

11/24

11/24 ROM M9312
CJM9AA0

AH-T208A-MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT © 1982
MADE IN USA



.REM %

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

IDENTIFICATION

PRODUCT CODE: AC-T207A-MC
PRODUCT NAME: CJM9AA0 11/24 ROM M9312
PRODUCT DATE: APRIL,1982
MAINTAINER: DIAGNOSTIC ENGINEERING

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 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	DECX/11

50
51
52
53

HISTORY

TABLE OF CONTENTS

55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89

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	LISTING

91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

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

NGNE

146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201

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.

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

203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258

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 SINCE 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 PAR0 (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.

260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311

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 ENTERING 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.

313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352

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 LISTING

```

354 %
355
356 .TITLE CJM9AA
357 .SBTTL ROM AREA 165000-165776
358 .SBTTL GO/NOGO MINIMUM DIAGNOSTIC
365 .TITLE CJM9AA0 11/24 ROM M9312
(1) :*COPYRIGHT (C) APRIL, 1982
(1) :*DIGITAL EQUIPMENT CORP.
(1) :*MAYNARD, MASS. 01754
(1) :*
(1) :*PROGRAM BY D. SOBIK
(1) :*
(1) :*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
(1) :*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
(1) :*
(1) $TN=1
(1) 000001
(1) 160000 $SWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
366
367 .SBTTL REGISTER DEFINITIONS
368 177570 SWR= 177570 ;SWITCH REGISTER (DIAGNOSTIC LIGHTS)
369 177776 PS= 177776 ;PROCESSOR STATUS WORD
370 ;SLU1 REGISTERS
371 177560 RCSR= 177560 ;RECEIVER CSR
372 177562 RBUF= 177562 ;RECEIVER BUFFER
373 177564 XCSR= 177564 ;TRANSMITTER CSR
374 177566 XBUF= 177566 ;TRANSMITTER BUFFER
375
376 ;MMU REGISTERS
377 172340 PAR0= 172340 ;KERNAL PAGE ADDRESS REGISTERS
378 172342 PAR1= 172342
379 172356 PAR7= 172356
380 172300 PDR0= 172300 ;KERNAL PAGE DESCRIPTOR REGISTERS
381 172302 PDR1= 172302
382 172316 PDR7= 172316
383 177640 UPAR0= 177640 ;USER PAGE ADDRESS REGISTERS
384 177642 UPAR1= 177642
385 177572 SR0= 177572 ;STATUS REGISTER 0
386 172516 SR3= 172516 ;STATUS REGISTER 3 (22-BIT)
    
```



```

388      165000 000177 006020      .SMTTL CPU TEST
389      165000 000177 006020      :*****
390      165004 000000
391
392      165006 012704 165002      START:  .=165000
393
394
395
396
397
398
399
400      165012 012737 000003 177570  CPUTST: MOV    #3, @#SWR      ;LIGHTS = 3, INDICATING CPUTST
401      165020 012737 165144 000004      MOV    #BADADD,@#4      ;SET UP TIME OUT VECTOR INCASE OF TRAP
402      165026 005037 000006      CLR    @#6              ;CLEAR PRIORITY OF TRAP ROUTINE
403      165032 010037 177640      MOV    R0, @#UPAR0     ;WE ARE USING THE UPAR'S HERE SIMPLY BECAUSE
404      165036 010137 177642      MOV    R1, @#UPAR1     ;THEY ARE AVAILABLE UNUSED INTERNAL REGS.
405
406
407
408      165042 005001      CLR    R1              ;R1 CONTENTS
409      165044 005201      INC    R1              ;000000
410      165046 005101      COM    R1              ;1
411      165050 006201      ASR    R1              ;177776
412      165052 006301      ASL    R1              ;177777
413      165054 006001      ROR    R1              ;177777
414      165056 005701      TST    R1              ;177776
415      165060 005401      NEG    R1              ;177777
416      165062 005301      DEC    R1              ;1
417      165064 005601      SBC    R1              ;0
418      165066 005501      ADC    R1              ;0
419      165070 001026      BNE    CPUERR         ;ERROR IF NOT ZERO
420
421
422
423      165072 012702 165126      MOV    #POOL, R2      ;SET UP ADDRESS OF DATA TABLE
424      165076 011201      MOV    (R2), R1      ;R1/POOL, SMODE 1
425      165100 022201      CMP    (R2)+, R1     ;DATA CORRECT? SMODE 2
426      165102 001021      BNE    CPUERR
427      165104 063201      ADD    @-(R2)+, R1   ;R1/POOL + 1, SMODE 3
428      165106 165201      SUB    @-(R2), R1   ;R1/POOL, SMODE 5
429      165110 044201      BIC    -(R2), R1    ;R1/0, SMODE 4
430      165112 056201 000004      BIS    4(R2), R1    ;R1/177777, SMODE 6
431      165116 037201 000006      BIT    @6(R2), R1   ;RESULT IS 177777, SMODE 7
432      165122 001411
433      165124 000411      BEQ    CPUERR
434
435      165126 165126      BR    CONT           ;BRANCH AROUND DATA TABLE
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
    
```



```

445
446      :      CHECK CONDITIONAL BRANCHES
447
448 165150 000277      CONT:  SCC      :SET ALL CONDITION CODES
449 165152 001375      BNE      CPUERR  :BR IF Z=0
450 165154 100374      BPL      CPUERR  :.. .. N=0
451 165156 102373      BVC      CPUERR  :.. .. V=0
452 165160 103372      BCC      CPUERR  :.. .. C=0
453 165162 002771      BLT      CPUERR  :.. .. N XOR V=1
454 165164 003370      BGT      CPUERR  :.. .. Z OR (N XOR V)=0
455 165166 101367      BHI      CPUERR  :.. .. C OR Z=0
456 165170 000257      CCC      :CLR ALL CONDITION CODES
457 165172 001765      BEQ      CPUERR  :BR Z=1
458 165174 100764      BMI      CPUERR  :BR N=1
459 165176 102763      BVS      CPUERR  :BR V=1
460 165200 103762      BCS      CPUERR  :BR C=1
461 165202 003761      BLE      CPUERR  :BR Z OR (N XOR V)=1
462 165204 101760      BLOS     CPUERR  :BR C OR Z=1
463 165206 000270      SEN
464 165210 002356      BGE      CPUERR  :N=1
                                     :BR N XOR V=0
465
466      :      CHECK BYTE INSTRUCTIONS, ALL DEST MODES
467
468                                     :R1 CONTENTS      N Z V C
469
470 165212 105001      CLRB     R1      :177400          0 1 0 0
471 165214 105201      INCB     R1      :177401          0 0 0 0
472 165216 105101      COMB     R1      :177776          1 ^ 0 1
473 165220 106201      ASRB     R1      :177777          1   1 0
474 165222 106301      ASLB     R1      :177776          1 0 0 1
475 165224 106001      RORB     R1      :177777          1 0 1 0
476 165226 105401      NEGB     R1      :177401          0 0 0 1
477 165230 105301      DECB     R1      :177400          0 1 0 1
478 165232 105601      SBCB     R1      :177777          1 0 0 1
479 165234 106101      ROLB     R1      :177777          1 0 0 1
480 165236 105501      ADCB     R1      :177400          0 1 0 1
481 165240 000301      SWAB     R1      :000377          1 0 0 0
482 165242 012703 000500      MOV      #500, R3 :SETUP FOR DMODE TESTING
483 165246 105063 000001      CLRB     1(R3)   :CLR LOC 501, DMODE 6
484 165252 110113      MOVB     R1, (R3) :500/000 377, DMODE 1
485 165254 120123      CMPB     R1, (R3)+ :SHOULD COMPARE, DMODE 2
486 165256 001333      BNE      CPUERR
487 165260 105143      COMB     -(R3)   :500/000 000, DMODE 4
488 165262 012703 165140      MOV      #DOUBLE, R3 :SETUP DEFERRED DMODES
489 165266 153733 165136      BISB     @#DATA1, @ (R3)+ :500/1, DMODE 3
490 165272 143753 165132      BICB     @#DATA2, @-(R3) :500/0, DMODE 5
491 165276 130173 000002      BITB     R1, @2(R3) :Z=1, DMODE 7
492 165302 001321      BNE      CPUERR
    
```



```

494
495
496
497 165304 012706 000500
498 165310 004767 000006
499 165314 000714
500 165316 000167 000042
501 165322 012701 000040
502 165326 070127 000010
503 165332 006700
504 165334 072127 000006
505 165340 073127 000071
506 165344 071027 000200
507 165350 005201
508 165352 074001
509 165354 001274
510 165356 062716 000002
511 165362 000207
    
```

; CHECK JSR/RTS AND EIS

```

    MOV #500, SP ;SET UP STACK
    JSR PC, EISTST
    BR CPUERR
    JMP MEMTST ;EXIT CPU TEST
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
    
```

```

513      ;:*****
514
515      .SBTTL  MEMTST
516
517      ;:*****
518
519      ;+
520      ;:
521      ;:   THIS TEST SIZES MEMORY AND CHECKS MEMORY FROM LOC 1000 TO END OF
522      ;:   MEMORY BY WRITING IN EACH LOCATION THE ADDRESS OF THE LOCATION
523      ;:   AND COMPARING THE LOCATION AND ITS CONTENTS. THE PROCEDURE IS
524      ;:   REPEATED USING THE COMPLEMENTS OF THE ADDRESS IN EACH LOCATION.
525      ;:   MEMORY IS CLEARED ON EXIT FROM THIS TEST IN 4K BLOCKS; THAT IS,
526      ;:   IF 122K IS PRESENT, 120K GETS CLEARED, THE REST HAS THE ADDRESS
527      ;:   WRITTEN IN IT.
528      ;:-
529
530      ;:*****
531
532 165364 012737 000002 177570 MEMTST: MOV    #2,    @#SWR    ;LIGHTS = 2, INDICATING MEMTST
533 165372 005037 172340          CLR    @#PAR0  ;MAP VECTOR SPACE TO PAR0
534 165376 012737 077406 172300          MOV    #77406, @#PDR0 ;4K PAGE R/W
535 165404 005037 172342          CLR    @#PAR1  ;PAR1 IS MOVABLE WINDOW INTO MEMORY
536 165410 012737 077406 172302          MOV    #77406, @#PDR1
537 165416 012737 177600 172356          MOV    #177600, @#PAR7 ;MAP PAR7 TO I/O PAGE
538 165424 012737 077406 172316          MOV    #77406, @#PDR7
539 165432 005037 177776          CLR    @#PS    ;ENSURE KERNEL MODE
540 165436 005237 177572          INC    @#SR0   ;TURN ON KT
541 165442 012737 000020 172516          MOV    #20,   @#SR3  ;SET UP FOR 22-BIT RELOCATION
542 165450 012702 000002          MOV    #2,    R2    ;WORD INCREMENT
543 165454 012737 165552 000004 1$:  MOV    #TIMOUT, @#4   ;SETUP TIMEOUT VECTOR FOR MEMORY SIZE
544 165462 012703 020000          MOV    #20000, R3   ;START AT VIRTUAL ZERO
545 165466 012705 010000          MOV    #10000, R5   ;PAGE LENGTH = 4K
546 165472 010301          MOV    R3,     R1   ;WORKING COPY OF FIRST ADDRESS
547 165474 010500          MOV    R5,     R0   ;COPY PAGE LENGTH
548 165476 010111          MOV    R1,     (R1) ;WRITE ADDRESS INTO LOCATION
549 165500 060201          ADD    R2,     R1   ;INCREMENT ADDRESS
550 165502 077003          SOB    R0,     2$   ;LOOP TILL PAGE DONE
551 165504 010301          MOV    R3,     R1   ;RESTORE INITIAL CONDITIONS FOR
552 165506 010500          MOV    R5,     R0   ; DATA CHECK AND COMPLEMENTING
553 165510 020111          CMP    R1,     (R1) ;GOOD DATA?
554 165512 001014          BNE    MEMERR    ;NO, HALT
555 165514 005121          COM    (R1)+    ;COM DATA AND INC ADDRESS
556 165516 077004          SOB    R0,     3$   ;LOOP TILL PAGE DONE
557 165520 010301          MOV    R3,     R1   ;START AGAIN TO TEST COM DATA
558 165522 010500          MOV    R5,     R0
559 165524 060111          ADD    R1,     (R1) ;RESULT SHOULD BE -1
560 165526 005221          INC    (R1)+    ;RESULT=0 AND SETUP NEXT ADDRESS
561 165530 001005          BNE    MEMERR
562 165532 077004          SOB    R0,     4$   ;FINISH THE PAGE
563 165534 062737 000200 172342          ADD    #200,   @#PAR1 ;RELOC TO A NEW PAGE
564 165542 000744          BR     1$       ;CHECK OUT A NEW PAGE
565 165544 005037 177572          MEMERR: CLR    @#SR0  ;TURN OFF KT
566 165550 000000          HALT          ;BAD MEMORY
567 165552 005037 177572          TIMEOUT: CLR   @#SR0 ;TURN OFF KT
568 165556 005037 172516          CLR    @#SR3   ;RESTORE 18-BIT RELOCATION
    
```



```

569 165562 000402          BR      1$          ;BRANCH AROUND ENTRY POINT
570 165564 000167 177222  JMP     CPUTST      ;BOOT ROM ENTRY POINT
571 165570 012737 000006 000004 1$:  MOV     #6,        @#4    ;TIMEOUT TRAPCATCHER
572 165576 160301          SUB     R3,        R1     ;GET RID OF VIRTUAL OFFSET
573 165600 005000          CLR     R0         ;CLR HIGH HALF OF DOUBLEWORD
574 165602 073027 000012  ASHC   #12,       R0     ;LSH 10. TO BUILD PHYSICAL ADDRESS
575 165606 063700 172342  ADD     @#PAR1,   R0     ;ADD PAGE BASE ADDRESS
576 165612 073027 177776  ASHC   #-2,      R0     ;ACCOUNT FOR 1-BIT MSD
577          .SBTTL  SLU1 TEST
578
579          ;:*****
580
581          ;      CHECK SLU1 VIA MAINTENANCE MODE
582
583          ;:*****
584
585 165616 012737 000001 177570  SLUTST: MOV     #1,        @#SWR ;LIGHTS = 1, INDICATING SLUTST
586 165624 005002          CLR     R2         ;FIRST ASCII CODE TO BE CHECKED
587 165626 012737 000004 177564  MOV     #4,        @#XCSR ;ENABLE MAINTENANCE MODE
588 165634 105737 177564  2$:  TSTB   @#XCSR   ;TRANSMITTER READY?
589 165640 100375          BPL     2$         ;NO, WAIT FOR READY
590 165642 110237 177566  MOVB   R2,        @#XBUF ;TRANSMIT CHAR
591 165646 105737 177560  3$:  TSTB   @#RCSR   ;DATA RECEIVED?
592 165652 100375          BPL     3$         ;NO
593 165654 120237 177562  CMPB   R2,        @#RBUF ;DID DATA LOOP AROUND CORRECTLY?
594 165660 001006          BNE     SLUERR     ;NO
595 165662 005202          INC     R2         ;SET UP NEXT PATTERN
596 165664 105702          TSTB   R2         ;DONE ALL PATTERNS?
597 165666 001362          BNE     2$        ;NO, CONTINUE
598 165670 005037 177564  CLR     @#XCSR     ;EXIT MAINTENANCE MODE
599 165674 000403          BR      PRINT      ;PRINT LAST ADDRESS
600 165676 005037 177564  SLUERR: CLR    @#XCSR ;EXIT MAINT MODE
601 165702 000000          HALT
    
```

```

603
604
605
606
607
608
609 165704 005037 177570
610 165710 012705 000010
611 165714 012703 000003
612 165720 005002
613 165722 073027 000001
614 165726 006102
615 165730 077304
616 165732 062702 000060
617 165736 105737 177564
618 165742 100375
619 165744 110237 177566
620 165750 077517
621
622
623
624
625
626
627
628 165752 013700 177640
629 165756 013701 177642
630 165762 000164 000002
631 165766 015
632 165767 012
633 165770 000240
634 165772 000240
635 165774 042060
636 165776 113422
637
638 165000
    
```

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

```

PRINT: CLR @#SWR ;LIGHTS = 0, ALL TESTS PASSED
MOV #10, R5 ;PRINT 8. DIGITS
1$: MOV #3, R3 ;SHIFT COUNT FOR DIGIT ASSEMBLY
CLR R2 ;CLR DIGIT ASSEMBLY AREA
2$: ASHC #1, R0 ;PUT BIT IN C-BIT
ROL R2 ;MOVE C-BIT TO R2
SOB R3, 2$ ;BUILD A DIGIT IN R2
ADD #'0, R2 ;CNVRT TO ASCII
3$: TSTB @#XCSR ;PRINTER READY?
BPL 3$
MOVB R2, @#XBUF ;PRINT IT
SOB R5, 1$ ;PRINT 8. DIGITS
    
```

////////////////////////////////////

: EXIT TO ODT OR TO BOOT ROM

////////////////////////////////////

```

MOV @#UPAR0, R0 ;RESTORE BOOT ROM PARAMETERS
MOV @#UPAR1, R1
CRLF: JMP 2(R4) ;RETURN TO BOOT ROM OR ODT
.BYTE 15
.BYTE 12
NOP
NOP
CRC: .ASCII '0D' ;#4 CPU ROM FOR M9312
.WORD 113422 ;CRC CHECKWORD FOR ROM
.END START
    
```


SLASH	362#	622	626							
STARS	362#	395	399	513	517	530	579	583	603	607
.HEADE	362#	365								

. ABS. 166000 000

ERRORS DETECTED: 0

CJM9AA.BIN,CJM9AA.LST/CRF/NL:TOC=CJM9AA.P11
RUN-TIME: 1 1 .1 SECONDS
RUN-TIME RATIO: 44/2=15.2
CORE USED: 7K (14 PAGES)