11/24

11/24 ROM M9312
CJM9ABO

AH-T208B-MC
FICHE 1 OF 1

d i g i t a l

```
                        .REM %
```

IDENTIFICATION

PRODUCT CODE: AC-T207B-MC

PRODUCT NAME: CJM9AB0 11/24 ROM M9312

PRODUCT DATE: SEPTEMBER,1982

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: J. HAMEL

```
49
50
51                                          HISTORY
52                                          -------
53
54
55                          CJM9AA  - DOCUMENT RELEASED APRIL 1982
56                          CJM9AB  - CHANGES MADE SO DIAGNOSTIC WOULD INTERFACE CLEANLY WITH
57                                    CONSOLE EMULATOR,AND FIXED TROUBLE OF LOSING LAST TWO
58                                    CHARACTERS OF MEMORY SIZE PRINOUT.
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION
1.1 ABSTRACT
1.2 SYSTEM REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS
2.1 LOADING AND STARTING PROCEDURE
2.2 PROGRAM OPTIONS
2.3 EXECUTION TIMES

3.0 ERROR INFORMATION
3.1 ERROR REPORTING PROCEDURES
3.2 ERROR HALTS

4.0 PERFORMANCE AND PROGRESS REPORTS
4.1 PERFORMANCE REPORTS
4.2 PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES

6.0 PROGRAM DESCRIPTION
6.1 PROGRAM EXECUTION CHARACTERISTICS
6.2 SUBTEST SUMMARIES
6.3 SPECIAL SUBROUTINE DESCRIPTION

7.0 HISTORY

8.0 LISTING

1.0 GENERAL PROGRAM INFORMATION

1.1 ABSTRACT

THIS DIAGNOSTIC IS GO/NOGO VERIFICATION OF AN 11/24 SYSTEM WHICH TESTS:

1. ALL SINGLE AND DOUBLE OPERAND INSTRUCTIONS, INCLUDING EIS, UTILIZING ALL SOURCE AND DESTINATION ADDRESSING MODES.

2. ALL OF MEMORY, IN 4K PAGES, VIA MEMORY MANAGEMENT AND PRINT THE MEMORY SIZE (LAST MEMORY ADDRESS +2).

3. SLU1 VIA MAINTENANCE MODE (ALSO CHECKS CONSOLE PRINTER AND INTERFACE WHEN PRINTING MEMORY SIZE).

4. THE DIAGNOSTIC IS COMPATIBLE WITH M9312 DIAGNOSTIC ROM FORMAT REQUIREMENTS.

1.2 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

THIS DIAGNOSTIC IS DESIGNED TO RUN ON AN 11/24 WITH CONSOLE TERMINAL AND 4K OF MEMORY (MINIMUM). FURTHER, IT ASSUMES THE PRESENCE OF THE MEMORY MANAGEMENT UNIT (MMU) CHIP.

B. SOFTWARE REQUIREMENTS

NONE

1.3 RELATED DOCUMENTS AND STANDARDS

THE FOLLOWING DOCUMENTS WERE USED OR REFERENCED DURING THE CREATION OF THIS DIAGNOSTIC:

1. DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS PROGRAMMING PRACTICES (DOC. NO. 175-003-009-02).

2. PDP-11 SYSMAC PACKAGE (MAINDEC-11-DZQAC-C3).

3. REQUIRMENTS FOR NEW BOOT ROMS AND CPU ROMS USED IN THE M9312 (K-SP-M9312-0-8).

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

NONE

1.5 ASSUMPTIONS

THE DIAGNOSTIC ASSUMES PROPER STEPUP OF THE DIP SWITCHPAK ON THE M9312 BOOTSTRAP MODULE (SEE M9312 USER'S MANUAL AND TABLE BELOW FOR THIS INFORMATION) AND PRESENCE OF A BOOT ROM IN THE M9312.

		S1 (1-10)		
BOOTSTRAP:	DIAGNOSTICS	FIRST DEVICE (ALL ROMS)	VIRTUAL ADDRESS	SWITCHPACK S1 SWITCHES ON
ODT	NO YES	2004 2006	165004 165006	1,9 1,9,10
DEVICE ROM #1	NO YES	0004 0006	173004 173006	9 9,10
DEVICE ROM #2	NO YES	0204 0206	173204 173206	4,9 4,9,10
DEVICE ROM #3	NO YES	0404 0406	173404 173406	3,9 3,9,10
DEVICE ROM #4	NO YES	0604 0606	173604 173606	3,4,9 3,4,9,10

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURE

IF DIAGNOSTIC IS SELECTED BY M9312 SWITCHPACK, IT WILL BE RUN ON POWER UP, BUT CAN ALSO SELECTED FROM MICRO-ODT BY COMMAND:

```
165000G
```

TO BOOT A DEVICE ROM DIRECTLY, USE THE VIRTUAL ADDRESS COLUMN OF THE ABOVE TABLE. FOR EXAMPLE, TO BOOT DEVICE ROM #3 WITH DIAGNOSTICS, USE 173406G (FROM MICRO-ODT).

2.2 PROGRAM OPTIONS

NONE

2.3 EXECUTION TIMES

A. FIRST PASS (QV)
A PASS OF THIS CODE TAKES APPROXIMATELY 6 SECONDS TO COMPLETE INCLUDING THE PRINTING OF THE MEMORY SIZE. THIS TIMING IS BASED ON A 11/24 CPU WITH 128K WORD OF MOS MEMORY.

B. LONGEST TEST
THE LONGEST SINGLE TEST IS THE MEMORY TEST WHICH TAKES APPROXIMATELY 5 SECONDS PER 128K WORDS OF MEMORY.

C. ADDITIONAL TIME FOR UNITS
APPROXIMATELY 5 SECONDS IS ADDED TO THE TEST TIME FOR EVERY ADDITIONAL 128K WORDS OF MEMORY. THIS TIMING IS BASED ON AN 11/24 WITH MOS MEMORY.

SAMPLE TEST TIMES BASED ON ABOVE FIGURES:

128KW=	6 SECONDS
256KW=	11 SECONDS
512KW=	21 SECONDS
1024KW=	41 SECONDS
1536KW=	61 SECONDS
1920KW=	76 SECONDS

D. FULL PASS TIME (ITERATIONS)
THIS PROGRAM DOES NOT DO ANY ITERATIONS ON ANY OF THE TESTS AND IN FACT ONLY MAKES ONE PASS THRU THE CODE FOR EACH START.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

SINCE THIS DIAGNOSTIC IS A GO/NOGO, LOW-LEVEL TEST, NO ERROR REPORTING, AS SUCH, IS IMPLEMENTED; HOWEVER, IF THE MICRO-ODT AND CONSOLE TERMINAL ARE OPERATIONAL THE ERROR HALT ADDRESS+2 WILL BE TYPED FOR THE OPERATOR.

3.2 ERROR HALTS

BADADD	=	165144	THIS ERROR IS CAUSED BY TRAPPING TO LOCATION 4 AT ANY TIME PRIOR TO EXECUTING THE MEMORY TEST ON THE FIRST 4K OF MEMORY. THE PROGRAM DOES ACCESSES TO SOME OF THE MEMORY MANAGEMENT REGISTERS DURING THIS TIME. IT MAY BE HELPFUL TO EXAMINE THE STACK BUT S INCE THE PROGRAM HAS NOT SET IT UP THE INFORMATION RECIEVED MAY NOT BE VALID.
CPUERR	=	165146	THIS ERROR INDICATES A FAILURE WITH EITHER THE BASE INSTRUCTION SET OR THE EIS INSTRUCTION SET. FIRST SUSPECT THE DCF11-A HYBRID OR THE CPU BOARD.
MEMERR	=	165550	THIS ERROR INDICATES A MEMORY SYSTEM FAILURE. FIRST SUSPECT THE MEMORY THEN THE KTF11-A. TO LOCATE THE FAILING BANK DIVIDE THE CONTENTS OF PARO (1772342) BY 200(8) THEN MULTIPLY BY 4.
SLUERR	=	165702	THIS HALT INDICATES A DATA ERROR IN THE CONSOLE SLU. THE GOOD DATA IS IN R2.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

NONE

4.2 PROGRESS REPORTS

THE PROGRAM REPORTS ITS PROGRESS USING TWO LEDS LOCATED ON THE 11/24 CPU BOARD. THEY ARE LOCATED ON THE HANDLE EDGE OF THE BOARD ALONG WITH THE RUN LIGHT. THEY ARE DRIVEN THROUGH BITS 0 AND 1 OF THE DISPLAY REGISTER WITH BIT 0 DRIVING THE LED FURTHEST FROM THE FRONT OF THE BOX AND BIT 1 DRIVING THE LED IN THE CENTER OF THE THREE LIGHTS. THE LIGHT CLOSEST TO THE FRONT OF THE BOX IS THE RUN LIGHT. THE PROGRAM FIRST WRITES AN OCTAL 3 TO THE REGISTER(TURNING BOTH LEDS ON) TO SIGNAL THE START OF THE CPU TEST. THE COUNT IS DECREMENTED BY ONE AT THE START OF EACH OF THE NEXT THREE TEST SECTIONS.

LIGHT COUNT	LAST TEST COMPLETED
3	NONE-SUCCESSFUL ENTRY TO PROGRAM
2	CPU TEST
1	MEMORY TEST
0	SLU TEST

5.0 DEVICE INFORMATION TABLES

POOL (LOC 165102) CONTAINS DATA USED BY DOUBLE OPERAND SECTION.

6.0 PROGRAM DESCRIPTION

6.1 PROGRAM EXECUTION CHARACTERISTICS

PROPER EXECUTION OF THE DIAGNOSTIC RESULTS IN PRINTOUT OF THE MEMORY SIZE AND BOOTING A PERIPHERAL OR ENTERRING MICRO-ODT (DEPENDING ON M9312 SWITCH SETTINGS).

6.2 SUBTEST SUMMARIES

A. CPUTST - THIS SUBTEST VERIFIES THE PDP-11 INSTRUCTION SET, INCLUDING EIS, FOR BOTH WORD AND BYTE FORMATS. IT CONSISTS OF FIVE SECTIONS - SINGLE OPERAND (DESTINATION MODE 0), DOUBLE OPERAND (ALL SOURCE MODES, DESTINATION MODE 0), CONDITIONAL BRANCHES, BYTE INSTRUCTIONS (ALL DESTINATION MODES), AND JSR/RTS WITH EIS.

B. MEMTST - THIS SUBTEST CHECKS ALL OF MEMORY IN 4K PAGES USING THE MMU, DETERMINES MEMORY SIZE, AND CONSTRUCTS A PHYSICAL ADDRESS FROM THE VIRTUAL ADDRESS WHICH CAUSED A TIMEOUT TRAP. LOOPING THROUGH EACH 4K PAGE IS CONTROLLED BY THE 1$ LOOP, AND; WITHIN THIS LOOP, THE 2$ LOOP LOADS EACH LOCATION WITH ITS ADDRESS, THE 3$ LOOP CHECKS THE DATA AND COMPLEMENTS IT, THE 4$ LOOP ADDS THE CONTENTS OF EACH LOCATION TO ITS COMPLEMENT AND INCREMENTS THE RESULT TO PRODUCE ZERO. SECTION TIMOUT TURNS OFF MEMORY MANAGEMENT AND BUILDS A PHYSICAL ADDRESS IN R0 AND R1.

NOTE: BECAUSE OF SPACE LIMITATIONS THERE
IS A KNOWN FLAW IN THE MEMORY
SIZING ROUTINE. IF MAXIMUM MEMORY
IS CONFIGURED ON THE SYSTEM(1920KW)
THE SIZING ROUTINE WILL REPORT A
MEMORY SYSTEM SIZE OF 2044KW. THIS
IS BECAUSE OF THE UNIBUS MAP WILL
MAP THE LAST 124K OF VIRTUAL
ADDRESS SPACE TO THE LOWER 124K
OF MAIN MEMORY.

C. SLUTST - THIS SUBTEST PLACES SLU1 IN MAINTENANCE MODE (SERIAL OUT OF UART TIED TO SERIAL IN OF UART) AND TESTS THAT ALL 8-BIT PATTERNS CAN BE TRANSMITTED AND RECEIVED. SECTION PRINT PRINTS THE MEMORY SIZE CALCULATED BY MEMTST, AND TRANSFERS CONTROL BACK TO ODT OR THE APPLICABLE BOOT ROM.

6.3 SPECIAL SUBROUTINE DESCRIPTION

THE PRINT ROUTINE TYPES THE LAST 22-BIT MEMORY ADDRESS+2 FOUND BY THE MEMORY SIZE ROUTINE. THE 16 HIGH ORDER BITS ARE SAVED IN R0 AND THE 6 LOW ORDER BITS ARE SAVED IN R1.

7.0 HISTORY

CJM9AA - DOCUMENT RELEASED APRIL 1982
CJM9AB - CHANGES MADE SO DIAGNOSTIC WOULD INTERFACE CLEANLY WITH CONSOLE EMULATOR,AND FIXED TROUBLE OF LOSING LAST TWO CHARACTERS OF MEMORY SIZE PRINOUT.

8.0 LISTING

```
                        %

                        .TITLE  CJM9AB
                        .SBTTL  ROM AREA 165000-165776
                        .SBTTL  GO/NOGO MINIMUM DIAGNOSTIC
                        .TITLE  CJM9AB0 11/24 ROM M9312
                        ;*COPYRIGHT (C) SEPTEMBER,1982
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY D.SOBIK
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
                        ;*
000001                  $TN=1
160000                  $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT

                        .SBTTL  REGISTER DEFINITIONS
177570                          SWR=    177570          ;SWITCH REGISTER (DIAGNOSTIC LIGHTS)
177776                          PS=     177776          ;PROCESSOR STATUS WORD
                        ;SLU1 REGISTERS
177560                          RCSR=   177560          ;RECEIVER CSR
177562                          RBUF=   177562          ;RECEIVER BUFFER
177564                          XCSR=   177564          ;TRANSMITTER CSP
177566                          XBUF=   177566          ;TRANSMITTER BUFFER

                        ;MMU REGISTERS
172340                          PAR0=   172340          ;KERNAL PAGE ADDRESS REGISTERS
172342                          PAR1=   172342
172356                          PAR7=   172356
172300                          PDR0=   172300          ;KERNAL PAGE DESCRIPTOR REGISTERS
172302                          PDR1=   172302
172316                          PDR7=   172316
177640                          UPAR0=  177640          ;USER PAGE ADDRESS REGISTERS
177642                          UPAR1=  177642
177572                          SR0=    177572          ;STATUS REGISTER 0
172516                          SR3=    172516          ;STATUS REGISTER 3 (22-BIT)
```

```
                    165000                          .=165000
165000  000177  006020          START:  JMP     @173024         ;TRANSFER TO SELECTED BOOT ROM OR ODT
165004  000000                  HALT                            ;ENTRY POINT  FOR ODT, NO DIAGNOSTICS
                                                                ;ENTRY POINT FOR ODT WITH DIAGNOSTICS
165006  012704  165002          MOV     #165002,R4      ;SET UP RETURN ADDRESS-2 IN R4

                                .SBTTL  CPU TEST
                                ;;*****************************************************************

                                ;       BASIC CPU TEST

                                ;;*****************************************************************
165012  012737  000003  177570  CPUTST: MOV     #3, @#SWR       ;LIGHTS = 3, INDICATING CPUTST
165020  012737  165144  000004          MOV     #BADADD,@#4     ;SET UP TIME OUT VECTOR INCASE OF TRAP
165026  005037  000006                  CLR     @#6             ;CLEAR PRIORITY OF TRAP ROUTINE
165032  010037  177640                  MOV     R0,     @#UPAR0 ;WE ARE USING THE UPAR'S HERE SIMPLY BECAUSE
165036  010137  177642                  MOV     R1,     @#UPAR1 ;THEY ARE AVAILABLE UNUSED INTERNAL REGS.
                                                                ;WE ARE STORING PARAMETERS FROM THE BOOT ROM

                                                                ;R1 CONTENTS    N Z V C
165042  005001                  CLR     R1              ;000000         0 1 0 0
165044  005201                  INC     R1              ;     1         0 0 0 0
165046  005101                  COM     R1              ;177776         1 0 0 1
165050  006201                  ASR     R1              ;177777         1 0 1 0
165052  006301                  ASL     R1              ;177776         1 0 0 1
165054  006001                  ROR     R1              ;177777         1 0 1 0
165056  005701                  TST     R1              ;177777         1 0 0 0
165060  005401                  NEG     R1              ;     1         0 0 0 1
165062  005301                  DEC     R1              ;     0         0 1 0 1
165064  005601                  SBC     R1              ;177777         1 0 0 1
165066  005501                  ADC     R1              ;     0         0 1 0 1
165070  001026                  BNE     CPUERR          ;ERROR IF NOT ZERO

                                ; SECTION FOR DOUBLE OPERAND, ALL SOURCE MODES, DEST MODE 0

165072  012702  165126                  MOV     #POOL,  R2      ;SET UP ADDRESS OF DATA TABLE
165076  011201                  MOV     (R2),   R1      ;R1/POOL, SMODE 1
165100  022201                  CMP     (R2)+,  R1      ;DATA CORRECT? SMODE 2
165102  001021                  BNE     CPUERR
165104  063201                  ADD     @(R2)+, R1      ;R1/POOL + 1, SMODE 3
165106  165201                  SUB     @-(R2), R1      ;R1/POOL , SMODE 5
165110  044201                  BIC     -(R2),  R1      ;R1/0 ,SMODE  4
165112  056201  000004          BIS     4(R2),  R1      ;R1/177777, SMODE 6
165116  037201  000006          BIT     @6(R2), R1      ;RESULT IS 177777, SMODE 7
165122  001411                  BEQ     CPUERR
165124  000411                  BR      CONT            ;BRANCH AROUND DATA TABLE

165126  165126          POOL:   .WORD   POOL
165130  165136                  .WORD   DATA1
165132  177777          DATA2:  .WORD   177777
165134  165132                  .WORD   DATA2
165136  000001          DATA1:  .WORD   1
165140  000500          DOUBLE: .WORD   500
165142  000500                  .WORD   500
165144  000000          BADADD: HALT
165146  000000          CPUERR: HALT
```

```
                                ;       CHECK CONDITIONAL BRANCHES

 165150  000277                 CONT:   SCC                     ;SET ALL CONDITION CODES
 165152  001375                         BNE     CPUERR          ;BR IF Z=0
 165154  100374                         BPL     CPUERR          ;"  "  N=0
 165156  102373                         BVC     CPUERR          ;"  "  V=0
 165160  103372                         BCC     CPUERR          ;"  "  C=0
 165162  002771                         BLT     CPUERR          ;"  "  N XOR V=1
 165164  003370                         BGT     CPUERR          ;"  "  Z OR (N XOR V)=0
 165166  101367                         BHI     CPUERR          ;"  "  C OR Z=0
 165170  000257                         CCC                     ;CLR ALL CONDITION CODES
 165172  001765                         BEQ     CPUERR          ;BR Z=1
 165174  100764                         BMI     CPUERR          ;BR N=1
 165176  102763                         BVS     CPUERR          ;BR V=1
 165200  103762                         BCS     CPUERR          ;BR C=1
 165202  003761                         BLE     CPUERR          ;BR Z OR (N XOR V)=1
 165204  101760                         BLOS    CPUERR          ;BR C OR Z=1
 165206  000270                         SEN                     ;N=1
 165210  002356                         BGE     CPUERR          ;BR N XOR V=0

                                ;       CHECK BYTE INSTRUCTIONS, ALL DEST MODES

                                                                ;R1 CONTENTS    N Z V C

 165212  105001                         CLRB    R1              ;177400         0 1 0 0
 165214  105201                         INCB    R1              ;177401         0 0 0 0
 165216  105101                         COMB    R1              ;177776         1 0 0 1
 165220  106201                         ASRB    R1              ;177777         1 0 1 0
 165222  106301                         ASLB    R1              ;177776         1 0 0 1
 165224  106001                         RORB    R1              ;177777         1 0 1 0
 165226  105401                         NEGB    R1              ;177401         0 0 0 1
 165230  105301                         DECB    R1              ;177400         0 1 0 1
 165232  105601                         SBCB    R1              ;177777         1 0 0 1
 165234  106101                         ROLB    R1              ;177777         1 0 0 1
 165236  105501                         ADCB    R1              ;177400         0 1 0 1
 165240  000301                         SWAB    R1              ;000377         1 0 0 0
 165242  012703  000500                 MOV     #500,   R3      ;SETUP FOR DMODE TESTING
 165246  105063  000001                 CLRB    1(R3)           ;CLR LOC 501, DMODE 6
 165252  110113                         MOVB    R1,     (R3)    ;500/000 377,DMODE 1
 165254  120123                         CMPB    R1,     (R3)+   ;SHOULD COMPARE, DMODE 2
 165256  001333                         BNE     CPUERR
 165260  105143                         COMB    -(R3)           ;500/000 000, DMODE 4
 165262  012703  165140                 MOV     #DOUBLE,R3      ;SETUP DEFERRED DMODES
 165266  153733  165136                 BISB    @#DATA1,@(R3)+  ;500/1, DMODE 3
 165272  143753  165132                 BICB    @#DATA2,@-(R3)  ;500/0, DMODE 5
 165276  130173  000002                 BITB    R1,     @2(R3)  ;Z=1, DMODE 7
 165302  001321                         BNE     CPUERR
```

```
                                ;       CHECK JSR/RTS AND EIS

165304  012706  000500                  MOV     #500,   SP      ;SET UP STACK
165310  004767  000004                  JSR     PC,     EISTST
165314  000714                          BR      CPUERR
165316  000421                          BR      MEMTST          ;EXIT CPU TEST
165320  012701  000040          EISTST: MOV     #40,    R1      ;R1/    40
165324  070127  000010                  MUL     #10,    R1      ;R1/   400
165330  006700                          SXT     R0              ;R0/     0
165332  072127  000006                  ASH     #6,     R1      ;R1/ 40000
165336  073127  000071                  ASHC    #71,    R1      ;R1/   200
165342  071027  000200                  DIV     #200,   R0      ;R0/1  R1/0
165346  005201                          INC     R1              ;R1/1
165350  074001                          XOR     R0,     R1      ;R1/0
165352  001275                          BNE     CPUERR
165354  062716  000002                  ADD     #2,     (SP)    ;FIX RETURN ADDRESS TO BYPASS ERROR BR
165360  000207                          RTS     PC              ;EXIT TO MEMORY TEST
```

```
                        ;;*****************************************************************

                        .SBTTL  MEMTST

                        ;;*****************************************************************

                        ;+
                        ;       THIS TEST SIZES MEMORY AND CHECKS MEMORY FROM LOC 1000 TO END OF
                        ;       MEMORY BY WRITING IN EACH LOCATION THE ADDRESS OF THE LOCATION
                        ;       AND COMPARING THE  LOCATION AND ITS CONTENTS. THE PROCEDURE IS
                        ;       REPEATED USING THE COMPLEMENTS OF THE ADDRESS IN EACH LOCATION.
                        ;       MEMORY IS CLEARED ON EXIT FROM THIS TEST IN 4K BLOCKS;THAT IS,
                        ;       IF 122K IS PRESENT, 120K GETS CLEARED. THE REST HAS THE ADDRESS
                        ;       WRITTEN IN IT.
                        ;-

                        ;;*****************************************************************

65362  012737  000012  177566  MEMTST: MOV     #12,@#XBUF      ;OUTPUT A LINE FEED .
65370  012702  000002                  MOV     #2,     R2      ;WORD INCREMENT
65374  010237  177570                  MOV     R2,     @#SWR   ;LIGHTS = 2, INDICATING MEMTST
65400  005037  172340                  CLR     @#PAR0          ;MAP VECTOR SPACE TO PAR0
65404  005037  172342                  CLR     @#PAR1          ;PAR1 IS MOVABLE WINDOW INTO MEMORY
65410  012737  177600  172356          MOV     #177600,@#PAR7  ;MAP PAR7 TO I/O PAGE
65416  012701  077406                  MOV     #77406,R1       ;SET UP R1 AS CONSTANT TO SETUP PDR'S
65422  010137  172300                  MOV     R1,@#PDR0       ;PAGE0=4K R/W
65426  010137  172302                  MOV     R1,@#PDR1       ;PAGE1=4K R/W
65432  010137  172316                  MOV     R1,@#PDR7       ;PAGE7=4K R/W
65436  005037  177776                  CLR     @#PS            ;ENSURE KERNEL MODE
65442  005237  177572                  INC     @#SR0           ;TURN ON KT
65446  012737  000020  172516          MOV     #20,    @#SR3   ;SET UP FOR 22-BIT RELOCATION
65454  012737  165552  000004  1$:     MOV     #TIMOUT,@#4     ;SETUP TIMEOUT VECTOR FOR MEMORY SIZE
65462  012703  020000                  MOV     #20000, R3      ;START AT VIRTUAL ZERO
65466  012705  010000                  MOV     #10000, R5      ;PAGE LENGTH = 4K
65472  010301                          MOV     R3,     R1      ;WORKING COPY OF FIRST ADDRRESS
65474  010500                          MOV     R5,     R0      ;COPY PAGE LENGTH
65476  010111          2$:     MOV     R1,     (R1)    ;WRITE ADDRESS INTO LOCATION
65500  060201                  ADD     R2,     R1      ;INCREMENT ADDRESS
65502  077003                  SOB     R0,     2$      ;LOOP TILL PAGE DONE
65504  010301                  MOV     R3,     R1      ;RESTORE INITIAL CONDITIONS FOR
65506  010500                  MOV     R5,     R0      ;   DATA CHECK AND COMPLEMENTING
65510  020111          3$:     CMP     R1,     (R1)    ;GOOD DATA?
65512  001014                  BNE     MEMERR          ;NO, HALT
65514  005121                  COM     (R1)+           ;COM DATA AND INC ADDRESS
65516  077004                  SOB     R0,     3$      ;LOOP TILL PAGE DONE
65520  010301                  MOV     R3,     R1      ;START AGAIN TO TEST COM DATA
65522  010500                  MOV     R5,     R0
65524  060111          4$:     ADD     R1,     (R1)    ;RESULT SHOULD BE -1
65526  005221                  INC     (R1)+           ;RESULT=0 AND SETUP NEXT ADDRESS
65530  001005                  BNE     MEMERR
65532  077004                  SOB     R0,     4$      ;FINISH THE PAGE
65534  062737  000200  172342          ADD     #200,   @#PAR1  ;RELOC TO A NEW PAGE
65542  000744                  BR      1$              ;CHECK OUT A NEW PAGE
65544  005037  177572  MEMERR: CLR     @#SR0           ;TURN OFF KT
65550  000000                  HALT                    ;BAD MEMORY
```

```
                                TIMOUT: CLR     @#SR0           ;TURN OFF KT
                                        CLR     @#SR3           ;RESTORE 18-BIT RELOCATION
                                        BR      1$              ;BRANCH AROUND ENTRY POINT
                                        JMP     CPUTST          ;BOOT ROM ENTRY POINT
                                1$:     MOV     #6,     @#4     ;TIMEOUT TRAPCATCHER
                                        SUB     R3,     R1      ;GET RID OF VIRTUAL OFFSET
                                        CLR     R0              ;CLR HIGH HALF OF DOUBLEWORD
                                        ASHC    #12,    R0      ;LSH 10. TO BUILD PHYSICAL ADDRESS
                                        ADD     @#PAR1, R0      ;ADD PAGE BASE ADDRESS
                                        ASHC    #-2,    R0      ;ACCOUNT FOR 1-BIT MSD
                                .SBTTL  SLU1 TEST

                                ;;**************************************************************

                                ;       CHECK SLU1 VIA MAINTENANCE MODE

                                ;;**************************************************************

                                SLUTST: MOV     #1,     @#SWR   ;LIGHTS = 1, INDICATING SLUTST
                                        CLR     R2              ;FIRST ASCII CODE TO BE CHECKED
                                        MOV     #4,     @#XCSR  ;ENABLE MAINTENANCE MODE
                                2$:     TSTB    @#XCSR          ;TRANSMITTER READY?
                                        BPL     2$              ;NO, WAIT FOR READY
                                        MOVB    R2,     @#XBUF  ;TRANSMIT CHAR
                                3$:     TSTB    @#RCSR          ;DATA RECEIVED?
                                        BPL     3$              ;NO
                                        CMPB    R2,     @#RBUF  ;DID DATA LOOP AROUND CORRECTLY?
                                        BNE     SLUERR          ;NO
                                        INC     R2              ;SET UP NEXT PATTERN
                                        TSTB    R2              ;DONE ALL PATTERNS?
                                        BNE     2$              ;NO, CONTINUE
                                        CLR     @#XCSR          ;EXIT MAINTENANCE MODE
                                        BR      PRINT           ;PRINT LAST ADDRESS
                                SLUERR: CLR     @#XCSR          ;EXIT MAINT MODE
                                        HALT
```

```
                                ;;*****************************************************************
                                ;       PRINT MEMORY SIZE (LAST ADDRESS + 2) AND EXIT
                                ;;*****************************************************************
165704  005037  177570  PRINT: CLR     @#SWR           ;LIGHTS = 0, ALL TESTS PASSED
165710  012705  000010         MOV     #10.,   R5      ;PRINT 8. DIGITS
165714  012703  000003  1$:    MOV     #3.,    R3      ;SHIFT COUNT FOR DIGIT ASSEMBLY
165720  005002                 CLR     R2              ;CLR DIGIT ASSEMBLY AREA
165722  073027  000001  2$:    ASHC    #1.,    R0      ;PUT BIT IN C-BIT
165726  006102                 ROL     R2              ;MOVE C-BIT TO R2
165730  077304                 SOB     R3,     2$      ;BUILD A DIGIT IN R2
165732  062702  000060         ADD     #'0,    R2      ;CNVRT TO ASCII
165736  105737  177564  3$:    TSTB    @#XCSR          ;PRINTER READY?
165742  100375                 BPL     3$
165744  110237  177566         MOVB    R2,     @#XBUF  ;PRINT IT
165750  077517                 SOB     R5,     1$      ;PRINT 8. DIGITS
165752  012705  120000         MOV     #120000,R5              ;GIVE PRINTER TIME TO PRINT
165756  077501  4$:    SOB     R5,     4$      ;OUT THE LAST TWO DIGITS
                                ;////////////////////////////////////////////////////////////////
                                ;       EXIT TO ODT OR TO BOOT ROM
                                ;////////////////////////////////////////////////////////////////
165760  013700  177640         MOV     @#UPAR0, R0     ;RESTORE BOOT ROM PARAMETERS
165764  013701  177642         MOV     @#UPAR1, R1
165770  000164  000002         JMP     2(R4)           ;RETURN TO BOOT ROM OR ODT
165774  042060                 .ASCII  "0D"            ;#4 CPU ROM FOR M9312
165776  051560         CRC:    .WORD   051560          ;CRC CHECKWORD FOR ROM

        165000                 .END    START
```

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0016

```
BADADD  165144            414     455#
CONT    165150            446     460#
CPUERR  165146            432     439     445     456#    461     462     463     464     465     466     467     469     470
                          471     472     473     474     476     498     504     510     520
CPUTST  165012            413#    582
CRC     165776            645#
DATA1   165136            449     452#    501
DATA2   165132            450#    451     502
DOUBLE  165140            453#    500
EISTST  165320            509     512#
MEMERR  165544            566     573     577#
MEMTST  165362            511     542#
PAR0  = 172340            391#    545*
PAR1  = 172342            392#    546*    575*    587
PAR7  = 172356            393#    547*
PDR0  = 172300            394#    549*
PDR1  = 172302            395#    550*
PDR7  = 172316            396#    551*
POOL    165126            436     448#
PRINT   165704            611     620#
PS    - 177776            383#    552*
RBUF  = 177562            386#    605
RCSR  = 177560            385#    603
SLUERR  165676            606     612#
SLUTST  165616            597#
SR0   = 177572            399#    553*    577*    579*
SR3   = 172516            400#    554*    580*
START   165000            402#    647
SWR   = 177570            382#    413*    544*    597*    620*
TIMOUT  165552            555     579#
UPAR0 = 177640            397#    416*    641
UPAR1 = 177642            398#    417*    642
XBUF  = 177566            388#    542*    602*    630*
XCSR  = 177564            387#    599*    600     610*    612*    628
$HD   = 000003            378     379
$SWR  = 160000            378     379#
$TN   = 000001            378#
.     = 166000            401#
```

```
COMMEN      1#
ENDCOM      1#
ESCAPE      1#
GETPRI      1#
GETSWR      1#
MULT        1#
NEWTST      1#
POP         1#
PUSH        1#
REPORT      1#
SETPRI      1#
SETUP       1#
SKIP        1#
SLASH       1#    368#    635     639
STARS       1#    368#    408     412     523     527     540     591     595     614     618
SWRSU       1#
TYPBIN      1#
TYPDEC      1#
TYPNAM      1#
TYPNUM      1#
TYPOCS      1#
TYPOCT      1#
TYPTXT      1#
$$ESCA      1#
$$NEWT      1#
$$SKIP      1#
.EQUAT      1#
.HEADE      1#    368#
.KT11       1#
.SETUP      1#
.SWRHI      1#
.$ACT1      1#
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#
.$CMTA      1#
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#
```

```
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$40CA      1#
.1170       1#


. ABS.  166000     000


 ERRORS DETECTED:  0

 CJM9AB,CJM9AB/CRF/NL:TOC/SOL=SYSMAC.SML,CJM9AB.P11
 RUN-TIME:  5  6  .2 SECONDS
 RUN-TIME RATIO:  39/12=3.1
 CORE USED:  34K  (68 PAGES)
```