# MS7AA-FA Memory Module Service Guide

Order Number: EK–MS7AA–SV .A01

These instructions describe the procedure for identifying and replacing a failing SIMM on the VAX 7000/10000 or DEC 7000/10000 MS7AA-FA 2-gigabyte memory module.

Copyright © Digital Equipment Corporation. 1994. All rights reserved.

**Digital Equipment Corporation**
**Maynard, Massachusetts**

The MS7AA-FA memory module is the 2-gigabyte memory module for VAX 7000/10000 and DEC 7000/10000 systems. It is populated with 36 64-Mbyte single in-line memory modules (SIMMs). Should a SIMM fail it can be replaced in the field.

These instructions tell how to identify the failing SIMM and how to replace a SIMM.

- Section 1 tells how to identify the failing SIMM from the operating system error log.
- Section 2 tells how to replace the SIMM.
- Section 3 tells how to identify the SIMM from the console level. This information may be needed if the operating system cannot be booted.

**NOTE**

**The part number for the 64-Mbyte SIMM is 54-21718-01. This SIMM can only be used on a 2-Gbyte module.**

## 1 How to Identify a Failing SIMM from an Operating System Error Log

First you must identify the failing SIMM.

1. From the error log, locate the error syndrome (for OpenVMS, see Example 1).
2. Determine if the string is odd or even; is it string 1, 3, 5, or 7, or string 0, 2, 4, or 6?
3. Determine if the memory interface controller (MIC) error is a MIC A or a MIC B error.
4. Find the SIMM number in the matrix of Table 1.

For example, from the OpenVMS AXP error log in Example 1, you see:

- The MS7AA-FA module has an error syndrome 34 (see ❶).
- The failing string is 3, which is odd (see ❷).
- The MIC is B (see ❸).

Therefore, from Table 1 you find 34 in the first column, labeled Syndrome. The string, 3, is odd so you look at the columns labeled Odd. The number under MIC B is J31, the socket that holds the failing SIMM.

**NOTE**

**The OSF/1 operating system error log will appear in the next version of this document.**

## Example 1: Sample OpenVMS System Error Report

```
V M S               SYSTEM ERROR REPORT        COMPILED 24-JAN-1994 08:28:00
                                                                   PAGE  23.

******************************* ENTRY      84. *******************************
ERROR SEQUENCE 630.                           LOGGED ON:  CPU_TYPE 00000002
DATE/TIME 21-JAN-1994 10:27:26.26                          SYS_TYPE 00000003
SYSTEM UPTIME: 0 DAYS 16:29:20
SCS NODE: SUVB02                                            VMS V1.5

HW_MODEL: 00000402 Hardware Model = 1026.

MEMORY ERROR KN7AA DEC 7000 MODEL 620
      CRD FLAGS           0000
      LOG REASON          0004
RELATED ENTRY 1 OF 1
      BAD PAGES       00000000
      MEMDSC SIZE     00000020
      MEMDSC OFFSET   00000060
      NUM OF FPRINTS  00000001
      FPRINT SIZE     00000050
      FPRINT OFFSET   00000080

 MEMORY DESCRIPTOR #1
      NODE            00000006
      LDEV            00004000
      MCR             0000000C
      AMR             00000343

 MEMORY DESCRIPTOR #2
      NODE            00000007
      LDEV            00004000
      MCR             0000000C
      AMR             0000034B

 1 FOOTPRINTS IN THIS PACKET

 CRD FOOTPRINT #1

      FOOTPRINT       0004000D
                      00000006
                                      Syndrome = 34(X)      ❶
                                      Bit in Error = 6.
                                      Failing string = 3.   ❷
                                      MICB error            ❸
                                      Failing node = 6.
      SYSTEM TIME                     20-JAN-1994 20:18:02.93
      LOW ADDRESS     00000000 0153AE00
      HIGH ADDRESS    00000000 115E2600
      CUM ADDRESS     00000000 100DF800
      SCRUB BLOCKSIZE 00000040
      STATIC FLAGS        0001
      LOG REASON          0008
      CALLER FLAGS    00000000
      SCRUB FAIL      00000000
      MATCH COUNT     0000000E
      SCRUB COUNT     0000000E
      LAST SCRUB TIME                 21-JAN-1994 08:50:02.93
```

## Table 1: 2-Gigabyte SIMM Isolation Matrix

| | String: Even | | Odd | | | String: Even | | Odd | |
|---|---|---|---|---|---|---|---|---|---|
| | MIC | | MIC | | | MIC | | MIC | |
| Syndrome | A | B | A | B | Syndrome | A | B | A | B |
| 00 | na | na | na | na | 51 | J10 | J34 | J11 | J35 |
| 01 | J22 | J36 | J23 | J37 | 52 | J10 | J34 | J11 | J35 |
| 02 | J22 | J36 | J23 | J37 | 54 | J8 | J28 | J9 | J29 |
| 04 | J20 | J36 | J21 | J37 | 58 | J4 | J28 | J5 | J29 |
| 07 | J2 | J24 | J3 | J25 | 61 | J4 | J18 | J5 | J19 |
| 08 | J18 | J36 | J19 | J37 | 62 | J4 | J20 | J5 | J21 |
| 0B | J22 | J30 | J23 | J31 | 64 | J4 | J20 | J5 | J21 |
| 0D | J2 | J24 | J3 | J25 | 68 | J10 | J14 | J11 | J15 |
| 0E | J14 | J26 | J15 | J27 | 70 | J4 | J22 | J5 | J23 |
| 10 | J18 | J32 | J19 | J33 | 80 | J14 | J26 | J15 | J27 |
| 13 | J8 | J16 | J9 | J17 | 83 | J10 | J22 | J11 | J23 |
| 15 | J8 | J20 | J9 | J21 | 85 | J12 | J18 | J13 | J19 |
| 16 | J14 | J32 | J15 | J33 | 86 | J20 | J36 | J21 | J37 |
| 19 | J10 | J36 | J11 | J37 | 89 | J6 | J26 | J7 | J27 |
| 1A | J10 | J26 | J11 | J27 | 8A | J12 | J26 | J13 | J27 |
| 1C | J8 | J16 | J9 | J17 | 8C | J6 | J22 | J7 | J23 |
| 1F | J24 | J32 | J25 | J33 | 8F | J18 | J34 | J19 | J35 |
| 20 | J14 | J34 | J15 | J35 | 91 | J6 | J28 | J7 | J29 |
| 23 | J12 | J14 | J13 | J15 | 92 | J10 | J28 | J11 | J29 |
| 25 | J6 | J18 | J7 | J19 | 94 | J2 | J28 | J3 | J29 |
| 26 | J20 | J32 | J21 | J33 | 98 | J8 | J30 | J9 | J31 |
| 29 | J4 | J26 | J5 | J27 | A1 | J2 | J32 | J3 | J33 |
| 2A | J10 | J26 | J11 | J27 | A2 | J12 | J30 | J13 | J31 |
| 2C | J2 | J24 | J3 | J25 | A4 | J12 | J36 | J13 | J37 |
| 2F | J6 | J20 | J7 | J21 | A8 | J4 | J30 | J5 | J31 |
| 31 | J8 | J30 | J9 | J31 | B0 | J18 | J34 | J19 | J35 |
| 32 | J6 | J30 | J7 | J31 | C1 | J20 | J32 | J21 | J33 |
| 34 | J12 | J30 | J13 | J31 | C2 | J18 | J34 | J19 | J35 |
| 38 | J2 | J28 | J3 | J29 | C4 | J16 | J28 | J17 | J29 |
| 40 | J14 | J30 | J15 | J31 | C8 | J16 | J28 | J17 | J29 |
| 43 | J6 | J24 | J7 | J25 | D0 | J8 | J22 | J9 | J23 |
| 45 | J2 | J24 | J3 | J25 | E0 | J24 | J34 | J25 | J35 |
| 46 | J16 | J36 | J17 | J37 | F1 | J2 | J16 | J3 | J17 |
| 49 | J8 | J26 | J9 | J27 | F2 | J16 | J32 | J17 | J33 |
| 4A | J4 | J32 | J5 | J33 | F4 | J22 | J34 | J23 | J35 |
| 4C | J6 | J24 | J7 | J25 | F8 | J12 | J14 | J13 | J15 |
| 4F | J12 | J16 | J13 | J17 | | | | | |

## 2 How to Replace a SIMM

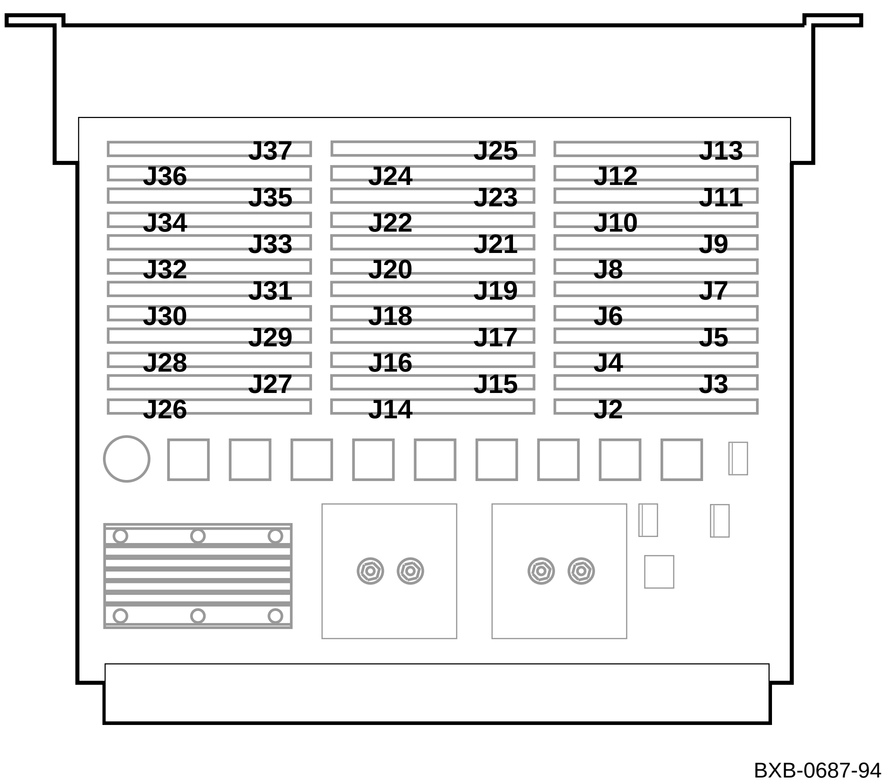
After you have determined the failing SIMM on the memory module, remove the module from the system and follow this procedure.

**CAUTION**

**You must wear an antistatic wrist strap attached to the cabinet when you handle any modules.**

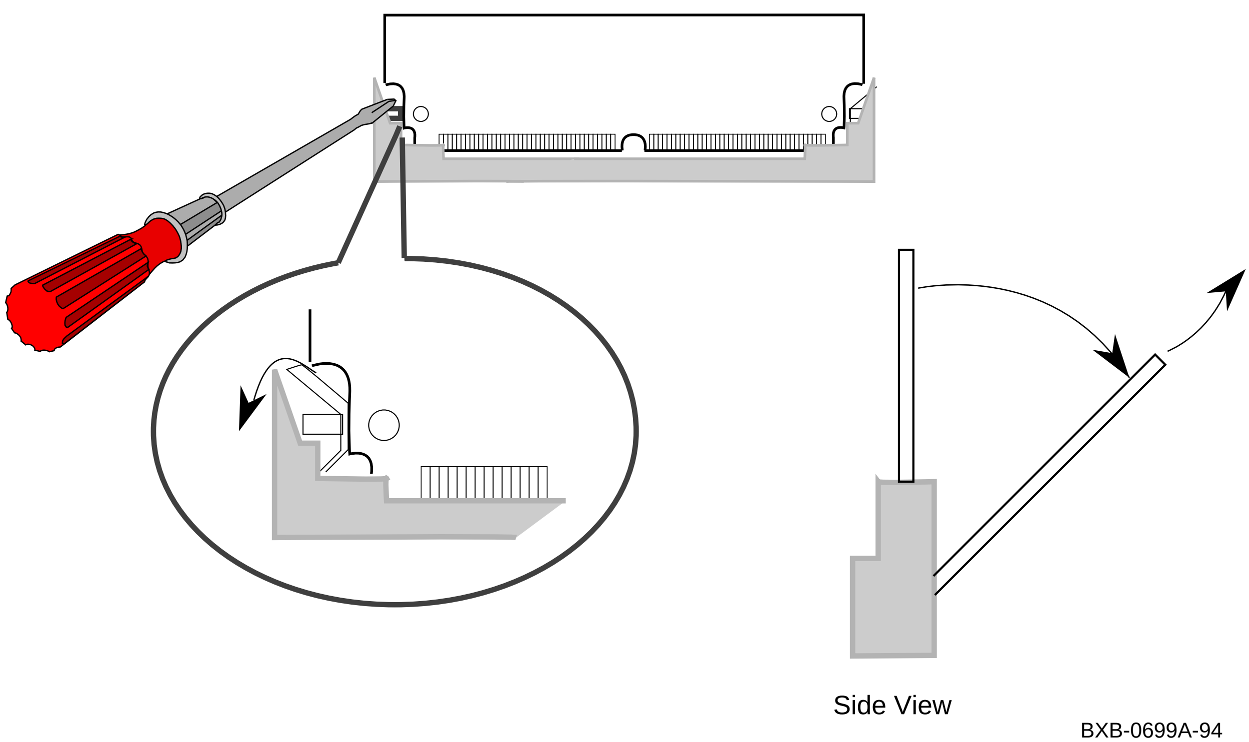
1. Remove the cover that shields side 1 of the module by removing the eight small Phillips screws.
2. Determine the location of the failing SIMM from Figure 1.
3. Locate the row of SIMMs on the module that holds the failing SIMM.

**Figure 1: SIMM J Connector Numbers**



4. Beginning with the SIMM closest to the gate arrays, remove each SIMM up to and including the failing SIMM. To remove a SIMM, release the latches on both ends of the SIMM connector. Insert a #1 Phillips screwdriver as shown in Figure 2, and rotate the screwdriver until the latch releases. Open both latches. Then turn the SIMM at a 45 degree angle toward the gate arrays and pull the card out of the connector.
5. Put the failing SIMM aside for return to the appropriate repair facility.
6. Insert a new SIMM in place of the failing SIMM, angling it into the connector at 45 degrees. Turn it to a vertical position until the latches snap into place. The connector is keyed in the center so that the correct side of the SIMM faces front.
7. Insert the other SIMMs back into their connectors.
8. Replace the module cover.

**Figure 2: Removing a SIMM**



## 3 How to Identify a Failing SIMM at Console Level on a DEC 7000/10000

While in console mode, you can determine which SIMM has failed. Example 2 shows a sample console session with the steps to take to identify a failing SIMM.

### Example 2: Sample Console Display

```
>>> set mode diag          ❶
>>> set d_startup on
d_startup set to on

>>> show mem               ❷

Set   Node   Size   Base Addr   Intlv   Position
---   ----   ----   ---------   -----   --------
 A      1   2048Mb   000000000   2-Way      0

      ❸ # 2048MB = 2GB = 8000 0000 (hex)

>>> mem_ex -t 1 -sa 1000000 -ea  7ffffc0

                              ❹ # 8000 0000 - 40 = 7FFF FFC0

      ID  Program          Device     Pass Hard/Soft Test     Time
-------- -------- --------------- -------- --------- ---- --------
      49   mem_ex             mem        0    0    0      20:59:54

CPU 0
unexpected exception/interrupt through vector 00000066
process mem_ex, pcb = 007F0620

 pc: 00000000 000D6B40  ps: 30000000 00000004
 r2: 00000000 0013F8A0  r5: 00000000 00001F04
 r3: 00000000 001ECCA0  r6: 00000000 1FBFFFF0
 r4: 00000000 00000020  r7: FFFFFFFF FFFFFFFF

               [listing of GPRs and FPRs]

Machine Check Logout - base: 00006000

 flags:      00000000 00000000          byte_count:    80000000 000001D8
 offsets:    000001A0 00000110          das_debug:     00E00555 00000020
 pt0:        00000001 00000100          pt1:           00000000 000000FC

                    [listing of registers]

 lbesr2:     00000000 0000007F          lbesr3:        00000000 0000007F
 lbecr0:     00000000 03000500 ❺        lbecr1:        00000000 000C8040

                                        # 03000500 x 20 (hex) = 6 000A000

 lmmr0:      00000000 00000000          lmmr1:         00000000 00000321

                     [more registers]

 ms7aa0_lber:00000000 00040203          ms7aa0_lbecr0: 00000000 03000500
 ms7aa0_lbecr1:00000000 000C8040        ms7aa0_mera:   00000000 00000C07
 ms7aa0_msynda:00000000 000000F3        ms7aa0_merb:   00000000 00000007
 ms7aa0_msyndb:00000000 000000F3

Failing FRU: ms7aa0         ❻
>>> CPU:0 Halt Code =  1
operator initiated halt
PC = 13ee0c

>>> dep -l ms7aa0:21c0 10000002     ❼

>>> mem_ex -t 1 -f -sa 60000000 -l 2000000     ❽
```

**Example 2 (continued on next page)**

**Example 2 (Cont.): Sample Console Display**

```
      ID  Program          Device     Pass Hard/Soft Test     Time
-------- -------- -------------- -------- --------- ---- --------
      4f   mem_ex            mem        0    0    0      21:01:29

>>> dep -l ms7aa0:21c0 10000000    ❾
>>> dep  -l ms7aa0:2140  ff        ❿
>>> dep  -l ms7aa0:2440  ff
>>> mem_ex -t 1 -f -sa 60000000 -l 2000000    ⓫

      ID  Program          Device     Pass Hard/Soft Test     Time
-------- -------- -------------- -------- --------- ---- --------
      51   mem_ex            mem        0    0    0      21:01:31

>>> ex -l ms7aa0:2140          ⓬      # Address of MERA register
ms7aa0: 00002140 00000015
>>> ex -l ms7aa0:2180          ⓭      # Address of MYSNDA register
ms7aa0: 00002180 00000045      ⓮
>>> ex -l ms7aa0:4180                # Address of MYSNDB register
ms7aa0: 00004180 000000F3
>>> ex -l ms7aa0:2100          ⓯      # Address of FADR register
ms7aa0: 00002100 03000500      ⓰
>>>
```

❶ Enter diagnostic mode.

❷ Determine the size of physical memory using the **show memory** command.

❸ Subtract 40 from the highest memory address to determine the ending address for mem_ex.

❹ Run mem_ex test 1 from 16 meg (100 0000) to the top of memory.

❺ Multiply the contents of the LBERC0 register by 20 (hex) to get the failing address.

❻ Determine the failing memory module, ms7aa0.

❼ Disable ECC checking on the failing module.

❽ Initialize all of memory on the failing module by running mem_ex test 1 with the **-f** option on the 32 meg address block that contains the failing address. This will clear the double-bit errors that were generated during memory self-test.

Starting address = 30 0500 X 20 = 6000 A000 = failing byte address
                                        ^ from callout ⓰
Test address = 6000 0000
Length = 20 0000

❾ Enable ECC checking on the memory module by depositing 1000 0000 into Memory Diagnostic Register A.

❿ Clear the error registers on the memory module.

⓫ Run mem_ex test 1 with the **-f** option on the 32 meg address block that contains the failing address.

⓬ Examine Memory Error Register A on the failing memory module to determine the failing syndrome (see Figure 3).

⓭ Examine Memory Error Syndrome Registers A and B to determine the failing bank.

⓮ The contents of Memory Error Syndrome Register A gives the error syndrome.

⓯ Examine the Failing Address Register (FADR) on the failing module. Use Table 2 to determine if the failing string is Odd or Even.

⓰ The contents of FADR indicates the string.

From this information you can identify the failing SIMM.

For example, from the console display in Example 2, you see:

- The MS7AA-FA module has an error syndrome 45 (see ⓮).
- The string is even (see ⓯). From the **show mem** command (see ❷) we know the interleave is 2-way. Using the contents of FADR ⓰ and Table 2 we know the string is even.
- The MIC is A (see ⓬) because CERA is set in MERA (see ⓬ and Figure 3).

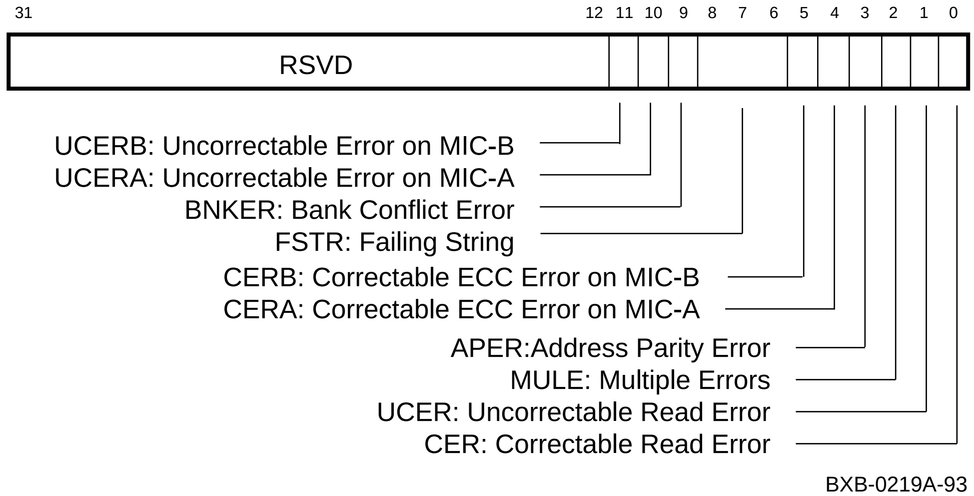
Therefore, from Table 1 you find 45 in the first column, labeled Syndrome. The failing string is even so you look at the columns labeled Even. The number under MIC A is J2, the socket that holds the failing SIMM.

**Table 2: Using FADR Bit to Determine Odd/Even String**

| No. of 2-Gbyte Modules | Interleave Count | FADR Bit |
|---|---|---|
| 1* | 2 | bit 1 (0 = Even string 1 = Odd string) |
| 2 | 4 | bit 2 (0 = Even string 1 = Odd string) |
| 4 | 8 | bit 3 (0 = Even string 1 = Odd string) |

*The interleave count for one 2-Gbyte module with four 512-Mbyte modules is 4. Use FADR bit 2 in this case.

**Figure 3: Memory Error Register A**




**NOTE**

**For more information about the memory registers, see the *MS7AA Memory Technical Manual*.**