrectable faults have been encountered. A noncorrectable fault is defined as any fault that prevents data from being stored and retrieved without apparent errors.



8539-2

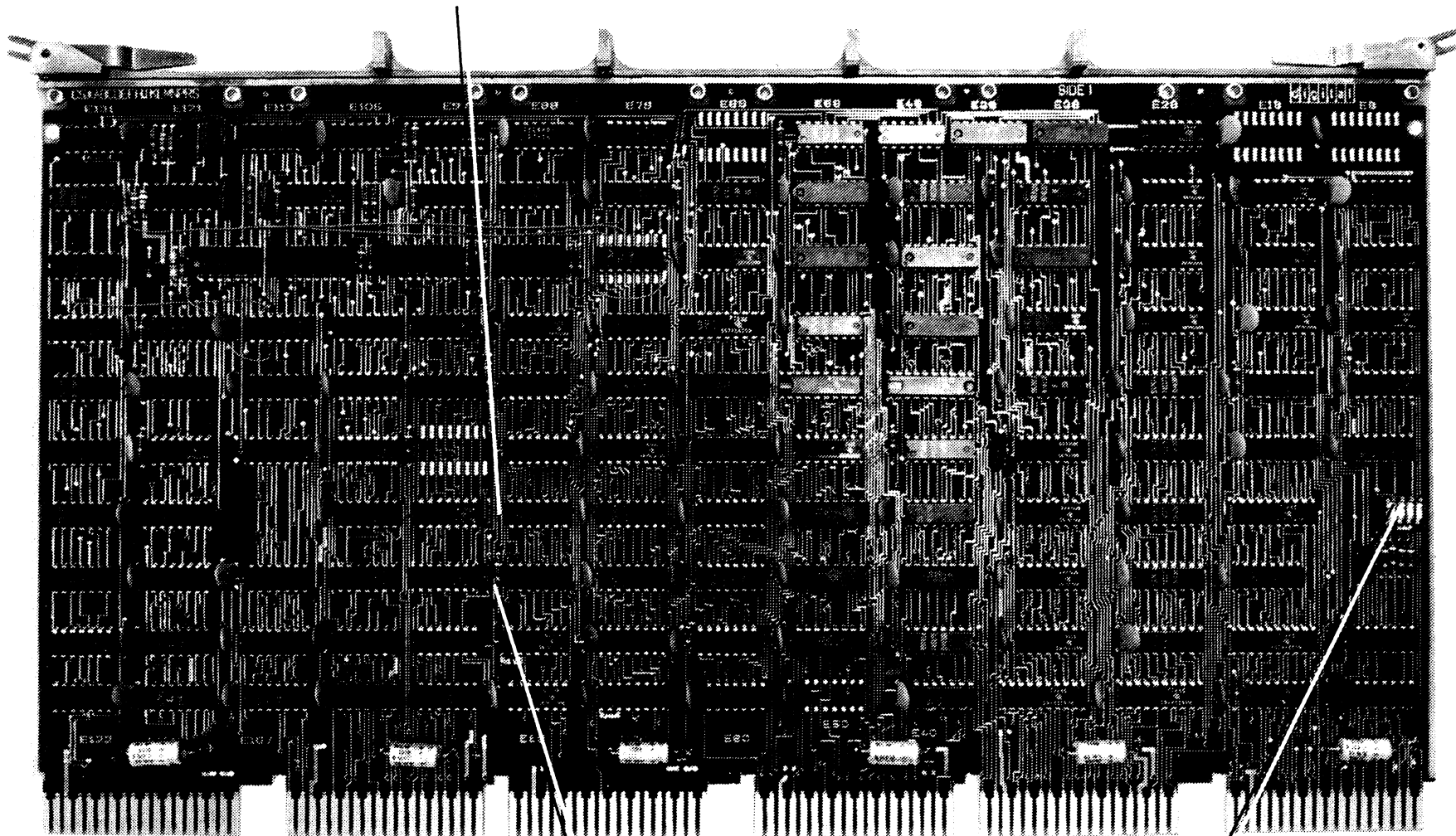
Figure 1-1 Backplane with Controller and Array Modules



11-5413

Figure 1-2 Simplified MF11S-K System Block Diagram

W2
NOT NORMALLY
USED.
WHEN INSERTED
SELECTS CSR
ADDRESS
772134



W1
NORMALLY IN
SELECTS CSR
ADDRESS
772136

ADDRESS
SWITCH
SETTING

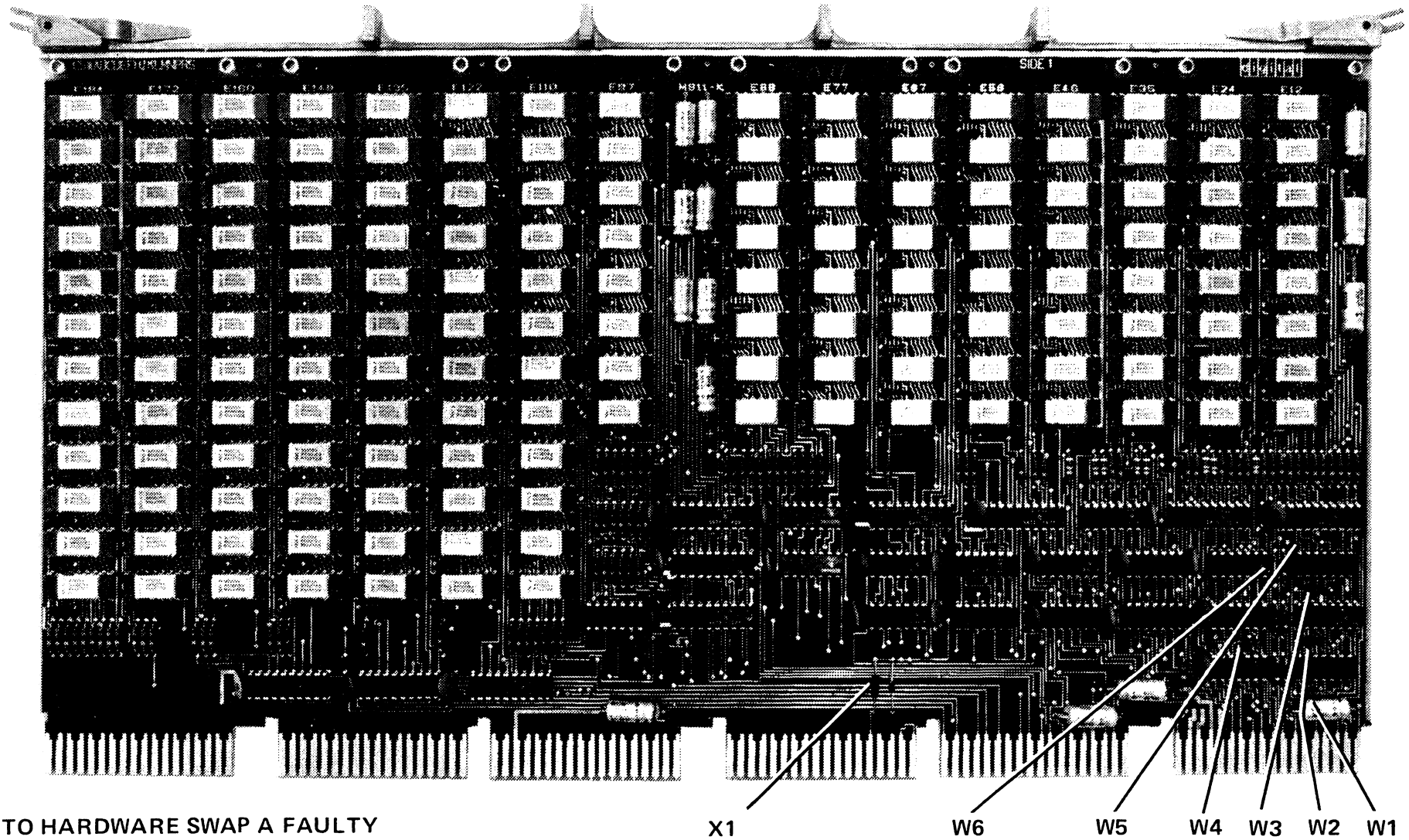
A 17 A 16 A 15 A 14

SETS
STARTING
ADDRESS
OF MEMORY

8539 7

Figure 1-3 MF11S-K Control Module

MS11KA USES JUMPERS W1, W2, W3, W4, X1
 MS11KB USES JUMPERS W1, W2, W3, W4, W6, X1



TO HARDWARE SWAP A FAULTY
 8K WORD SECTION TO THE TOP
 OF MEMORY, USE W5 AS IN THE FOLLOWING EXAMPLE.
 MOS DEVICES DRIVEN BY RAS SEL A FAULTY.
 REMOVE W1, W4, INSERT W5, CONNECT
 E1-12 TO E2-4.

X1	W6	W5	W4	W3	W2	W1
ALWAYS	USED ON	NOT				
USED	MS11KB	NORMALLY				
	ONLY	USED				

8539-9

NOTE
 The MS11-KA can be replaced by the MS11-KE.
 The latter option has faster access time MOS
 RAMs. The system access time is not reduced, how-
 ever.

Figure 1-4 MF11S-KA Array Module

1.4 BATTERY SUPPORT MODE

A battery backup option can be utilized with the MF11S-K memory. Since MOS memories lose data whenever power is removed, a backup battery is necessary if the customer wishes to ensure data integrity in the case of a power failure. The power buses on the control and array modules are partitioned so that the voltage supplied by the battery, when the power is down, powers only the logic needed to refresh the memory. Table 1-1 lists the power requirements for the backup battery.

Table 1-1 Backup Battery Power Requirements

System Capacity	Battery Backup Power Required Typ	Max
32K × 16	7.2 W	11.1 W
64K × 16	12 W	18 W
96K × 16	16.7 W	24.4 W
128K × 16	21.6 W	30.6 W

1.5 SPECIFICATIONS

1.5.1 Unibus Interface

The MF11S-K memory interfaces directly with the Unibus. The Unibus loading listed in Table 1-2 should be expected.

Table 1-2 Unibus Loading

Signal	Load
Data Lines	1 Driver, 1 Receiver
All Other Signals	1 Receiver or 1 Driver

1.5.2 Power Supply Requirements

The current drawn from the +12 V supply varies directly with the frequency of operation. The figures given in Tables 1-3 and 1-4 will be for standby (includes refresh) and a cycle time of 800 ns.

Table 1-3 Power Supply Requirements

M7983 Controller					
+5 V			+5 VBB		
Typ		Max	Typ		Max
4.0 A (20 W)		5.0 A (25 W)	0.6 A (3 W)		0.9 A (4.5 W)

MS11-KA (32K × 16-Bit Array Module)					
+5 V			+5 VBB		
Typ		Max	Typ		Max
0.3 A (1.5 W)		0.5 A (2.5 W)	0.45 A (2.25 W)		0.6 A (3.0 W)
-12 V			+12 V		
		Standby		Active	
Typ	Max	Typ	Max	Typ	Max
25 mA (0.3 W)	50 mA (0.6 W)	0.16 A (1.9 W)	0.25 A (3 W)	0.6 A (7.2 W)	1.0 A (12 W)

Table 1-4 System Power

	+5 V		+5 V _{BB}		-12 V		+12 V				Power					
	Typ	Max	Typ	Max	Typ	Max	Standby		Active		Standby		Active		Battery Backup Mode	
							Typ	Max	Typ	Max	Typ	Max	Typ	Max	Typ	Max
32K × 16	4.3 A (21.5 W)	5.5 A (27.5 W)	1.0 A (5 W)	1.5 A (7.5 W)	25 mA (0.3 W)	50 mA (0.6 W)	0.16 A (1.9 W)	0.25 A (3 W)	0.6 A (7.2 W)	1.0 A (12 W)	28.7 W	38.6 W	34 W	47.6 W	(7.2 W)	(11.1 W)
64K × 16	4.6 A (23 W)	6.0 A (30 W)	1.5 A (7.5 W)	2.1 A (10.5 W)	50 mA (0.6 W)	100 mA (1.2 W)	0.32 A (3.8 W)	0.5 A (6 W)	0.76 A (9.1 W)	1.25 A (15 W)	35 W	48 W	40.2 W	56.7 W	(12 W)	(18 W)
96K × 16	4.9 A (24.5 W)	6.5 A (32.5 W)	2.0 A (10 W)	2.7 A (13.5 W)	75 mA (0.9 W)	150 mA (1.8 W)	0.48 A (5.8 W)	0.76 A (9.1 W)	0.92 A (11 W)	1.5 A (18 W)	41.2 W	56.9 W	46.4 W	65.8 W	(16.7 W)	(24.4 W)
128K × 16	5.2 A (26 W)	7.0 A (35 W)	2.5 A (12.5 W)	3.3 A (16.5 W)	100 mA (1.2 W)	175 mA (2.1 W)	0.64 A (7.7 W)	1.0 A (12 W)	1.1 A (13.2 W)	1.75 A (21 W)	47.4 W	65.6 W	52.9 W	74.6 W	(21.4 W)	(30.6 W)

1.5.3 Access and Cycle Times

Cycle time is defined as the time from the assertion of MSYN at the output of the Unibus memory receivers (memory not busy) to the time the memory is ready to accept the next MSYN (i.e., when the Memory Busy flip-flop is cleared). Access time is defined as the time from the assertion of MSYN at the output of the Unibus receiver to the time that SSYN is asserted on the bus.

1.5.3.1 Memory Storage – Access and cycle times are listed in Table 1-5.

Table 1-5 Memory Storage – Access and Cycle Times

		Typ	Max
DATI or DATIP	Read Access, No error	520 ns	550 ns
	Read Access, Single error Corrected or Double error	630 ns	675 ns
	Read cycle time	750 ns	800 ns
DATO or DATOB	Internal RMW SSYN response	85 ns	120 ns
	Internal RMW cycle time	1130 ns	1200 ns
DATO or DATOB Following DATIP	Write SSYN Response	85 ns	120 ns
	Write cycle time	750 ns	800 ns

1.5.3.2 Refresh Cycle – Refresh cycle time: Typ 865 ns; Max 910 ns.

Refresh cycle time is defined as the time from the assertion of REF CYC (1) H to the resetting of the MBSY flip-flop.

Note that all the access and cycle times listed in Table 1-5 are modified during a refresh cycle conflict. In the worst case, for example, memory access is increased by the refresh cycle time. Read access time would be 550 ns + 910 ns = 1460 ns.

DATOB time would be 1200 ns + 910 ns = 2110 ns.

Table 1-6 lists the refresh repetition rates.

Table 1-6 Refresh Repetition Rate

Typical Rate	25.0 μ s
Fastest Rate	22.5 μ s
Slowest Rate	29 μ s

1.5.3.3 ECC INIT Routine – ECC INIT Time: Typ 65 ms; max 70 ms.

During an ECC INIT routine, all zeros data is written at an 8 μ s repetition rate, and BUS AC LO is asserted.

1.5.3.4 Control and Status Register (CSR) – CSR access times are listed in Table 1-7.

Table 1-7 CSR Access

		Typ	Max
DATI or DATIP Cycle	CSR Read Access	60 ns	80 ns
DATO or DATOB Cycle	CSR Write SSYN Response	60 ns	80 ns

The CSR can be accessed regardless of whether a memory cycle is in progress or not; e.g., the memory can be in a refresh cycle while the CSR is being accessed. The cycle time depends on the capability of the bus master.

1.6 BACKPLANE UTILIZATION

The nine slot backplane has been designed specifically for the MF11S-K memory. The backplane specifications should be examined carefully before configuration and power application.

The M7984 array module will fit in slots 4 through 7 of the DD11-K backplane. The M7983 controller, however, must be placed in slot 8. The array slots are dedicated according to memory size. The first 32K of memory must go in slot 7, the second in slot 6, the third in slot 5, and the fourth in slot 4. Slot 1 must be used for the Unibus cable-in jumper. Slot 9 must contain either a Unibus cable-out jumper or a terminator. Slots that are unused should contain bus grant cards.

For information concerning other uses of the DD11-K backplane, see Chapter 3.

CHAPTER 2 FUNCTIONAL DESCRIPTION OF THE ERROR CORRECTION LOGIC

2.1 GENERAL

When the MF11S-K memory system sends data to the bus master, it checks the integrity of the data and corrects single bit errors before placing the data on the Unibus. The error correction logic makes use of a modified hamming code to carry out these checking and correcting functions.

2.2 MODIFIED HAMMING CODE AND CHECK BITS

The modified hamming code uses combinations of parity checks on various bit patterns within a memory data word. The error correction logic implements the modified hamming code by forming a parity check bit for each of these various bit patterns. Figure 2-1 shows the matrix formed by the modified hamming code used in the MF11S-K memory.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MF11S-K WORD
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	2 PDP-11 WORDS
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	4 PDP-11 BYTES
C1	X			X			X		X			X			X		X			X			X		X			X			X		
C2	X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X			
C4	X	X	X	X					X	X	X	X					X	X	X	X					X	X	X	X					
C8	X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
C16	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X	
C32	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X									
CT	X			X		X	X			X	X		X			X		X	X		X			X		X	X		X			X	
	BYTE 3								BYTE 2								BYTE 1								BYTE 0								

11-5414

Figure 2-1 Modified Hamming Code Matrix

Each check bit, except Ct, is the result of odd parity generated on the data bits designated by X. For example, C1 is a function of data bits 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29, and 31. Ct is the result of even parity. It ensures the detection of double errors. The fact that it uses even parity ensures that an all zeros Internal Memory Bus error is detected as a double (fatal) error.

Each data bit is described by a unique combination of 3, 5, or 7 check bits. Data bit 0, for example, is described by C32, C16, and C8. No other data bit is described uniquely by these three check bits. If, in a given 32-bit data word, bit 0 only were logic 1, C16 and C8 would be logic 0s, while Ct, along with C32, C4, C2, and C1, would be logic 1s.

2.3 MEMORY WORD SIZE

Six check bits would be required to describe a 16-bit PDP-11 word. Only one additional check bit, however, is required to describe 32 data bits. The 32-bit data configuration within the MF11S-K memory thus increases the efficiency of check bit use by more than 40 percent.

2.4 USE OF THE 39-BIT MEMORY WORD

The internal operation of the MF11S-K memory differs from that of conventional memories so that it can carry out the error correction algorithm. Since check bits are generated to correspond to two data words (32 bits) whenever one word or byte is accessed in a read or write cycle, the remaining bytes in the double-word location and the seven corresponding check bits must all be examined by the error correction logic. The following paragraphs trace, in general terms, the function of the memory as it performs read and write cycles.

2.5 READ OPERATION (DATI AND DATIP)

2.5.1 Accessing the Data

When a bus master initiates a read cycle, two bytes are addressed via the Unibus (Figure 2-2). The memory control logic brings the 39-bit memory word (32 data bits plus 7 stored check bits) from the array into the Memory Data register (MDR). The 39-bit memory word includes the requested data. A parity circuit then generates seven new check bits on the 32 data bits just read from the array.

2.5.2 Generation of Syndrome Bits

At this point, two separate sets of check bits exist. Both sets describe the 32 data bits in the MDR. In order to determine the validity of the data, the error correction logic compares the stored check bits with the newly generated check bits. The Exclusive OR function of the two sets of check bits forms a set of seven syndrome bits as shown in Figure 2-3. A discrepancy between a stored check bit and a corresponding newly generated check bit will set a syndrome bit to a logic 1 (Table 2-1). After the syndrome bits are formed, they are latched. Three examples will be used to clarify the explanation.

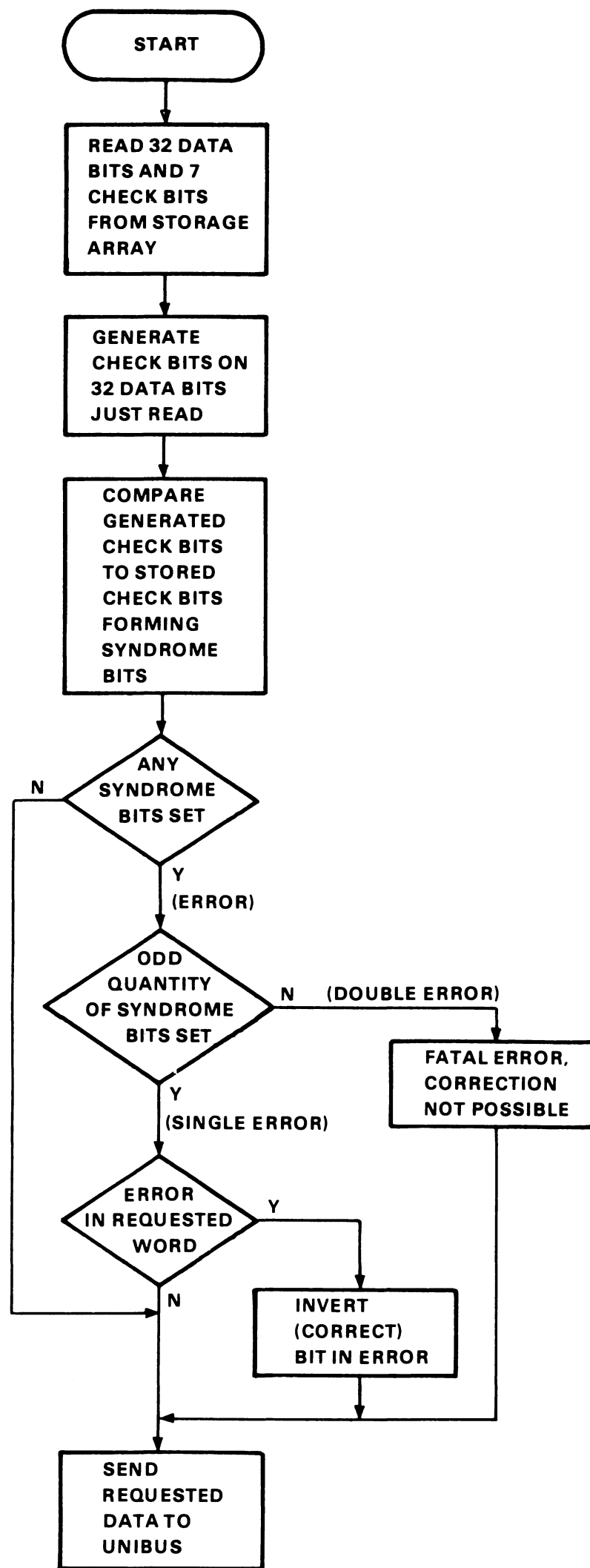
2.5.3 Example A, Data Bit 0 Set

At this point, it will be useful to follow up the example given at the end of Paragraph 2.2, in which data bit 0, only, was set. If the data that was stored in the memory location addressed is identical to the data retrieved from that location, the two sets of check bits will agree. No syndrome bits will be set to logic 1s, and when the syndrome bits are latched, all latches will be in the reset state.

However, if bit 0 is a 1 in the data stored and yet a 0 in the data that is retrieved from the array, the syndrome bits will register the discrepancy. The stored check bits, C8 and C16, will be 0s. Stored check bits Ct, C32, C4, C2, and C1 will be 1s. In the set of newly generated check bits, however, only Ct will be 0. The other newly generated check bits will be 1s (Table 2-2).

The Exclusive OR function of the syndrome generation circuit causes ST, S16, and S8 to be set and then latched as logic 1s.

An examination of the decoder chart in Figure 2-4 will show that syndrome pattern 1011000 will be decoded to point to data bit 0, within byte 0.



11-5420

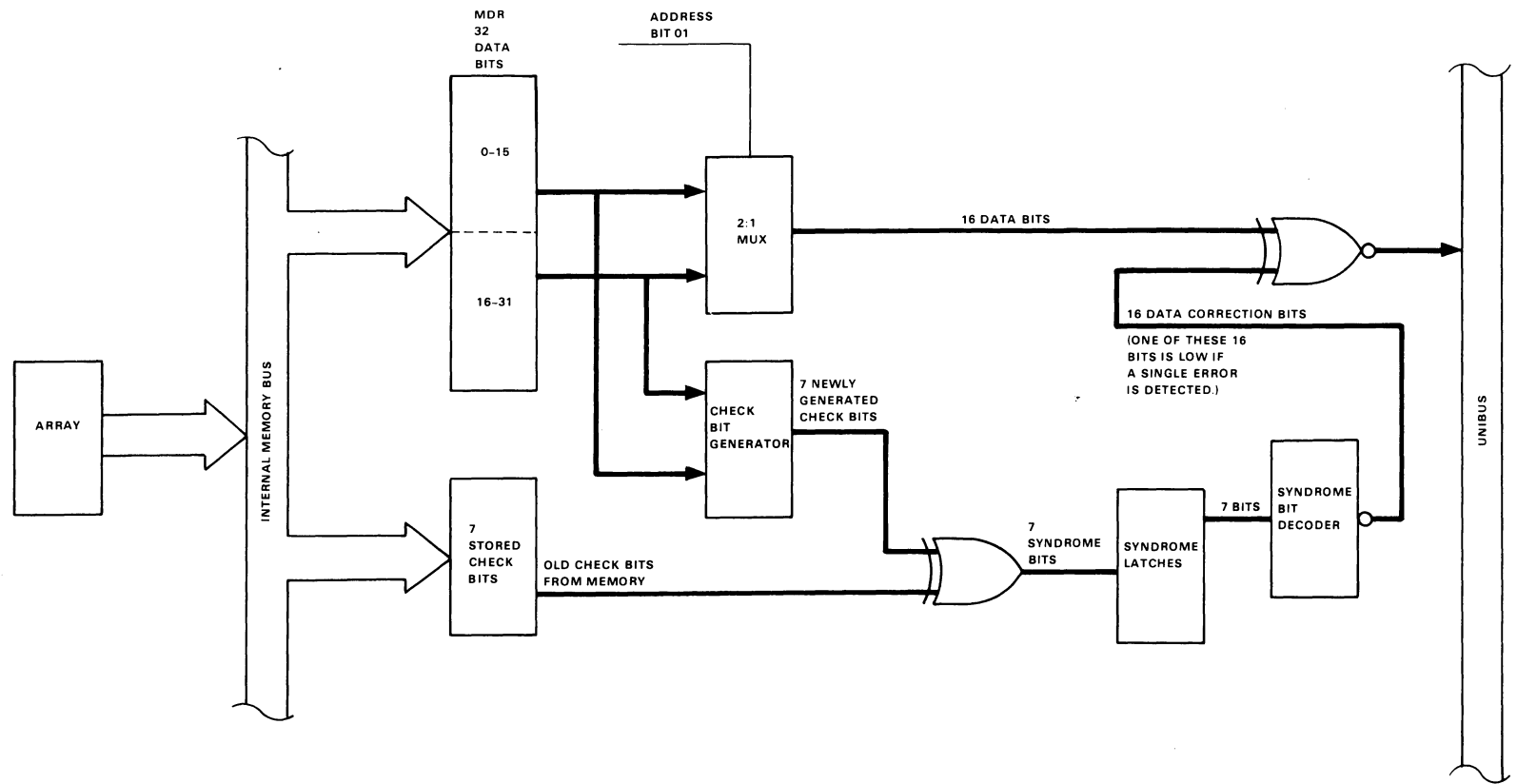
Figure 2-2 Read Cycle Flowchart

Table 2-1 Syndrome Bit Generation

Stored Check Bit	Generated Check Bit	Syndrome Bit
0	0	0
0	1	1
1	0	1
1	1	0

Table 2-2 Error Correction, Example A

	Data Bit 0	Check Bits						
		CT	C32	C16	C8	C4	C2	C1
Stored	1	1	1	0	0	1	1	1
Retrieved	0	0	1	1	1	1	1	1
	0	ST	S32	S16	S8	S4	S2	S1
Syndrome Bits	–	1	0	1	1	0	0	0



11-5419

Figure 2-3 ECC Logic Used on a Read Cycle Block Diagram

S ₃₂	S ₁₆	S ₈	
0	0	0	
0	0	1	
0	1	0	
0	1	1	BYTE 0
1	0	0	
1	0	1	BYTE 1
1	1	0	BYTE 2
1	1	1	BYTE 3

S ₄	S ₂	S ₁	
0	0	0	BIT 0
0	0	1	BIT 1
0	1	0	BIT 2
0	1	1	BIT 3
1	0	0	BIT 4
1	0	1	BIT 5
1	1	0	BIT 6
1	1	1	BIT 7

11-5417

Figure 2-4 Syndrome Bit Decoding

2.5.4 The Error Correction Process

After the syndrome bits are latched, the control logic examines them to see if any are 1s. If no syndrome bits are 1s, indicating that no error has been detected, the requested data is sent to the Unibus.

However, if, as in Example A, a logic 0 in data bit 0 is retrieved, syndrome bits ST, S16, and S8 will be set to logic 1s. The error correction logic checks to find whether the quantity of syndrome bits set to 1 is odd or even. If an odd number of syndrome bits is set, the error is treated as a single error condition.

In Example A, above, ST, S16 and S8 are logic 1s, indicating a single error condition. Latched syndrome bits S32, S16, and S8, are decoded to determine whether or not the error is in the data word requested by the bus master. If the data requested is in the lower word of the 32-bit data unit, which, in this case, contains the error, the decoder circuit further decodes latched syndrome bits S4, S2, and S1 to identify the bit in error. This bit (bit 0 in Example A, above) is then inverted before the requested data is sent to the Unibus. If the error is not in the requested data, it remains uncorrected, and the data that is requested by the bus master is sent to the Unibus. It is important to note that data is never corrected except when it is to be gated to the Unibus on a read.

2.5.5 Example B, Handling a Double Error

The MF11S-K memory does not attempt to correct double errors. It will detect a double error and, on a read, notify the processor that a fatal error has been encountered, regardless of the location of the errors within the 39-bit memory word. It is important to notice that the use of the check bit, Ct, will ensure that an even number of syndrome bits is set on a double error.

Suppose, for example, that data bits 1 and 3 are the only logic 0s in a word stored in the location accessed. When the word is retrieved, bits 1 and 3 are both logic 1s. The stored check bits are all 1s except C2, which is 0. Of the newly generated check bits, Ct will be 0, the rest will all be 1s. (Table 2-3).

The syndrome bit pattern generated, 1000010, shows an even number of bits to be 1s. The error correction logic decodes this pattern to indicate a double error. The requested data word is then sent, uncorrected, to the Unibus. The memory notifies the processor that a fatal error has occurred, causing the initiation of a trap routine, if enabled by software. The data that is sent may, or may not, contain errors depending on the location of the two errors.

Table 2-3 Double Error, Example B

	Data Bit 1	Check Bits							
		Bit 3	CT	C32	C16	C8	C4	C2	C1
Stored	0	0	1	1	1	1	1	0	1
Retrieved	1	1	0	1	1	1	1	1	1
			ST	S32	S16	S8	S4	S2	S1
Syndrome Bits			1	0	0	0	0	1	0

2.5.6 Handling a Check Bit Error, Example C

The MF11S-K memory will detect a check bit error, as well as a data bit error, on a read operation. However, it must make a distinction between the two types of errors. Data bit errors require correction; check bits errors do not.

For example, if zeros are stored in a 32-bit data location, the stored check bits C32, C16, C8, C4, C2, and C1 will all be 1s. Ct will be a 0. Suppose that when the location is accessed on a read, the data bits are retrieved without error, but that check bit C2 is a logic 0 instead of a logic 1. Syndrome bit S2 will be set to 1 and latched, as shown in Table 2-4.

Table 2-4 Check Bit Error, Example C

	Data	CT	C32	C16	C8	C4	C2	C1
Stored	0	0	1	1	1	1	1	1
Retrieved	0	0	1	1	1	1	0	1
		ST	S32	S16	S8	S4	S2	S1
Syndrome Bits		0	0	0	0	0	1	0

A look at Figure 2-4 will show that the syndrome bit pattern generated, 0000010, will cause no byte, and, therefore, no bit to be selected for error correction. The requested data will be gated to the Unibus unaltered.

2.6 WRITE OPERATION IN GENERAL (DATO AND DATOB)

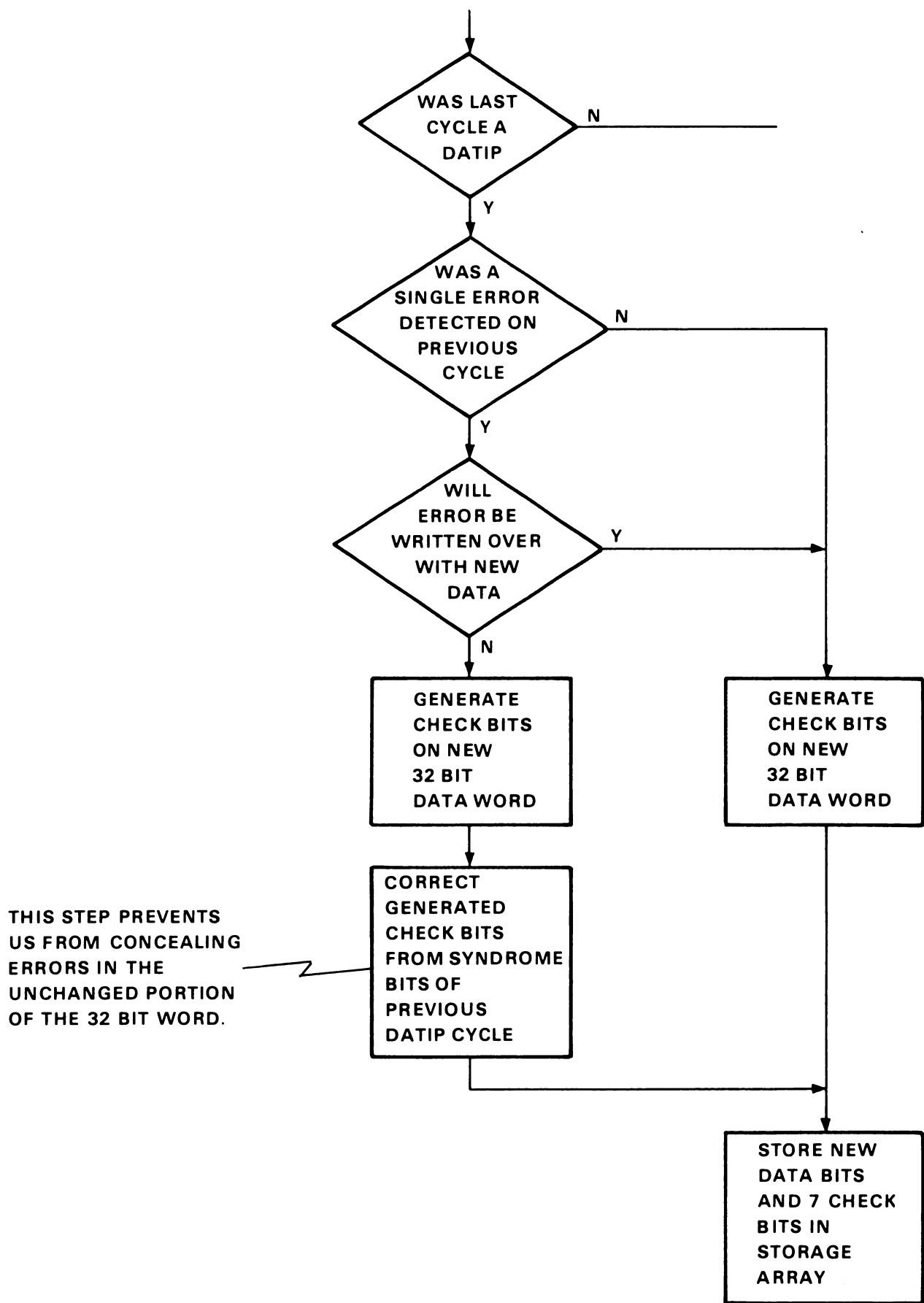
On a write operation, data is sent from the bus master in one or two byte units. However, since MF11S-K memory words are organized as four data bytes plus seven check bits, the write operation never changes more than half the data in a memory location.

The MF11S-K memory begins a normal write operation with an internal read cycle. This is necessary because new check bits must be generated on a full double word at the memory locations addressed, even though only one or two bytes are to be altered. The entire write operation thus involves an internal read-modify-write routine within the MF11S-K memory, although only a write cycle has been called for on the Unibus control lines.

However, when the memory performs a write cycle following a DATIP cycle, it accesses the same location on both cycles. This means that a read-modify-write routine is not necessary, since the data in the bytes of the double word that are not to be written over is already stored in the MDR. This data does not need to be accessed a second time to enable the generation of new check bits.

2.7 WRITE OPERATION FOLLOWING A DATIP

At the beginning of a write operation, the MF11S-K memory must determine whether or not the previous cycle was a DATIP, as shown in Figure 2-5.



11-5418

Figure 2-5 Write Cycle Flowchart

In fact, if the previous cycle was a DATIP, the error correction logic checks the syndrome bits latched during the DATIP cycle to see if a single error was detected on that cycle. Note that if a double error has been detected on the previous DATIP cycle, the memory will perform a read-modify-write cycle instead of a straight write cycle.

The memory control logic moves the data to be written from the Unibus to the appropriate byte location(s) in the MDR. The 32 data bits in the MDR, at this point, contain new data (one or two bytes) and old data from the unused bytes in the memory word (two or three bytes).

2.7.1 No Single Error Detected

If no single error from the previous cycle has been detected, the 32 data bits in the MDR should all be correct. New check bits are then generated on the 32-bit double word; and the new data (one or two bytes only), together with the seven newly generated check bits, is then stored in the array.

2.7.2 Single Error Detected

If one or more syndrome bits remain latched as logic 1 from the previous DATIP cycle, a single error has been detected. If the error is in a data bit, three, five, or seven syndrome bits will be set. If one syndrome bit is set, it indicates the presence of a check bit error.

The error correction logic then decodes the syndrome bit pattern to determine which byte in the 32-bit data word contains the bit in error. If the error is in a location to be written over, or if the bit in error is one of the old check bits, it is ignored. Check bits are generated on the new double word; the new data, together with the seven new check bits, is stored in the array.

However, a problem arises if the error is located in a byte that is to remain unaltered. Simple generation of new check bits would mean that the check bits would describe bad data as if it were correct. Check bits must be generated on the 32 data bits containing the bad data, as before. But these check bits must then be corrected so that they still point to the bit in error. Figure 2-6 shows how the Exclusive ORing of the latched syndrome bits (from the previous cycle) with the newly generated check bits accomplishes the correction process.

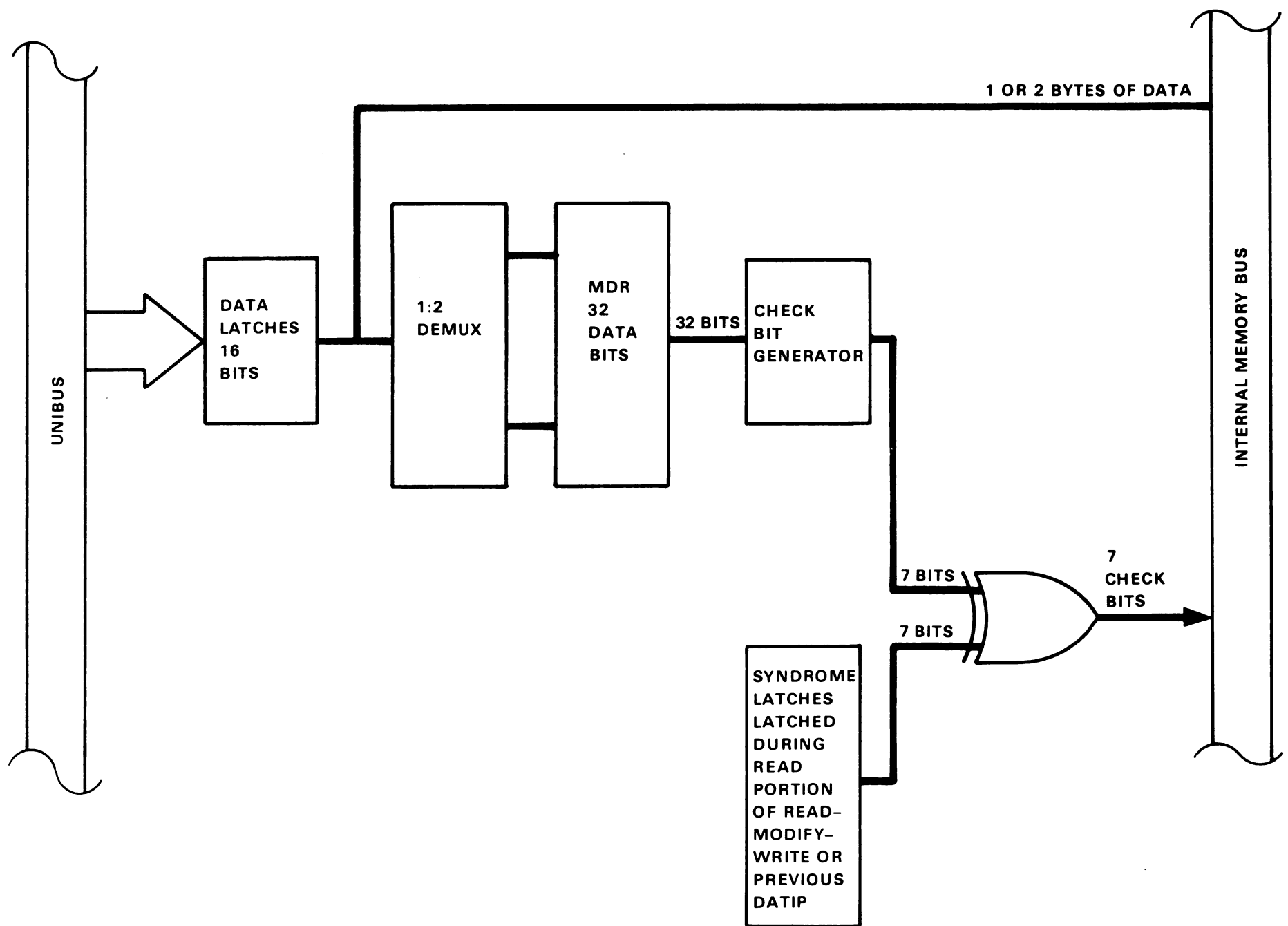
The new data and corrected check bits are then stored in the array. It is important to note that this step prevents the memory from concealing errors in the unchanged portion of the 32-bit word. Notice, also, that data containing errors is not corrected on a write cycle. Check bits that describe the memory location are merely updated so that the error information they contain remains valid.

2.8 WRITE OPERATION NOT FOLLOWING A DATIP

When the MF11S-K memory executes a read-modify-write routine in response to a write command that does not follow a DATIP, it begins by reading 32 data bits and 7 check bits from the 39-bit memory location containing the bytes to be written over (see the flowchart in Figure 2-7). Then, as on a read cycle, the error correction logic generates check bits on the data from the array and compares them with the stored check bits to form syndrome bits. These syndrome bits are latched, because it is important to keep track of any errors that are indicated.

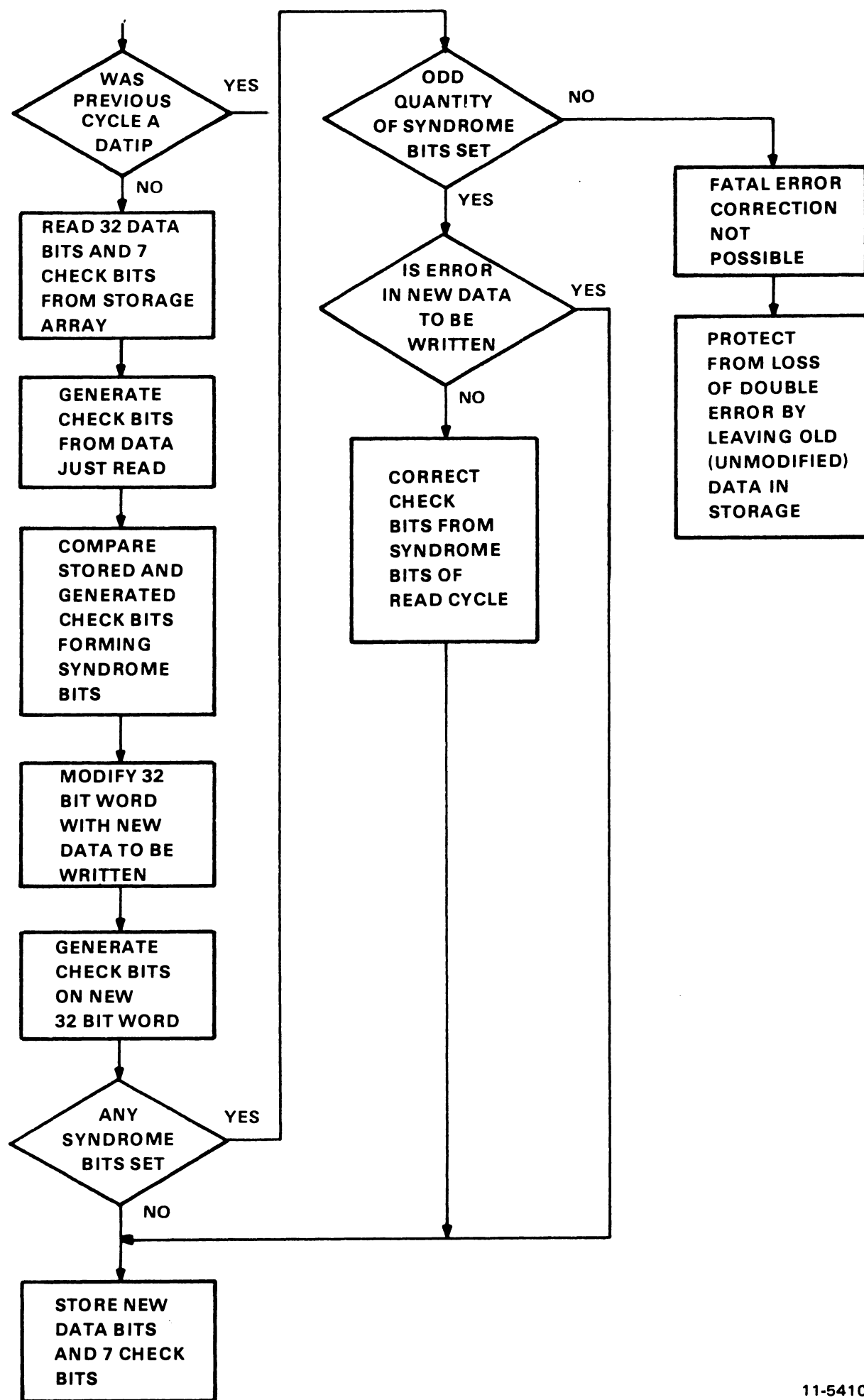
Next, the contents of the MDR are modified as the byte or bytes to be written are latched in the appropriate byte locations. At this point, check bits are generated again. This second set of newly generated check bits corresponds to the modified memory word that is to be stored in the array.

The error correction logic then checks latched syndrome bits, which reflect the old double word. If no syndrome bits are latched as 1s, the data in the MDR should be correct. The byte, or bytes, to be written, together with the second set of generated check bits, is stored in the array.



11-5415

Figure 2-6 ECC Logic Used on a Write Cycle



11-5410

Figure 2-7 Read-Modify-Write Cycle Flowchart

2.8.1 Error Detection on a Write

However, if any latched syndrome bits are 1s, an error in the old word is indicated. The error correction logic then checks to find whether the number of latched syndrome bits set to 1 is odd or even.

2.8.2 Double Error Condition

If the number of latched syndrome bits set to 1 is even, it indicates that a double error existed in the old word read from memory. To preserve the double error condition, all modification of data and check bits normal to a write operation is inhibited. The old data word and old check bits are returned unaltered to the array.

Detection of a double error on a write cycle will not cause the initiation of a trap routine or notify the processor of a fatal error. The processor is warned of double errors only on DATI and DATIP cycles.

It is important to save double error conditions for troubleshooting purposes, however. If one or both of the errors were soft, and measures were not taken to preserve the errors, the double error condition might be impossible to locate subsequently.

2.8.3 Single Error Condition

If an odd number of latched syndrome bits are found to be logic 1s on a write cycle following a DATIP, they indicate that a single error existed in the old double word. The latched syndrome bit pattern is then decoded so that the error correction logic can determine whether or not the single error is located in a byte to be written over. If the error is in a byte to be written, it is ignored, and the data to be written, together with the second set of newly generated check bits, is stored in the array.

However, if the error is found to be in a byte that is not to be written over, the check bits must be altered. The second set of newly generated check bits describes the data in the MDR as if it were correct. Exclusive ORing of the syndrome bits with the second set of newly generated check bits produces corrected check bits, as shown in Figure 2-6. At this point, the data to be written and the corrected check bits are stored in the array. Note that the bit in error remains uncorrected. Only the check bits that describe the double word have been altered, ensuring that the check bits can be used to form syndrome bits, and thus locate the error, on the next read cycle to that location.

CHAPTER 3

INSTALLATION OF THE MF11S-K MEMORY

3.1 GENERAL

Since the nine slot backplane, DD11-K, has been designed specifically for the MF11S-K, the backplane specifications should be examined thoroughly before configuration and power application.

3.2 DD11-K BACKPLANE SLOT ALLOCATION

The M7984 array module will fit in slots 7 through 4 of the DD11-K backplane. The M7983 controller, however, must be placed in slot 8. The array slots are dedicated according to memory size. The first 32K of memory must go in slot 7, the second in slot 6, the third in slot 5, and the fourth in slot 4. Slot 1 must be used for the Unibus cable-in jumper. Slot 9 must contain either a Unibus cable-out jumper or a terminator. Figure 3-1 shows the DD11-K backplane slot allocation.

The internal memory bus runs on the A and B connectors to slots 8 through 4 (Figure 3-2).

While all of the slots are available as small peripheral controller slots, if the MF11S-K memory is not present, care must be used in slot allocation with or without the presence of MF11S-K memory modules. Hex SPC modules can be installed but power and ground pins must be compatible (see appendix). Slots 2 and 3 are pinned according to the modified Unibus configuration on connectors A and B. Some single module core and MOS memories could be plugged into these slots if no MF11S-K memory is installed in the backplane.

MF11S-K memories and other memory types are mutually exclusive with respect to the backplane. The MF11S-K requires ± 12 V where the MS11-E through J series of MOS memories requires ± 15 V and core memories require +20 V.

After the backplane has been configured with the MF11S-K modules and Unibus jumpers (or jumper and terminator), the slots that remain can be used for any SPC control module that picks up Unibus signals on connectors C, D, E, and F. Slots that are unused altogether should contain bus grant cards.

NOTE

**Be sure to remove the NPG wire on the backplane
when installing an NPG option in any SPC slot.**

The backplane power harness will plug into any power distribution board slot.

3.3 MF11S-K OPTIONS

The options available in the MF11S-K memory are listed below.

3.3.1 MF11S-K (32K MOS System Complete)

This option contains the following:

- One M7983 controller
- One M7984 32K × 16 array module
- One DD11-K backplane with appropriate power harness
- One M9202 Unibus jumper
- Seven bus grant cards
- Print set
- User's manual
- Diagnostic DZMML

NOTE

An additional letter at the end of the option name specifies the appropriate power harness depending on the box in which the backplane is installed; i.e., the MF11S-KK comes with the DD11-KK power harness for the BA11-K 10-1/2 in (26.67 cm) box. The MF11S-KF comes with the DD11-KF power harness for the BA11-F 21 in (53.34 cm) box.

3.3.2 MS11-KA (32K × 16 MOS Array Module)

This option contains the following:

One M7984-AA 32K × 16 MOS array module.

NOTE

M7984-AA = M7984-AB = M7984-AC = M7984-AD = M7984-AE, etc.

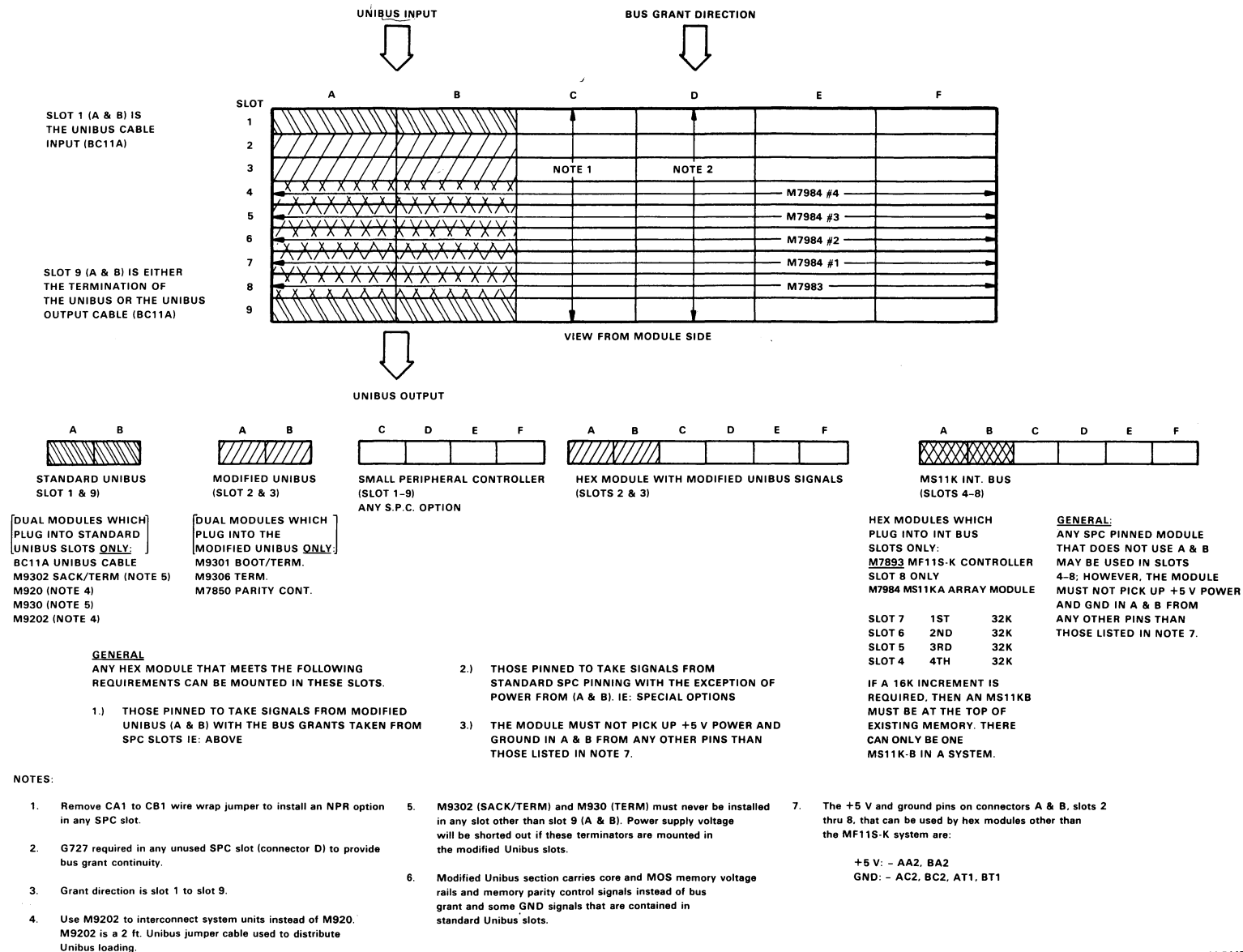
The MS11-KA can be replaced by an MS11-KE. The latter option contains faster access time MOS chips. Using this option does not, however, reduce system access time.

The letter *A* designates the memory module size (32K × 16 bits). The second letter identifies the vendor of the MOS chips used on the array.

3.3.3 Options Required Per Storage Capacity

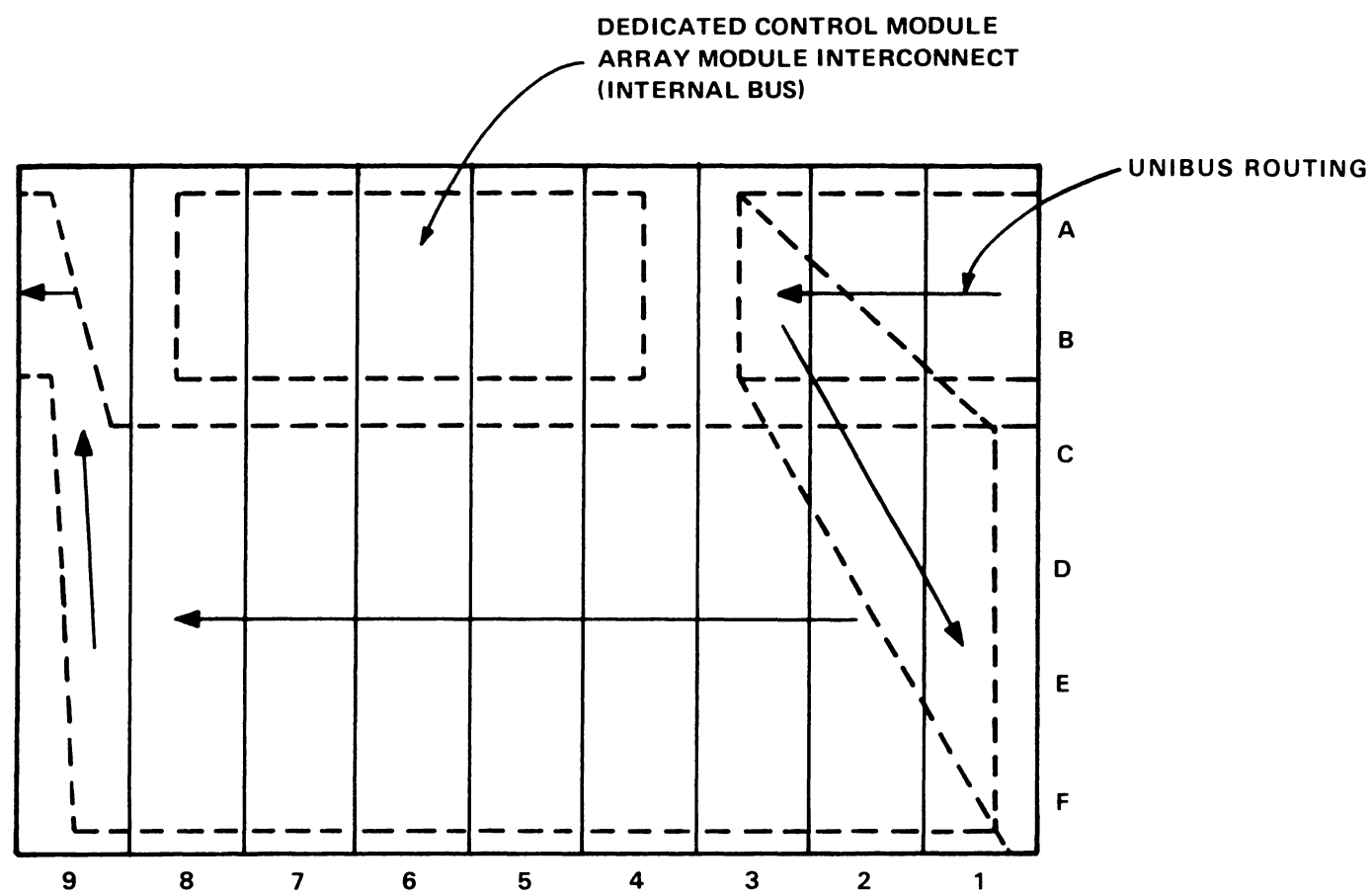
Storage Capacity Options Required	
32K × 16	One MF11S-K
64K × 16	One MF11S-K One MS11-KA or MS11-KE
96K × 16	One MF11S-K Two MS11-KA or MS11-KE
128K × 16	One MF11S-K Three MS11-KA or MS11-KE

Figure 3-3 shows a fully configured MF11S-K 128K memory.



11-5447

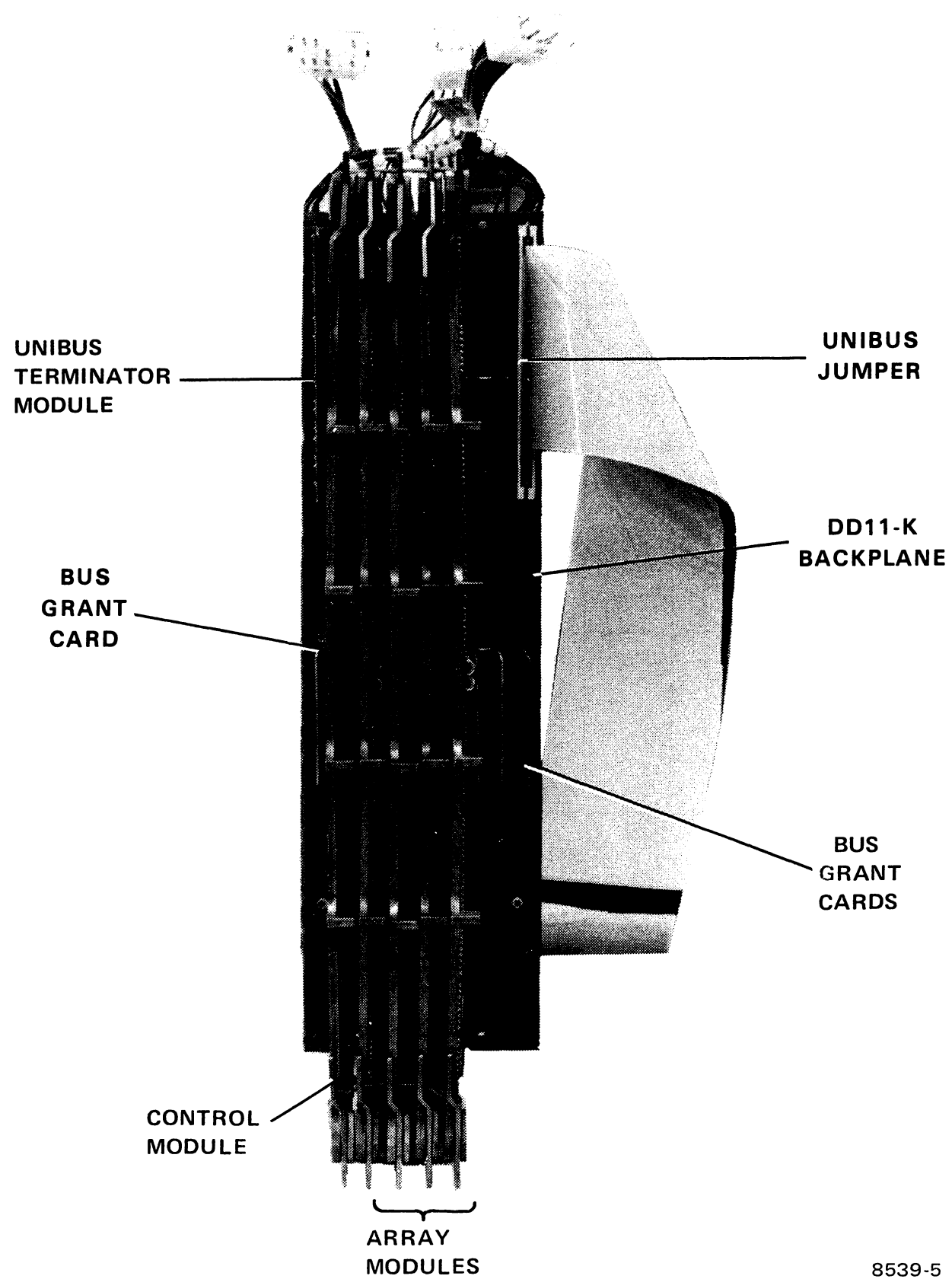
Figure 3-1 DD11-L Backplane Utilization



NOTE:
Routing from connectors AB → CDEF is staggered
to reduce wire bundling. Critical signals are on twisted pair.

11-5444

Figure 3-2 DD11-K Backplane Unibus Routing



8539-5

Figure 3-3 Fully Configured DD11-K Backplane

3.4 MIXING THE MF11S-K AND OTHER MEMORIES

Other memories can be used in the address space both above and below the MF11S-K. These other memories, whether core or MOS, must be installed in another system box because of the power supply restrictions mentioned in Paragraph 3.2. The MF11S-K requires ± 12 V.

The address of the lowest location in the MF11S-K is controlled by four microswitches located on the M7983 controller (Table 3-1). The highest address is a function of the number of array modules used.

Table 3-1 gives the switch settings used for various starting addresses for the MF11S-K memory. Starting address 0 should be used if the MF11S-K is the only memory in the system or if it is to occupy the lowest memory address space in the system.

Table 3-1 Starting Address Selection

Starting Address Decimal	Starting Address Octal	Switch Setting			
		(A17) SW1-1	(A16) SW1-2	(A15) SW1-3	(A14) S1-4
0	0	F	F	F	F
8K	040000	F	F	F	N
16K	100000	F	F	N	F
24K	140000	F	F	N	N
32K	200000	F	N	F	F
40K	240000	F	N	F	N
48K	300000	F	N	N	F
56K	340000	F	N	N	N
64K	400000	N	F	F	F
72K	440000	N	F	F	N
80K	500000	N	F	N	F
88K	540000	N	F	N	N
96K	600000	N	N	F	F

NOTE: N = ON
F = OFF

The user should notice that the maximum memory size in an MF11S-K system is 128K words. Thus, if the address switches are set improperly, and the MF11S-K memory size is too large to fit in the space allotted, an upper portion of the MF11S-K memory will wrap around to the bottom of memory. This condition may cause the MF11S-K memory to overlay some other memory within the system.

For example, if the switches are set so that S1-1 is on and S1-2, S1-3, and S1-4 are off, the starting address selected will be 64K. If three 32K array modules are installed in the system, the upper 32K address space will be wrapped around to overlay whatever memory is in the first 32K locations.

3.4.1 Address Space Above MF11S-K

There are no restrictions on putting other memories in the address space above the MF11S-K.

3.4.2 Address Space Below MF11S-K

Other Memory Below MF11S-K	Max MF11S-K Size
8K→ 32K	96K
40K→ 64K	64K
72K→ 96K	32K

For a full 128K system, other types of memories may, in some cases, be placed in the address space above the MF11S-K.

3.5 SUPPLY VOLTAGES

Check the +5 V and ± 12 V levels on the backplane before installing modules. When system power can be applied, the voltage level on the +12 V line on the array module should be adjusted at the power supply to provide the correct level.

3.6 FIRST TIME START-UP

The following procedure should be followed for first time start up of the MF11S-K memory system upon installation.

1. Ensure that all power is removed from the memory, including battery power.
2. Apply power.
3. Load MAINDEC-11-DZMML-A, the MF11S-K system diagnostic program.
4. Run DZMML for two complete passes (with the console switch register set at zeros).
5. Run DZMML for two complete passes with single errors enabled (bit 11 on the console switch register set). There must be no single errors for installation. However, after installation, single errors are allowable and are not considered faults.

CHAPTER 4

CONDITION STATUS REGISTER FUNCTION AND LOGIC

4.1 GENERAL

The CSR controls the memory diagnostic modes and manipulates and stores error information as well. Thereby, it gives the user access to the memory that he must have for troubleshooting and programming purposes.

The CSR address will normally be 772136. A pair of jumpers allows the option of choosing address 772134 (Table 4-1).

Table 4-1 CSR Address Modification

Jumper Installed	CSR Address Selected
W1	772136
W2	772134

This allows the user to have two controllers on the same system. They would, of course, have to be in two separate backplanes. Note that these addresses are the two highest addresses of the parity module (M7850). The CSR can be addressed even if the memory is busy, i.e., finishing a data cycle or performing a refresh cycle. Contents are not changed on a refresh cycle or with battery backup master reset. Figure 4-1 shows the bit allocation of the CSR.

On power up, all bits except CSR bits 11–5 are cleared to zeros. All used bits can be written and read from the Unibus, with the unused bits (12 and 14) being read as zeros.

Note that pressing the start button will enable BUS INIT, clearing the CSR (except bits 11–5). Pressing continue will not clear the CSR.

4.2 READING FROM AND WRITING TO THE CSR

CSR bits can be read by addressing the CSR and performing a DATI operation. They can be written by addressing the CSR, placing a 1 or 0 on the appropriate UNIBUS data line, and performing a DATO.

4.3 CSR BIT FUNCTIONS

A discussion of the individual CSR bits follows.

15	DOUBLE ERROR
14	NOT USED
13	SET INHIBIT MODE
12	NOT USED
11	A17 OR CHECK BIT T
10	A16 OR CHECK BIT 32
9	A15 OR CHECK BIT 16
8	A14 OR CHECK BIT 8
7	A13 OR CHECK BIT 4
6	A12 OR CHECK BIT 2
5	A11 OR CHECK BIT 1
4	SINGLE ERROR
3	INHIBIT MODE 0 PROTECTS 1ST 16K POINTER 1 PROTECTS 2ND 16K
2	DIAGNOSTIC CHECK
1	DISABLE ERROR CORRECTION
0	DOUBLE ERROR INDICATION ENABLE

NOTE:
IF DIAGNOSTIC CHECK
BIT IS SET THEN
CHECK BITS ARE ALWAYS
STORED IN CSR ON
DATI CYCLE TO MEMORY.
IF NOT SET THAN ADDRESS IS
STORED ON DOUBLE ERROR
ONLY ON DATI OR
DATIP CYCLE.

11-5395

Figure 4-1 CSR Bit Allocation

4.3.1 Double Error Indication Enable (CSR Bit 0)

If a double error occurs on a DATI or DATIP cycle to the memory and CSR bit 0 is set, then BUS PBL will be asserted on the Unibus at the same time as the data. Bit 0 is reset by BUS INIT L.

4.3.2 Disable Error Correction Mode (CSR Bit 1)

If CSR bit 1 is set, the syndrome bit latches are never enabled on any type of cycle to memory. Thus, the error correction circuits are disabled and no single-bit error correction or double-error detection will occur. Correct check bits are still written into memory, however. This is a diagnostic aid to allow writing and reading data from memory without interference from the error correction logic.

Also, this is a means of clearing a soft double error in memory. Note that a soft double error may be caused by the occurrence of one hard error and one soft error, or two soft errors within a memory word (a hard double error cannot be cleared). The disable error correction bit is set and the appropriate word in memory is written. No double error is sensed on the read portion of the read-modify-write cycle because the syndrome latches are disabled. On the write portion, the new check bits are generated on whatever data is in the MDR and are not corrected when written into memory. The next read to that address with ECC enabled will not show any errors.

No double-error or single-error information is logged in the CSR having this mode enabled. This bit is reset by BUS INIT L.

4.3.3 Diagnostic Ckeck Mode (CSR Bit 2)

CSR bit 2 is reset by BUS INIT L. The diagnostic check mode allows a means of writing data stored in CSR bits 11–5 into the MOS RAM chips used for check bit storage. A DATO cycle to memory is used for this. The cycle is always the shorter write cycle in this mode instead of the read-modify-write cycle. Since the check bits to be written are already in the CSR, it is not necessary to generate them with a read-modify-write cycle.

This mode also allows a means of reading data stored in the check bit storage into CSR bits 11–5. A DATI cycle to memory is used for this.

Operation in the diagnostic check mode, therefore, allows both an integrity check on the check bit storage chips and a check on the error correction logic. With a data field of all 1s, the check bits can be set so that they change bit 0 from a 1 to a 0. This can be repeated on each bit. The data can then be complemented and the test repeated.

For example, the check bit pattern generated on a data pattern of all 0s will be 0111111 (where CT is 0). If the user alters the check bit pattern to read 1100111, the error correction circuitry should detect an error in bit 0 of the double word (Table 4-2). In order to implement this example, the user should perform a DATO to the CSR, writing the 1100111 (starting with CT) pattern into bits 11–5 of the CSR (note that CT corresponds to CSR bit 11), and writing a 1 into CSR bit 2 (the diagnostic check bit). With the CSR thus set up, the user should perform several DATO cycles to memory, writing 0s into several consecutive locations.

Table 4-2 Data Modification Through the Check Bits

Check Bit Pattern	CT	C32	C16	C8	C4	C2	C1
Generated on all 0s data	0	1	1	1	1	1	1
Imposed check bit pattern	1	1	0	0	1	1	1

The diagnostic check bit (CSR bit 2) should then be set to 0, with a DATO to the CSR. When the user subsequently initiates DATI cycles to the same memory locations, the error correction logic should invert bit 0 of the data, placing the corrected data on the Unibus. Notice that bit 0 will be inverted only on every other word, because no correction will take place on bit 16 of the double word.

If a double error occurs in this mode (with CSR bit 1 = 0) the D ERR bit in the CSR is set, but the check bits read from memory are stored in CSR bits 11–5 and not the D ERR address bits.

4.3.4 Inhibit Mode Pointer (CSR Bit 3)

CSR bit 3 is reset by BUS INIT L. The inhibit mode pointer works in conjunction with the set inhibit mode (bit 13). When bit 13 is set to a 1, a 16K portion of memory is inhibited from operating in the ECC disable mode or diagnostic check mode. The inhibit mode pointer indicates which 16K portion is being inhibited; i.e., if bit 3 = 0, the first 16K of memory is inhibited. If bit 3 = 1, the second 16K of memory is inhibited. With bit 13 set to a 0, bit 3 becomes inoperative.

Bit 3, in conjunction with bit 13, therefore, allows a 16K portion of memory to always have ECC coverage. The system diagnostic can reside in this protected portion of memory and can disable ECC and/or run the diagnostic check mode in the rest of memory without itself becoming vulnerable to single errors.

4.3.5 Single Error (CSR Bit 4)

On a DATI or DATIP cycle to memory, if a single error occurs, bit 4 is set. An examination of the CSR can thus be performed to see if a single error has occurred. This bit is cleared by BUS INIT L.

4.3.6 Address and Check Bit Storage (CSR Bits 11–5)

CSR bits 11–5 are not cleared by BUS INIT L or on +5 VBB power up.

Address Storage (DIAG CK Bit 2 = 0, ECC Disable Bit 1 = 0)

If a double error occurs on a DATI or DATIP cycle, then address bits A17–A11 are stored in these bits. This locates the double error within a 1K segment of memory.

Check Bit Storage (Diag CK Bit 2 = 1)

During a diagnostic check mode, CSR bits 11–5 are used to store the check bits to be written into memory or the check bits read from memory. If a double error occurs in the diagnostic mode, and the ECC Disable bit (CSR bit 1) equals 0, then the check bits are stored in the CSR together with the double-error bit.

4.3.7 Set Inhibit Mode (CSR Bit 13)

CSR bit 13 is cleared by BUS INIT L. When set to a logic 0, it allows the diagnostic check mode and/or ECC disable mode to operate over the entire installed memory.

When bit 13 is set to a logic 1, it enables the INH MODE POINTER to inhibit either the first or second 16K portion of memory from ever going into the diagnostic check or ECC disable mode.

4.3.8 Double Error (Bit 15)

On a DATI or DATIP (ECC disable bit 1 equals 0), CSR bit 15 is set if a double error occurs. The upper address bits A17 – A11 are stored in CSR bits 11–5 (DIAG CK bit 2 = 0) and will point to the 1K segment of memory that was addressed when the double error occurred. If DIAG CK bit 2 = 1, then the check bits read from memory are stored in CSR bits 11–5. Bit 15 also energizes an LED for a visual indication of the error. This bit is cleared by BUS INIT L. Notice, however, that although bit 15 and the LED will be cleared by BUS INIT L, the double error to which it refers will not be cleared.

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

What features are most useful? _____

What faults or errors have you found in the manual? _____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

☐ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name _____	Street _____
Title _____	City _____
Company _____	State/Country _____
Department _____	Zip _____

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, Ma 01532
Attention: Communications Services (NR2/M15)
Customer Services Section

Order No. _____

Fold Here

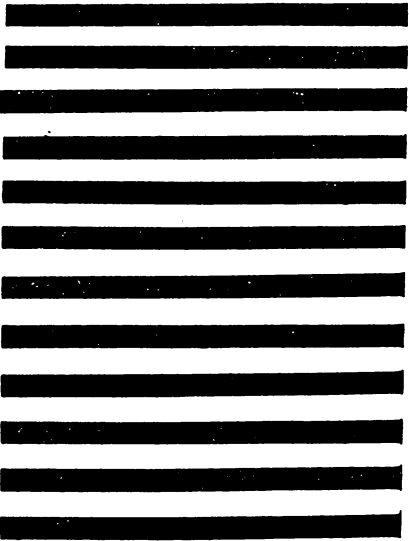
Do Not Tear - Fold Here and Staple

**FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.**

**BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES**

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
Maynard, Massachusetts 01754



digital

digital equipment corporation