.REM %

IDENTIFICATION

PRODUCT CODE: AC-T207B-MC
PRODUCT NAME: CJM9AB0 11/24 ROM M9312
PRODUCT DATE: SEPTEMBER, 1982
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: J. HAMEL

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C): 1982, 1983 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL    PDP    UNIBUS    MASSBUS
DEC        DECUS   DECTAPE   DECX/11
HISTORY

CJM9AA - DOCUMENT RELEASED APRIL 1982
CJM9AB - CHANGES MADE SO DIAGNOSTIC WOULD INTERFACE CLEANLY WITH CONSOLE EMULATOR. AND FIXED PROBLEM OF LOSING LAST TWO 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. SLUT 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-D2QCAC-C3).

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

NONE
1.5 ASSUMPTIONS

The diagnostic assumes proper stepup of the dip switchpack 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.

<table>
<thead>
<tr>
<th>Bootstrap</th>
<th>Diagnostics</th>
<th>First Device (All ROMs)</th>
<th>Virtual Address</th>
<th>Switchpack S1 Switches On</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODT</td>
<td>NO</td>
<td>2004</td>
<td>165004</td>
<td>1,9,10</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>2006</td>
<td>165006</td>
<td></td>
</tr>
<tr>
<td>Device ROM #1</td>
<td>NO</td>
<td>0004</td>
<td>173004</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>0006</td>
<td>173006</td>
<td></td>
</tr>
<tr>
<td>Device ROM #2</td>
<td>NO</td>
<td>0204</td>
<td>173204</td>
<td>4,9,10</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>0206</td>
<td>173206</td>
<td></td>
</tr>
<tr>
<td>Device ROM #3</td>
<td>NO</td>
<td>0404</td>
<td>173404</td>
<td>3,9,10</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>0406</td>
<td>173406</td>
<td></td>
</tr>
<tr>
<td>Device ROM #4</td>
<td>NO</td>
<td>0604</td>
<td>173604</td>
<td>3,4,9,10</td>
</tr>
<tr>
<td></td>
<td>YES</td>
<td>0606</td>
<td>173606</td>
<td></td>
</tr>
</tbody>
</table>

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 #5 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/no go, low-level test, no error
reporting, as such, is implemented. However, if the micro-gdt
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 since the program has not set it up the
   information received may not be valid.

CPRERR = 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(#) 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


<table>
<thead>
<tr>
<th>LIGHT COUNT</th>
<th>LAST TEST COMPLETED</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>NONE-SUCCESSFUL ENTRY TO PROGRAM</td>
</tr>
<tr>
<td>2</td>
<td>CPU TEST</td>
</tr>
<tr>
<td>1</td>
<td>MEMORY TEST</td>
</tr>
<tr>
<td>0</td>
<td>SLU TEST</td>
</tr>
</tbody>
</table>

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 ENTERING MICRO-ODT (DEPENDING ON M9312 SWITCH SETTINGS).

6.2 SUBTEST SUMMARIES

A. CPU TEST - 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 SLUT 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 PRINTOUT.

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.SOBIEK
  ;*
  ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
  ;*PACKAGE (MAINDEC-11-D26AC-C5), JAN, 1981.
  ;*
  .STN=1
  $SWR=$60000  ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPEOUT

  .SBTTL REGISTER DEFINITIONS
  .SWR= 177570  ;;SWITCH REGISTER (DIAGNOSTIC LIGHTS)
  .PS= 177776  ;;PROCESSOR STATUS WORD
  .SLUT REGISTERS
  .RCSR= 177560  ;;RECEIVER CSR
  .RBUF= 177562  ;;RECEIVER BUFFER
  .XCSR= 177564  ;;TRANSMITTER CSR
  .XBUF= 177566  ;;TRANSMITTER BUFFER
  .MMU REGISTERS
  .PARO= 172340  ;;KERNEL PAGE ADDRESS Registers
  .PAR1= 172342
  .PAR2= 172356
  .PAR3= 172360
  .PDRO= 172300  ;;KERNEL PAGE DESCRIPTOR Registers
  .PDRI= 172302
  .PD2= 172316
  .PD3= 172320
  .UPA0= 177640  ;;USER PAGE ADDRESS Registers
  .UPA1= 177642
  .SR0= 177572  ;;STATUS REGISTER 0
  .SR3= 172516  ;;STATUS REGISTER 3 (22-BIT)
BC 11/24 ROM M9312 MBCY 11 30(1046) 15-SEP-82 17:04 PAGE 11 K 1

REGISTER DEFINITIONS

START: JMP @173024 ;TRANSFER TO SELECTED BOOT ROM OR ODT
HALT ;ENTRY POINT FOR ODT. NO DIAGNOSTICS
ENTRY POINT FOR ODT WITH DIAGNOSTICS
MOV #165002,R4 ;SET UP RETURN ADDRESS-2 IN R4

;SBTL CPU TEST
;
;************************************************************
;
;BASIC CPU TEST
;
;*************************************************************

;PUTST: MOV #3, @SWR ;LIGHTS = 3, INDICATING CPU STS
MOV @BADADD,#4 ;SET UP TIME OUT VECTOR IN CASE OF TRAP
CLR #6 ;CLEAR PRIORITY OF TRAP ROUTINE

MOV R0, @#UPAR0 ;WE ARE USING THE UPAR'S HERE SIMPLY BECAUSE
MOV R1, @#UPAR1 ;THEY ARE AVAILABLE UNUSED INTERNAL REGS.

;WE ARE STORING PARAMETERS FROM THE BOOT ROM

R1 CONTENTS N I V C

CLR R1
INC R1
COM R1
ASR R1
ASL R1
ROR R1
TST R1
NEG R1
DEC R1
SBC R1
ADC R1
BNE CPUERR ;ERROR IF NOT ZERO

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

MOV #POOL, R2 ;SET UP ADDRESS OF DATA TABLE
MOV (R2), R1 ;R1/POOL, SMODE 1
CMP (R2)+, R1 ;DATA CORRECT?? SMODE 2
BNE CPUERR

BIC R1
SUB #R2, R1
BIT 80(R2), R1
BEQ CPUERR ;RESULT IS 177777, SMODE 7

BR CONT ;BRANCH AROUND DATA TABLE

POOL: .WORD POOL
DATA1: .WORD 1
DATA2: .WORD 177777
DOUBLE: .WORD 500
WORD 500
BADADD: HALT
CPUERR: HALT
; CHECK CONDITIONAL BRANCHES
; CONT: SCC CPUERR ;SET ALL CONDITION CODES
BNE CPUERR ;BR IF Z=0
BPL CPUERR ;... N=0
BMI CPUERR ;... V=0
BCC CPUERR ;... C=0
BLE CPUERR ;... Z OR (N XOR V)=1
BGT CPUERR ;... Z OR (N XOR V)=0
BMI CPUERR ;... C OR Z=0
; CCC CPUERR ;CLR ALL CONDITION CODES
BGE CPUERR ;BR Z=1
BEQ CPUERR ;BR N=1
BVS CPUERR ;BR V=1
BHS CPUERR ;BR C=1
BPL CPUERR ;BR Z OR (N XOR V)=1
BMI CPUERR ;BR C OR Z=1
; SEN CPUERR ;N=1
BGE CPUERR ;BR N XOR V=0

; CHECK BYTE INSTRUCTIONS, ALL DEST MODES
; R1 CONTENTS N Z V C
CLR R1 ;177400 0 1 0 0
INC R1 ;177401 0 0 0 0
COM R1 ;177776 1 0 0 1
ASR R1 ;177777 1 0 1 0
AAS R1 ;177776 1 0 0 1
ROR R1 ;177777 1 0 1 0
NEG R1 ;177401 0 0 0 1
DEC R1 ;177777 1 0 0 1
SBC R1 ;177777 0 1 0 1
ROL R1 ;177400 0 1 0 1
ADC R1 ;177777 1 0 0 1
SWAP R1 ;177777 1 0 0 1
MOV #500, R3 ;SETUP FOR DMODE TESTING
CLR (R3) ;CLR LOC 501, DMODE 6
MOV R1, (R3) ;500/000 377,DMODE 1
CMP R1, (R3) ;SHOULD COMPARE, DMODE 2
BNE CPUERR
COMB -(R3) ;DMODE 4
MOV #DOUBLE.R3 ;SETUP DEFERRED DMODES
BISB @DATA1,R3 ;500/1, DMODE 5
BICB @DATA2,B-(R3) ;500/0, DMODE 5
BITB R1, @2(R3) ;Z=1, DMODE 7
; CHECK JSR/RTS AND EIS

MOV #500, SP          ; SET UP STACK
JSR PC, EISTST
BR CPUERR
BR MEMST

EISTST: MOV #40, R1   ; R1/40
         MUL #10, R1   ; R1/400
         SXT R0       ; R0/0
         ASH #6, R1   ; R1/40000
         ASHC #71, R1 ; R1/200
         DIV #200, R0 ; R0/1 R1/0
         INC R1       ; R1/1
         XOR R0, R1   ; R1/0
         BNE CPUERR
ADD #2, (SP)          ; FIX RETURN ADDRESS TO BYPASS ERROR BR
RTS PC                ; EXIT TO MEMORY TEST
.SBTTL MEMST

* 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.

MENST: MOV #12, @XBUF ; OUTPUT A LINE FEED.
MOV #2, R2 ; WORD INCREMENT
MOV R2, @SWR ; LIGHTS = 2, INDICATING MEMST
CLR @PARQ ; MAP VECTOR SPACE TO PARQ
CLR @PAR7 ; PAR7 IS MOVABLE WINDOW INTO MEMORY
MOV #177600, @PAR7 ; MAP PAR7 TO I/O PAGE
MOV #77406, A1 ; SET UP R1 AS CONSTANT TO SETUP PDR'S
MOV R1, @PDIR0 ; PAGE=4K R/W
MOV R1, @PDIR1 ; PAGE=4K R/W
MOV R1, @PDIR7 ; PAGE=4K R/W
CLR @PS ; ENSURE KERNEL MODE
INC @SR0 ; TURN ON KT
MOV #20, @SR3 ; SET UP FOR 22-BIT RELOCATION
MOV #17756, @PAR5 ; SETUP TIMEOUT VECTOR FOR MEMORY SIZE
MOV #200000, R3 ; START AT VIRTUAL ZERO
MOV #100000, R5 ; PAGE LENGTH = 4K
MOV R3, R1 ; WORKING COPY OF FIRST ADDRESS
MOV R3, R0 ; COPY PAGE LENGTH
2$: MOV R1, (R1) ; WRITE ADDRESS INTO LOCATION
ADD R2, R1 ; INCREMENT ADDRESS
SOF R0, 2$ ; LOOP TILL PAGE DONE
MOV R3, R1 ; RESTORE INITIAL CONDITIONS FOR
MOV R5, R0 ; DATA CHECK AND COMPLEMENTING
3$: CMP R1, (R1) ; GOOD DATA?
BNE MEMERR ; NO, HALT
COM (R1) ; COM DATA AND INC ADDRESS
SOF R0, 3$ ; LOOP TILL PAGE DONE
MOV R3, R1 ; START AGAIN TO TEST COM DATA
MOV R5, R0
4$: ADD R1, (R1) ; RESULT SHOULD BE ±?
BNE MEMERR ; NO AND SETUP NEXT ADDRESS
INC (R1) ; SETUP NEXT ADDRESS
BNE MEMERR
SOF R0, 4$ ; FINISH THE PAGE
ADD #200, @PAR1 ; RELOC TO A NEW PAGE
BR 1$ ; CHECK OUT A NEW PAGE
CLRM R0, 4$ ; TURN OFF KT
MEMERR: CLR @SR0 ; BAD MEMORY
CJM9AB0 11/24 ROM M9312 MACY11 30(M46) 15-SEP-82 17:04 PAGE 15
CJM9AB.P11 15-SEP-82 17:03

TIMOUT: CLR #SRO ; TURN OFF RT
CLR #SR3 ; RESTORE 18-BIT RELOCATION
BR 15 ; BRANCH AROUND ENTRY POINT
JMP CPUTST ; BOOT ROM ENTRY POINT
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 SLUT TEST

;-----------------------------------------------
; CHECK SLUT 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
TSTB #XCSR ; TRANSMITTER READY?
BPL #XBUF 2 ; NO, WAIT FOR READY
MOV R2, #XBUF 3 ; TRANSMIT CHAR
TSTB #XCSR ; DATA RECEIVED?
BPL #RBUF 4 ; NO
CMPB R2, #RBUF 5 ; DATA LOOP AROUND CORRECTLY?
BNE #SLUERR 6 ; NO
INC R2 ; SET UP NEXT PATTERN
TSTB R2 ; DONE ALL PATTERNS?
BNE #SLUERR 7 ; NO
CLR #XCSR ; EXIT MAINTENANCE MODE
PRINT #PRINT LAST ADDRESS
HALT

;-----------------------------------------------

SEQ 0014
;;;*****************************************************************************

;; PRINT MEMORY SIZE (LAST ADDRESS + 2) AND EXIT

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

PRINT: CLR @SWR: LIGHTS = 0, ALL TESTS PASSED

1$: MDV #10, R5: PRINT 8. DIGITS
	MDV #3, R3: SHIFT COUNT FOR DIGIT ASSEMBLY
	CLR R2: CLR DIGIT ASSEMBLY AREA

2$: ASHC #1, R0: PUT BIT IN C-BIT
	ROL R2: MOVE C-BIT TO R2
	SDB R3, 2$: BUILD A DIGIT IN R2
	ADD #0, R2: CVRT TO ASCII

3$: TSTB @#XCSR: PRINTER READY?
	BPL 3$: 

4$: MOV R8, @#XBUF: PRINT IT
	SDB R5, 1$: PRINT 8. DIGITS
	MOV #120000, R5: GIVE PRINTER TIME TO PRINT
	SDB R5, 4$: OUT THE LAST TWO DIGITS

; EXIT TO ODT OR TO BOOT ROM

;*****************************************************************************
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>BADADD</td>
<td>165144</td>
</tr>
<tr>
<td>CONT</td>
<td>165150</td>
</tr>
<tr>
<td>CPUERR</td>
<td>165146</td>
</tr>
<tr>
<td>CPUTST</td>
<td>165012</td>
</tr>
<tr>
<td>CRC</td>
<td>165776</td>
</tr>
<tr>
<td>DATA1</td>
<td>165136</td>
</tr>
<tr>
<td>DATA2</td>
<td>165152</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>165140</td>
</tr>
<tr>
<td>EISTST</td>
<td>165320</td>
</tr>
<tr>
<td>MEM</td>
<td>165544</td>
</tr>
<tr>
<td>MEMTST</td>
<td>165362</td>
</tr>
<tr>
<td>PRO</td>
<td>172360</td>
</tr>
<tr>
<td>PART</td>
<td>172362</td>
</tr>
<tr>
<td>PART2</td>
<td>172356</td>
</tr>
<tr>
<td>PDRO</td>
<td>172300</td>
</tr>
<tr>
<td>PDRI</td>
<td>172302</td>
</tr>
<tr>
<td>PD7</td>
<td>172316</td>
</tr>
<tr>
<td>POOL</td>
<td>165126</td>
</tr>
<tr>
<td>PRINT</td>
<td>165704</td>
</tr>
<tr>
<td>PS</td>
<td>177776</td>
</tr>
<tr>
<td>RBUF</td>
<td>177562</td>
</tr>
<tr>
<td>RCSR</td>
<td>177560</td>
</tr>
<tr>
<td>SLUER</td>
<td>165676</td>
</tr>
<tr>
<td>SUTST</td>
<td>165616</td>
</tr>
<tr>
<td>SRQ</td>
<td>177572</td>
</tr>
<tr>
<td>SR3</td>
<td>172516</td>
</tr>
<tr>
<td>START</td>
<td>165000</td>
</tr>
<tr>
<td>SWR</td>
<td>177570</td>
</tr>
<tr>
<td>TIMOUT</td>
<td>165552</td>
</tr>
<tr>
<td>UPA0</td>
<td>177640</td>
</tr>
<tr>
<td>UPA1</td>
<td>177642</td>
</tr>
<tr>
<td>XBUF</td>
<td>177566</td>
</tr>
<tr>
<td>XLSR</td>
<td>177564</td>
</tr>
<tr>
<td>$HD</td>
<td>000003</td>
</tr>
<tr>
<td>$SWR</td>
<td>160000</td>
</tr>
<tr>
<td>$TN</td>
<td>000001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>166000</td>
</tr>
<tr>
<td>MACRO NAME</td>
<td>368#</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>COMREM</td>
<td></td>
</tr>
<tr>
<td>ENDCOM</td>
<td></td>
</tr>
<tr>
<td>ESCAPE</td>
<td></td>
</tr>
<tr>
<td>GETPRI</td>
<td></td>
</tr>
<tr>
<td>GETSWR</td>
<td></td>
</tr>
<tr>
<td>MULTI</td>
<td></td>
</tr>
<tr>
<td>NETST</td>
<td></td>
</tr>
<tr>
<td>POP</td>
<td></td>
</tr>
<tr>
<td>PUSH</td>
<td></td>
</tr>
<tr>
<td>REPORT</td>
<td></td>
</tr>
<tr>
<td>SE TPRI</td>
<td></td>
</tr>
<tr>
<td>SETUP</td>
<td></td>
</tr>
<tr>
<td>SKIP</td>
<td></td>
</tr>
<tr>
<td>SLASH</td>
<td></td>
</tr>
<tr>
<td>STARS</td>
<td>368#</td>
</tr>
<tr>
<td>SWRSU</td>
<td></td>
</tr>
<tr>
<td>TYPBIN</td>
<td></td>
</tr>
<tr>
<td>TYPDECL</td>
<td></td>
</tr>
<tr>
<td>TYPNAME</td>
<td></td>
</tr>
<tr>
<td>TYPNUM</td>
<td></td>
</tr>
<tr>
<td>TYPPOCS</td>
<td></td>
</tr>
<tr>
<td>TYPDOC</td>
<td></td>
</tr>
<tr>
<td>TYPTEXT</td>
<td></td>
</tr>
<tr>
<td>S$ESCA</td>
<td></td>
</tr>
<tr>
<td>S$NEWT</td>
<td></td>
</tr>
<tr>
<td>S$SKIP</td>
<td></td>
</tr>
<tr>
<td>.EQUAT</td>
<td></td>
</tr>
<tr>
<td>.HEAD</td>
<td></td>
</tr>
<tr>
<td>.EESREM</td>
<td></td>
</tr>
<tr>
<td>.SETUP</td>
<td></td>
</tr>
<tr>
<td>.SWRHI</td>
<td></td>
</tr>
<tr>
<td>.SACT1</td>
<td></td>
</tr>
<tr>
<td>.SAPTH</td>
<td></td>
</tr>
<tr>
<td>.SAPTY</td>
<td></td>
</tr>
<tr>
<td>.SASTA</td>
<td></td>
</tr>
<tr>
<td>.SCATC</td>
<td></td>
</tr>
<tr>
<td>.SCIFA</td>
<td></td>
</tr>
<tr>
<td>.SDB2D</td>
<td></td>
</tr>
<tr>
<td>.SDB2D</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td></td>
</tr>
<tr>
<td>.SDD</td>
<td>368#</td>
</tr>
</tbody>
</table>
CROSS REFERENCE TABLE -- MACRO NAMES

.\$UPR      1#
.\$TRAP     1#
.\$TPB      1#
.\$TPD      1#
.\$TYPE     1#
.\$TPD     1#
.\$LCA      1#
.1170        1#

. ABS. 166000  000

ERRORS DETECTED: 0

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