

IDENTIFICATION

PRODUCT CODE:	MAINDEC-8E-D11B-D
PRODUCT NAME:	M18-E BOOTSTRAP DIAGNOSTIC
DATE CREATED:	JAN. 17, 1972
MAINTAINER:	DIAGNOSTIC PROGRAMMING GROUP
AUTHOR:	JOHN VROBEL

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

THE M18-E BOOTSTRAP DIAGNOSTIC VERIFIES CORRECT OPERATION OF THE M18-E BOOTSTRAP LOADER OPTION IN ALL ITS STANDARD CONFIGURATIONS. THE DIAGNOSTIC PRODUCES A VISUAL TYPE OUT AND/OR A BINARY OBJECT TAPE OF THE BOOTSTRAP BLOCK OF DATA INFORMATION LOADED INTO CORE BY THE M18-E MODULE UNDER TEST. THIS VISUAL TYPEOUT AND BINARY OBJECT TAPE CAN THEN BE SAVED FOR THE TESTING OF M18-E MODULES OF THE SAME CONFIGURATION.

THE DIAGNOSTIC IS AVAILABLE IN A LOW AND HIGH CORE VERSION. THE VERSION TO BE USED TO TEST A M18-E MODULE WILL DEPEND ON THE MEMORY LOCATIONS UTILIZED BY THAT PARTICULAR MODULE. THE LOW CORE VERSION OF THE DIAGNOSTIC OCCUPIES AND USES MEMORY LOCATIONS 0200-1777 AND THE HIGH CORE VERSION OCCUPIES AND USES MEMORY LOCATIONS 4200-5777. USE THE VERSION THAT DOES NOT CONFLICT WITH THE MEMORY LOCATIONS OF THE BOOTSTRAP BLOCK FOR THE M18-E MODULE UNDER TEST.

2. REQUIREMENTS

PDP8/E COMPUTER
ASR-33 TELETYPE OR EQUIVALENT,
LOW OR HIGH SPEED PAPER TAPE READER,
LOW OR HIGH SPEED PAPER TAPE PUNCH,
M18-E BOOTSTRAP DIAGNOSTIC,
M18-E BOOTSTRAP LOADER OPTION.

3. STARTING ADDRESS

THE STARTING ADDRESS OF THE LOW CORE VERSION IS 0200.
THE STARTING ADDRESS OF THE HIGH CORE VERSION IS 4200.

4. PRELIMINARY PROGRAMS

ALL OTHER DIAGNOSTICS FOR THE COMPUTER AND PERIPHERALS SHOULD BE RUN SUCCESSFULLY.

5. OPERATION SWITCH SETTINGS

SWR0=1 VERIFICATION BY BINARY OBJECT TAPE,
SWR0=0 VERIFICATION BY VISUAL TYPEOUT,
SWR1=1 PUNCH BINARY OBJECT TAPE,
SWR2=1 LOW SPEED PAPER TAPE PUNCH,
SWR2=0 HIGH SPEED PAPER TAPE PUNCH,
SWR6=8 MEMORY FIELD OF BINARY LOADER,
SWR9-11 AMOUNT OF EXTENDED MEMORY FIELDS.

6. OPERATOR AND PROGRAM ACTION

- A. INSTALL THE M18-E MODULE TO BE TESTED.
- B. LOAD THE DIAGNOSTIC INTO THE SAME MEMORY FIELD AS UTILIZED BY THE M18-E MODULE UNDER TEST USING THE STANDARD BINARY LOADER TECHNIQUE.
- C. IF THE OPERATOR WISHES TO TEST THE MODULE USING ITS BINARY OBJECT TAPE, LOAD THE BINARY OBJECT TAPE INTO THE SAME MEMORY FIELD AS OCCUPIED BY THE DIAGNOSTIC USING THE

STANDARD BINARY LOADER TECHNIQUE.

- D. DISABLE THE I/O DEVICE USED BY THE MODULE UNDER TEST, FOR EXAMPLE, PLACE NO TAPE IN READER, TURN OFF READER OR PUNCH, OR DISCONNECT THE M8350 TO THE DEVICE.
- E. SET THE SWITCH REGISTER TO THE STARTING ADDRESS OF THE DIAGNOSTIC 0200/4200 AND PRESS ADDRESS LOAD.
- F. SET THE SWITCH REGISTER TO THE INITIAL ADDRESS OF THE BOOTSTRAP DATA BLOCK OF INFORMATION OF THE PARTICULAR MODULE UNDER TEST AND PRESS CLEAR AND THEN CONTINUE, THE COMPUTER SHOULD HALT AT ADDRESS 0202/4202.
- G. SET THE SWITCH REGISTER TO THE STARTUP ADDRESS OF THE MODULE UNDER TEST AND PRESS CLEAR AND THEN CONTINUE, THE COMPUTER SHOULD HALT AT ADDRESS 0205/4205.
- H. IF THE OPERATOR HAS SELECTED TO TEST THE MODULE USING THE BINARY OBJECT TAPE, SET SWR0=1. IF VERIFICATION IS DESIRED BY VISUAL TYPEOUT, SET SWR0=0.
- I. IF THE OPERATOR WISHES TO PUNCH A NEW BINARY OBJECT TAPE, SET SWR1=1 AND SWR2=1 FOR LOW SPEED PUNCH OR SWR2=0 FOR HIGH SPEED PUNCH.
- J. SET SWR6=0 TO THE MEMORY FIELD OF THE BINARY LOADER AND SWR9=11 TO THE AMOUNT OF EXTENDED MEMORY FIELDS AND PRESS CLEAR AND THEN CONTINUE.
- K. THE BINARY LOADER WILL BE RELOCATED FROM THE FIELD SPECIFIED IN SWR6=0 TO A BUFFER AREA WITHIN THE DIAGNOSTIC; THE SWITCH REGISTER SETTINGS FOR THE STARTUP ADDRESS AND THE INITIAL ADDRESS OF THE BOOTSTRAP INFORMATION WILL BE CHECKED TO MAKE SURE THEY DO NOT CONFLICT WITH THE DIAGNOSTIC.
- L. THE DIAGNOSTIC WILL NOW LOAD A DATA PATTERN OF 2525 INTO ALL MEMORY LOCATIONS NOT OCCUPIED BY THE DIAGNOSTIC AND THEN SIGNAL THE OPERATOR WITH A BELL ON THE TTY TO TOGGLE THE BOOTSTRAP SWITCH AS THE DATA PATTERN IS BEING CHECKED, THIS WILL VERIFY THAT THE COMPUTER IS NOT EFFECTED BY THE MODULE WHILE THE COMPUTER IS IN THE RUN STATE, THE OPERATOR MUST TOGGLE THE BOOTSTRAP SWITCH AT LEAST 10 TIMES DURING THIS TEST AND THEN HIT A KEY ON THE TTY TO EXIT TO NEXT TEST.
- M. THE DIAGNOSTIC WILL NOW LOAD A DATA PATTERN OF 5252 INTO ALL MEMORY LOCATIONS NOT OCCUPIED BY THE DIAGNOSTIC AND THEN SIGNAL THE OPERATOR WITH A BELL ON THE TTY TO TOGGLE THE BOOTSTRAP SWITCH AS THE DATA PATTERN IS BEING CHECKED, THIS WILL VERIFY THAT THE COMPUTER IS NOT EFFECTED BY THE MODULE WHILE THE COMPUTER IS IN THE RUN STATE, THE OPERATOR MUST TOGGLE THE BOOTSTRAP SWITCH AT LEAST 10 TIMES DURING AND THEN HIT A KEY ON THE TTY TO EXIT TO NEXT TEST.
- N. THE DIAGNOSTIC WILL NOW LOAD A DATA PATTERN OF 2525 IN ALL MEMORY LOCATIONS NOT OCCUPIED BY THE DIAGNOSTIC, LOAD A HALT IN THE BOOTSTRAP BLOCK +1 JUST IN CASE THE BOOTSTRAP DATA DOESN'T HANG, THEN HALT IN ADDRESS 1640/5640, THE OPERATOR MUST TOGGLE THE BOOTSTRAP SWITCH ONCE THEN ST

INCORRECT SWITCH SETTINGS FOR THE STARTUP ADDRESS OF THE MODULE AND THE INITIAL ADDRESS OF THE BOOTSTRAP DATA BLOCK OF INFORMATION WILL RESULT IN A HALT AT ADDRESS 0545/4545, THE OPERATOR MAY RE-SET THE SWITCH REGISTER FOR THE INITIAL ADDRESS OF THE DATA BLOCK OF INFORMATION AND HIT CONTINUE TO RE-START THE SETUP PROCEDURE.

8. RESTRICTIONS

THE OPERATOR MUST NOTE THAT ENCODED BOOTSTRAPS OF STANDARD CONFIGURATIONS SHOULD HANG WHEN LOADED AND AUTOMATICALLY STARTED-UP. UNENCODED BOOTSTRAPS SHOULD NOT HANG BUT SHOULD RESULT IN A HALT AT THE BOOTSTRAP BLOCK #1.

THE I/O DEVICE THAT THE BOOTSTRAP USES MUST BE DISABLED.

THE INITIAL ADDRESS OF THE BOOTSTRAP BLOCK AND THE STARTUP ADDRESS OF THE MODULE MUST NOT CONFLICT WITH THE DIAGNOSTIC.

THE STARTUP ADDRESS OF THE MODULE MUST BE WITHIN THE BOOTSTRAP BLOCK OF INFORMATION.

THE BINARY OBJECT TAPE USED TO TEST A PARTICULAR MODULE MUST BE USED WITH THE SAME (LOW OR HIGH) VERSION OF THE DIAGNOSTIC FROM WHICH IT WAS PUNCHED.

THE DIAGNOSTIC AND BINARY OBJECT TAPE MUST BE LOADED INTO THE SAME MEMORY FIELD AS UTILIZED BY THE MODULE UNDER TEST.

THE OPERATOR MUST NOTE THAT THE DOWNWARD MOTION OF THE BOOTSTRAP SWITCH DOES NOT IN ANYWAY EFFECT THE COMPUTER. THE BOOTSTRAP SHOULD ONLY LOAD ON THE UPWARD MOTION OF THE BOOTSTRAP SWITCH.

9. GENERAL INFORMATION

THE FIRST TIME AN OPERATOR RUNS THIS DIAGNOSTIC TO TEST A PARTICULAR TYPE OF MODULE HE MUST VERIFY THE MODULE BY THE VISUAL TYPEOUT METHOD. IF THIS INFORMATION IS CORRECT, HE SHOULD THEN PUNCH A BINARY OBJECT TAPE OF THE INFORMATION LOADED BY THE MODULE. THE OPERATOR SHOULD THEN LABEL AND SAVE THIS BINARY OBJECT TAPE AND VISUAL TYPEOUT FOR THE TESTING OF MODULES OF THE SAME CONFIGURATION.

THE STARTUP ADDRESS OF THE MODULE REFERED TO IN THIS DOCUMENTATION IS THE ADDRESS AT WHICH THE BOOTSTRAP PROGRAM WILL AUTOMATICALLY START AFTER BEING LOADED BY THE MODULE. THE INITIAL ADDRESS OF THE DATA BLOCK OF INFORMATION IS THE FIRST ADDRESS LOCATION INTO WHICH THE FIRST OF THE 32 DATA WORDS WILL BE LOADED.

THE BOOTSTRAP SWITCH IS LOCATED TO THE LEFT OF THE SWITCH REGISTER ON THE PDP8/E FRONT PANEL AND IS LABELED "SW".

THE 32 DECIMAL WORDS OF DATA INFORMATION LOADED BY THE BOOTSTRAP MODULE IS DEPENDENT ON THE DIODES LABELED "WORD 1-32" ON THE M18-E MODULE. CUT DIODES RESULT IN A DATA 1 AND UNCUT DIODES RESULT IN A DATA 0.

ENCODED BOOTSTRAP MODULES ARE THOSE WHOSE "WORD DIODES" ARE CUT FOR CERTAIN DATA PATTERNS AND INSTRUCTIONS.

AND/OR RE-START THE COMPUTER AT ADDRESS 0266/4266, THE DIAGNOSTIC WILL CHECK ALL MEMORY LOCATIONS NOT OCCUPIED BY THE DIAGNOSTIC AND THE BOOTSTRAP BLOCK FOR A CORRECT DATA PATTERN AND THEN CHECK THE HALT LOADED INTO THE BOOTSTRAP BLOCK +1. IF VISUAL TYPEOUT WAS PREVIOUSLY SELECTED THE BOOTSTRAP BLOCK OF INFORMATION LOADED BY THE MODULE WILL BE TYPED OUT ON THE TTY AND THE OPERATOR MUST VERIFY THAT THIS IS CORRECT. IF VERIFICATION WAS PREVIOUSLY SELECTED BY THE BINARY OBJECT TAPE THE DIAGNOSTIC WILL COMPARE THE BOOTSTRAP INFORMATION LOADED BY THE MODULE TO THAT INFORMATION LOADED BY THE BINARY OBJECT TAPE.

- O. IF THE OPERATOR HAS SELECTED TO PUNCH A BINARY OBJECT TAPE THE COMPUTER WILL HALT AT ADDRESS 0307/4307 TO ALLOW THE OPERATOR TO PREPARE THE PUNCH. THE OPERATOR MUST THEN HIT CONTINUE TO PUNCH THE BINARY OBJECT TAPE.
- P. THE DIAGNOSTIC WILL NOW LOAD A DATA PATTERN OF 5252 INTO ALL MEMORY LOCATIONS NOT OCCUPIED BY THE DIAGNOSTIC AND THEN HALT AT ADDRESS 1640/5640. THE OPERATOR MUST TOGGLE THE BOOTSTRAP SWITCH ONCE THEN STOP AND/OR RESTART THE DIAGNOSTIC AT ADDRESS 0323/4323. THE DIAGNOSTIC WILL CHECK THE DATA PATTERN IN ALL MEMORY LOCATIONS NOT OCCUPIED BY THE DIAGNOSTIC AND THE BOOTSTRAP BLOCK, CHECK THE HALT IN THE BOOTSTRAP BLOCK +1, AND COMPARE THE BOOTSTRAP BLOCK OF INFORMATION TO THAT FOUND IN STEP N. THE COMPUTER SHOULD THEN HALT AT ADDRESS 0332/4332 INDICATING A SUCCESSFUL PASS COMPLETE. IF THE OPERATOR HITS CONTINUE THE DIAGNOSTIC SHOULD ENTER THE FIRST TEST SECTION L. IF THE OPERATOR WISHES TO TEST A MODULE OF THE SAME CONFIGURATION, THE DIAGNOSTIC CAN BE RESTARTED FROM ADDRESS 0333/4333, THUS ELIMINATING THE INITIAL SETUP PROCEDURE.

7. ERRORS

A NON-RECOVERABLE ERROR MAY OCCUR IF THE BOOTSTRAP BLOCK OF INFORMATION DESTROYS THE DIAGNOSTIC IN CORE. IF THIS SHOULD OCCUR, IT IS POSSIBLE TO SINGLE STEP THE ACTUAL LOAD OF THE BOOTSTRAP MODULE.

IF A RECOVERABLE DATA ERROR DOES OCCUR THE COMPUTER SHOULD HALT WITH THE DEFECTIVE ADDRESS IN THE MD, THE DEFECTIVE DATA IN THE AC, AND THE MEMORY FIELD WHERE THE VALUES WERE FOUND IN THE DF INDICATORS. THE OPERATOR MAY HIT CONTINUE TO DISPLAY THE VALUE EXPECTED IN THE AC.

THE FOLLOWING MEMORY ADDRESSES LISTED BELOW ARE RECOVERABLE DATA ERROR HALTS AND FAILURES, FOR MORE INFORMATION REFERENCE THE DIAGNOSTIC LISTING AND/OR SECTION 6 OF THE DOCUMENT.

- | | |
|-----------|--|
| 0753/4753 | DATA PATTERN OF 2525 OR 5252 LOADED INTO CORE BY THE DIAGNOSTIC WAS INCORRECT. |
| 1036/5036 | HALT LOADED INTO THE BOOTSTRAP BLOCK +1 BY THE DIAGNOSTIC WAS INCORRECT. |
| 1072/5072 | THE BOOTSTRAP INFORMATION LOADED INTO CORE BY THE MODULE UNDER TEST WAS INCORRECT. |

UNENCODED BOOTSTRAP MODULES ARE THOSE WHOSE "WORD DICODES"
ARE NOT CUT RESULTING IN AN ALL 0'S PATTERN.

THE MEMORY FIELD, STARTUP ADDRESS, AND THE INITIAL ADDRESS
OF THE BOOTSTRAP INFORMATION IS DEPENDENT ON THE SPLIT
LUG JUMPERS LOCATED ON THE M18-E MODULE AND LABELED
F2-F2, S0-S11, AND I2-I11 RESPECTIVELY.

THE BINARY LOADER MAY BE REPLACED AFTER RUNNING THE DIAGNOSTIC
TO ITS ORIGINAL LOCATIONS IN MEMORY BY LOAD AND STARTING
ADDRESS 1200/5202.

AN EXAMPLE OF THE BOOTSTRAP DATA TYPEOUT IS SHOWN BELOW.

M18-E BOOTSTRAP DATA	
ADRS	DATA
0023	6007
0024	6751
0025	6745
0026	5025
0027	7200
0030	6733
0031	5031
0032	7777
0033	7777
0034	7777
0035	7777
0036	7777
0037	7777
0040	7777
0041	7777
0042	7777
0043	7777
0044	7777
0045	7777
0046	7777
0047	7777
0050	7777
0051	7777
0052	7777
0053	7777
0054	7777
0055	7777
0056	7777
0057	7777
0060	7777
0061	7777
0062	7777

10. LISTING

```

/M18-E BOOTSTRAP DIAGNOSTIC
/COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION
6887 CAP=6887
7421 MQL=7421
/
/ THE LOW VERSION STARTING ADDRESS IS 0200,
/ THE HIGH VERSION STARTING ADDRESS IS 4200,
/
/ OPERATION SWITCH REGISTER FOR M18-E DIAGNOSTIC
/
/ SWR0=0 VERIFICATION BY OCTAL DUMP TYPE OUT
/ SWR0=1 VERIFICATION BY BINARY OBJECT TAPE
/ SWR1=1 PUNCH BINARY OBJECT TAPE
/ SWR2=1 LOW SPEED PUNCH
/ SWR2=0 HIGH SPEED PUNCH
/ SWR6=0 FIELD OF BINARY LOADER
/ SWR9=11 AMOUNT OF EXTENDED FIELDS
/
/ ROUTINE TO SAVE THE INITIAL BLOCK ADDRESS
/ AND THE STARTUP ADDRESS OF THE BOOTSTRAP,
/ SAVE BINARY LOADER IN PROGRAM BUFFER AREA,
/ CHECK TO MAKE SURE THAT ADDRESSES DO NOT INTERFERE
/ WITH THE LOCATIONS OF THE DIAGNOSTIC,
/ ALSO SAVE OPERATION SWITCHES,
/
0200 *0200
/
0200 7604 BEGIN, LAS
0201 3335 DCA STRBLK /SAVE START OF BOOTSTRAP BLOCK
0202 7402 HLT
0203 7604 LAS
0204 3336 DCA STRADD /SAVE STARTUP ADDRESS OF BOOTSTRAP
0205 7402 HLT
0206 7604 LAS
0207 3337 DCA EXTSAV /SAVE OPERATING SWITCHES
0210 6224 RIF
0211 1342 TAD KCDF /MAKE PRESENT FIELD CDF
0212 3213 DCA PREFLD
0213 0000 PREFLD, 0
0214 1335 TAD STRBLK /GET START OF BLOCK
0215 1340 TAD AMOUNT /GET LENGTH OF BLOCK *I
0216 3341 DCA HLTLOC /MAKE HALT LOCATION
0217 1344 TAD FIRPAS /GET PASS FLAG
0220 7600 SZA CLA /IS IT FIRST PASS
0221 4777 JMS MOVBIN /YES, MOVE THE BINARY LOADER
0222 4776 JMS CHKADD /CHECK THAT SWITCHES DO NOT CONFLICT
/
/LOAD MEMORY WITH DATA PATTERN 2525 AND SIGNAL
/OPERATOR TO TOGGLE SWITCH AS MEMORY IS BEING CHECKED,
/ SWITCH SHOULD BE TOGGLED AT LEAST 10 TIMES
/OPERATOR MUST HIT TTY KEY TO EXIT TO NEXT TEST,
/ THIS WILL VERIFY THAT THE PROCESSOR AND THE MEMORY

```

```

/ WILL NOT BE AFFECTED BY BOOTSTRAP WHEN THE COMPUTER
/ IS RUNNING,
/
0223 6887 TEST1, CAF /CLEAR THE WORLD
0224 3344 DCA FIRPAS
0225 1746 TAD I XLOD /GET JMS FOR LOAD
0226 3775 DCA LODCHK /SETUP FOR LOAD
0227 4774 JMS MEMGO /SETUP FOR LOAD MEMORY
0230 2525 2525 /LOAD MEMORY WITH DATA PATTERN
0231 4773 JMS BELL /DATA PATTERN TO BE USED
0232 1745 TAD I XCHK /SIGNAL OPERATOR
0233 3775 DCA LODCHK /GET JMS FOR CHECK
0234 4774 JMS MEMGO /SETUP FOR CHECK MEMORY
0235 2525 2525 /CHECK MEMORY
0236 6831 KSF /COMPARE TO THIS PATTERN
0237 5234 JMP .=3 /WAIT FOR OPERATOR TO CONTINUE
/
/LOAD MEMORY WITH DATA PATTERN 5252 AND SIGNAL
/OPERATOR TO TOGGLE SWITCH AS MEMORY IS BEING CHECKED,
/ SWITCH SHOULD BE TOGGLED AT LEAST 10 TIMES
/OPERATOR MUST HIT TTY KEY TO EXIT THIS TEST
/ THIS WILL VERIFY THAT THE PROCESSOR AND THE MEMORY
/ WILL NOT BE AFFECTED BY BOOTSTRAP WHEN THE COMPUTER
/ IS RUNNING,
/
0240 7300 TEST2, CLA CLL
0241 1746 TAD I XLOD /GET JMS FOR LOAD
0242 3775 DCA LODCHK /SETUP FOR LOAD MEMORY
0243 4774 JMS MEMGO /SETUP FOR LOAD MEMORY
0244 5252 5252 /LOAD MEMORY WITH DATA PATTERN
0245 4773 JMS BELL /DATA PATTERN TO BE USED
0246 1745 TAD I XCHK /SIGNAL OPERATOR
0247 3775 DCA LODCHK /SETUP FOR CHECK MEMORY
0250 4774 JMS MEMGO /CHECK MEMORY
0251 5252 5252 /COMPARE TO THIS PATTERN
0252 6831 KSF
0253 5250 JMP .=3 /WAIT FOR OPERATOR TO CONTINUE
/
/LOAD MEMORY WITH DATA PATTERN 2525,
/LOAD A HALT INTO BOOTSTRAP BUFFER *1 JUST
/ INCASE THE BOOTSTRAP DOESN'T HANG,
/ THEN GO HALT AND WAIT FOR THE OPERATOR TO
/ TOGGLE THE BOOTSTRAP SWITCH ONCE,
/ THE OPERATOR MUST THEN RESTART THE PROGRAM AT
/ LOCATION 0200/4200,
/
0254 7300 TEST3, CLA CLL
0255 1746 TAD I XLOD /GET JMS FOR LOAD
0256 3775 DCA LODCHK /SETUP FOR LOAD
0257 4774 JMS MEMGO /SETUP FOR LOAD MEMORY
0260 2525 2525 /LOAD MEMORY WITH DATA PATTERN
0261 1343 TAD KHLT
0262 3734 DCA I WATHLT /STORE WAIT HALT AT END OF DIAG.
0263 1343 TAD KHLT
0264 3741 DCA I HLTLOC /STORE HLT IN BOOTSTRAP BLOCK *1

```

0265 5734

JMP I WATHLT

/GO HALT AND WAIT FOR OPERATOR TO
/TOGGLE SWITCH AND RESTART PROGRAM/CHECK MEMORY NOT OCCUPIED BY THE BOOTSTRAP
/FOR CORRECT PATTERN OF 2925, IF OPERATOR HAS
/INHIBITED VISUAL PRINT OUT ASSUME THAT THE
/BINARY OBJECT TAPE OF THE BOOTSTRAP INFORMATION
/HAS BEEN LOADED INTO CORE AND COMPARE THIS OBJECT
/TO THE INFORMATION BOOTSTRAPED INTO CORE,
/IF VISUAL CHECK PRINT OUT BOOTSTRAP BUFFER INFORMATION,
/IF OPERATOR HAS SELECTED TO PUNCH NEW OBJECT TAPE
/HALT AND WAIT FOR OPERATOR TO PREPARE THE
/PAPER TAPE PUNCH SELECTED,
/OPERATOR MUST HIT CONTINUE TO PUNCH TAPE,

0266	7300	RESTR3, CLA CLL	
0267	1337	TAD EXTSAV	/GET OPERATION SWITCHES
0270	7710	SPA CLA	/SHR0=1 IS INHIBIT OCTAL DUMP
0271	5274	JMP INHDP	/INHIBIT OCTAL DUMP OF BOOTSTRAP
0272	4772	JMS MOVBUF	/MOVE BOOTSTRAP TO BUFFER AREA
0273	4771	JMS TYPBUF	/TYPE OCTAL DUMP OF BOOTSTRAP
0274	1745	INHDP, TAD I XCHK	/GET JMS FOR CHECK
0275	3775	DCA LODCHK	/SETUP FOR CHECK MEMORY
0276	7320	CLA CLL CML	
0277	4774	JMS MEMGO	
0300	2525	2525	/CHECK MEMORY OTHER THAN BOOTSTRAP
0301	4770	JMS CHKHLT	/CHECK HLT STORED IN BLOCK +1
0302	4767	JMS COMPAR	/COMPARE BOOTSTRAP TO BUFFER AREA
0303	1337	TAD EXTSAV	/GET OPERATION SWITCHES
0304	7004	RAL	
0305	7700	SM CLA	/SHR1=1 IS PUNCH NEW TAPE
0306	5311	JMP TEST4	/INHIBIT TAPE AND GO TO NEXT TEST
0307	7402	HLT	/WAIT FOR OPERATOR TO PREPARE PUNCH
0310	4766	JMS BPUN	/OPERATOR MUST HIT CONTINUE TO PUNCH

/LOAD MEMORY WITH DATA PATTERN 5252,
/LOAD A HALT INTO BOOTSTRAP BUFFER +1 JUST
/IN CASE THE BOOTSTRAP DOESN'T HANG,
/THEN GO HALT AND WAIT FOR THE OPERATOR TO
/TOGGLE THE BOOTSTRAP SWITCH ONCE,
/THE OPERATOR MUST THEN RESTART THE PROGRAM AT
/LOCATION 0323/4323.

0311	7300	TEST4, CLA CLL	
0312	1746	TAD I XLOD	/GET JMS FOR LOAD
0313	3775	DCA LODCHK	/SETUP FOR LOAD
0314	4774	JMS MEMGO	/LOAD MEMORY WITH DATA PATTERN
0315	5252	5252	
0316	1343	TAD KHLT	
0317	3734	DCA I WATHLT	/STORE WAIT HALT AT END OF DIAG.
0320	1343	TAD KHLT	
0321	3741	DCA I HLTLOC	/STORE A HLT IN BLOCK +1
0322	5734	JMP I WATHLT	/GO HALT AND WAIT FOR OPERATOR TO /TOGGLE SWITCH AND RESTART PROGRAM

/CHECK MEMORY NOT OCCUPIED BY THE BOOTSTRAP
/FOR A CORRECT PATTERN OF 9252, THEN COMPARE
/THE BOOTSTRAP INFORMATION TO THAT FOUND IN
/TEST 3.

0323	7300	RESTR4, CLA CLL	
0324	1745	TAD I XCHK	/GET JMS FOR CHECK
0325	3775	DCA LODCHK	/SETUP FOR CHECK MEMORY
0326	7320	CLA CLL CML	
0327	4774	JMS MEMGO	/CHECK MEMORY OTHER THAN BOOTSTRAP
0330	5252	5252	
0331	4770	JMS CHKHLT	/CHECK HLT IN BLOCK +1
0332	7402	HLT	/END OF TEST
0333	5223	JMP TEST1	/LOOP ON PROGRAM

0334	1640	WATHLT, INBUF +40	
0335	0000	STRBLK, 0	
0336	0000	STRADD, 0	
0337	0000	EXTSAV, 0	
0340	0040	AMOUNT, 0040	
0341	0000	HLTLOC, 0	
0342	0201	KCDF, CDF	
0343	7402	KHLT, HLT	
0344	7777	FIRPAS, 7777	
0345	0760	XCHK, KCHK	
0346	0761	XLOD, KLOD	

0366	1223		
0367	1043		
0370	1023		
0371	0421		
0372	0400		
0373	0514		
0374	0600		
0375	0743		
0376	0520		
0377	1000		
	0400		

PAGE

/ROUTINE TO MOVE BOOTSTRAP INFORMATION FROM
/BOOTSTRAP AREA TO PROGRAM BUFFER AREA.

0400	0000	MOVBUF, 0	
0401	1777	TAD STRBLK	
0402	3364	DCA BCNT6	
0403	1776	TAD AMOUNT	
0404	7041	CIA	
0405	3365	DCA BCNT7	
0406	1302	TAD XINBUF	
0407	3366	DCA BCNT8	
0410	1764	TAD I BCNT6	
0411	3766	DCA I BCNT8	
0412	2364	ISE BCNT6	
0413	7000	NOP	


```

0414 2366      ISE BCNT8
0415 7000      NOP
0416 2365      ISE BCNT7
0417 5210      JMP MOVBUF +10
0420 5600      JMP I MOVBUF

/ROUTINE TO TYPE BOOTSTRAP DATA INFORMATION
/
0421 0000      TYPBUF, 0
0422 1360      TAD TEXTMS
0423 3364      DCA BCNT6
0424 1361      TAD TEXTLG
0425 3365      DCA BCNT7
0426 1764      TAD I BCNT6
0427 4305      JMS TYPE
0430 2364      ISE BCNT6
0431 7000      NOP
0432 2365      ISE BCNT7
0433 5226      JMP TYPBUF +5
0434 1777      TAD STRBLK
0435 3364      DCA BCNT6
0436 1776      TAD AMOUNT
0437 7041      CIA
0440 3365      DCA BCNT7
0441 1364      STRTYP, TAD BCNT6
0442 4264      JMS OCTEL
0443 1347      TAD K7774
0444 3366      DCA BCNT8
0445 1354      TAD K0240
0446 4305      JMS TYPE
0447 2366      ISE BCNT8
0450 5245      JMP ,=3
0451 1764      TAD I BCNT6
0452 4264      JMS OCTEL
0453 1350      TAD K0215
0454 4305      JMS TYPE
0455 1351      TAD K0212
0456 4305      JMS TYPE
0457 2364      ISE BCNT6
0460 7000      NOP
0461 2365      ISE BCNT7
0462 5241      JMP STRTYP
0463 5621      JMP I TYPBUF

```

/MAKE A SPACE

```

/ROUTINE TO TYPE OCTAL INFORMATION,
/
0464 0000      OCTEL, 0
0465 7106      RTL CLL
0466 7006      RTL
0467 3363      DCA ACSAV1
0470 1347      TAD K7774
0471 3367      DCA BCNT9
0472 1363      TAD ACSAV1
0473 0775      AND K0207
0474 1353      TAD K0260

```

```

0475 4305      JMS TYPE
0476 1363      TAD ACSAV1
0477 7006      RTL
0500 7004      RAL
0501 3363      DCA ACSAV1
0502 2367      ISE BCNT9
0503 5272      JMP ,=11
0504 5664      JMP I OCTEL

/TYPE, 0
0505 0000      TYPE, 0
0506 6046      TLS
0507 6041      TSP
0510 5307      JMP ,=1
0511 6042      YCF
0512 6032      KCC
0513 5705      JMP I TYPE

/BELL, 0
0514 0000      BELL, 0
0515 1352      TAD K0207
0516 4305      JMS TYPE
0517 5714      JMP I BELL

/ROUTINE TO CHECK THAT ADDRESSES SUBMITTED BY OPERATOR
/DO NOT CONFLICT WITH DIAGNOSTIC, IF SWITCH ERROR OCCURES
/THE COMPUTER SHOULD HALT, RE-SET SWITCH FOR STARTING
/ADDRESS OF BLOCK AND HIT CONTINUE TO TRY AGAIN.
/
0520 0000      CHKADD, 0
0521 7300      CLA CLL
0522 1356      TAD SAFADD
0523 7040      CMA
0524 1777      TAD STRBLK
0525 7630      SEL CLA
0526 5334      JMP STRUP
0527 1397      TAD XBEGIN
0530 7041      CIA
0531 1774      TAD HLTLOC
0532 7630      SEL CLA
0533 9345      JMP ADDHLT
0534 1777      STRUP, TAD STRBLK
0535 7041      CIA
0536 1773      TAD STRADD
0537 7510      SPA
0540 9345      JMP ADDHLT
0541 7181      CIA STL
0542 1355      TAD LENGTH
0543 7620      SNL CLA
0544 5720      JMP I CHKADD
0545 7602      ADDHLT, HLT CLA
0546 5772      JMP BEGIN

/
0547 7774      K7774, 7774
0550 0215      K0215, 0215
0551 0212      K0212, 0212

```

0552 0207
 0553 0260
 0554 0240
 0555 0037
 0556 1777
 0557 0177
 0560 1107
 0561 7732
 0562 1600
 0563 0000
 0564 0000
 0565 0000
 0566 0000
 0567 0000
 /
 0572 0200
 0573 0336
 0574 0341
 0575 0765
 0576 0340
 0577 0335
 0600

PAGE
 /
 /ROUTINE TO DETERMINE FIELDS TO BE CHECKED
 /
 MEMGO, 0
 TAD PREFLD /GET PRESENT FIELD CDF
 DCA FLDGO
 SZL CLA
 JMP SPECHK /CHECK FOR ALL BUT PROG, + BOOTSTRAP
 TAD XBEGIN /GET STARTING ADDRESS OF CHECK
 DCA TSTOP
 TAD LASTLC /GET ENDING ADDRESS OF CHECK
 DCA TBEGIN
 JMS FLDCHK /ENTER ROUTINE TO LOAD OR CHECK
 CLA CLL CMA
 DCA TSTOP /START AT 0
 DCA TBEGIN /END AT 0
 EXTCHK, TAD EXTSAV
 AND K0007
 SNA /IS IT TEST EXTENDED MEMORY
 JMP EXIT /NO DO NOT TEST EXTENDED MEMORY
 CMA
 DCA FLDAMN /SETUP FOR FIELDS TO TEST
 STRCHK, DCA BCNT3 /START WITH 0
 TAD BCNT3
 TAD K00F
 DCA FLDGO /FIELD TO BE CHECKED
 RIF
 CIA
 TAD BCNT3
 SEA CLA /ARE WE IN THIS FIELD
 JMS FLDCHK /NO, ENTER ROUTINE TO LOAD OR CHECK
 TAD BCNT3
 TAD K0010

0636 2363
 0637 5223
 0640 7300
 0641 2200
 0642 5600
 /
 0643 0000
 0644 1777
 0645 3262
 0646 1643
 0647 3367
 0650 2243
 0651 1643
 0652 3370
 0653 2243
 0654 1643
 0655 3260
 0656 2243
 0657 1600
 0660 0000
 0661 3767
 0662 0000
 0663 4314
 0664 5257
 0665 5643
 /
 0666 0000
 0667 1777
 0670 3310
 0671 1666
 0672 3367
 0673 2266
 0674 1666
 0675 3370
 0676 2266
 0677 1666
 0700 3303
 0701 2266
 0702 1600
 0703 0000
 0704 7041
 0705 1767
 0706 7640
 0707 5350
 0710 0000
 0711 4314
 0712 5302
 0713 5666

ISE FLDAMN
 JMP STRCHK /MORE TO GO
 CLA CLL
 ISE MEMGO
 JMP I MEMGO /EXIT
 /ROUTINE TO LOAD MEMORY WITH DATA PATTERN
 LODMEM, 0
 TAD PREFLD /SETUP FOR PRESENT FIELD
 DCA TMSFLD
 TAD I LODMEM /GET STARTING ADDRESS
 DCA BCNT4
 ISE LODMEM
 TAD I LODMEM /GET ENDING ADDRESS
 DCA BCNT5
 ISE LODMEM
 TAD I LODMEM /GET FIELD TO GO
 DCA NEWFLD
 ISE LODMEM
 TAD I MEMGO /GET DATA PATTERN
 NEWFLD, 0 /MODIFIED BY TEST
 DCA I BCNT4
 TMSFLD, 0 /MODIFIED BY TEST
 JMS ENDTST /IS IT END OF TEST
 JMP NEWFLD -1 /NO, CONTINUE
 JMP I LODMEM /YES, EXIT
 /ROUTINE TO CHECK MEMORY FOR CORRECT DATA PATTERN,
 CHKMEM, 0
 TAD PREFLD /GET PRESENT FIELD
 DCA CHKTHS
 TAD I CHKMEM /GET STARTING ADDRESS
 DCA BCNT4
 ISE CHKMEM
 TAD I CHKMEM /GET ENDING ADDRESS
 DCA BCNT5
 ISE CHKMEM
 TAD I CHKMEM /GET FIELD TO TEST
 DCA FRMFLD
 ISE CHKMEM
 TAD I MEMGO /GET EXPECTED DATA
 FRMFLD, 0
 CIA
 TAD I BCNT4 /GET DATA PATTERN
 SEA CLA /ARE THEY THE SAME
 JMP ACERR1 /NO, INDICATE
 CHKTHS, 0 /MODIFIED BY TEST
 JMS ENDTST /IS IT END OF TEST
 JMP FRMFLD -1 /NO, CONTINUE
 JMP I CHKMEM /YES EXIT
 /ROUTINE TO CHECK FOR END OF TEST

```

0714 0000  ENDTST, 0
0715 1367  TAD BCNT4 /GET ENDING ADDRESS
0716 7041  CIA
0717 1370  TAD BCNT5 /GET PRESENT ADDRESS
0720 7650  SNA CLA /IS IT LAST ADDRESS TO TEST
0721 2314  ISE ENDTST /YES, EXIT
0722 2367  ISE BCNT4
0723 5714  JMP I ENDTST /EXIT
0724 5714  JMP I ENDTST /EXIT

/Routine to test all but prog. + bootstrap
0725 1362  SPECHK, TAD LASTLC /GET START OF CHECK
0726 3344  DCA TBEGIN
0727 7340  CLA CLL CMA
0730 1773  TAD STRBLK
0731 3345  DCA TSTOP /GET END OF CHECK
0732 4342  JMS FLDCHK
0733 7301  CLA CLL IAC /ENTER ROUTINE TO LOAD OR CHECK
0734 1772  TAD HLTLOC
0735 3344  DCA TBEGIN
0736 1774  TAD XBEGIN
0737 3345  DCA TSTOP /GET END OF CHECK
0740 4342  JMS FLDCHK /ENTER ROUTINE TO LOAD OR CHECK
0741 5215  JMP EXTCHK /EXIT

/Routine to load or check memory
0742 0000  FLDCHK, 0
0743 0000  LODCHK, 0 /MODIFIED BY TEST
0744 0000  TBEGIN, 0 /MODIFIED BY TEST
0745 0000  TSTOP, 0 /MODIFIED BY TEST
0746 0000  FLDGO, 0 /MODIFIED BY TEST
0747 5742  JMP I FLDCHK /EXIT

/Routine to display core pattern errors,
/IF AN ERROR OCCURS THE MACHINE WILL
/HLT WITH BAD ADDRESS IN HQ AND BAD DATA IN AC,
/HIT CONTINUE,
/THE MACHINE WILL HLT WITH EXPECTED DATA IN AC,
0750 1367  ACERR1, TAD BCNT4 /GET BAD ADDRESS
0751 7421  HQL /LOAD AC TO HQ
0752 1767  TAD I BCNT4 /GET BAD DATA PATTERN
0753 7402  ERHLT1, HLT /AC = DATA PATTERN FOUND
0754 7300  CLA CLL /AC = EXPECTED DATA PATTERN
0755 1670  TAD I MEMGO /GET GOOD NUMBER
0756 7402  HLT
0757 5356  JMP ,=1

0760 4266  KCHK, JMS CHKHEM
0761 4243  KLOD, JMS LODMEM
0762 1641  LASTLC, INBUF +41
0763 0000  FLDAMN, 0
0764 0010  K0010, 0010
    
```

```

0765 0007  K0007, 0007
0766 0000  BCNT3, 0
0767 0000  BCNT4, 0
0770 0000  BCNT5, 0

0772 0341
0773 0335
0774 0342
0775 0337
0776 0557
0777 0213
1000 PAGE

/Routine to move the binary loader
1000 0000  MOVBIN, 0
1001 1306  TAD BUFBIN
1002 3277  DCA BCNT1 /SETUP BINARY LOADER BUFFER
1003 1777  TAD PREFLD
1004 3215  DCA SETFLD
1005 1305  TAD K7600
1006 3300  DCA BCNT2 /SETUP FOR BINARY LOADER
1007 1774  TAD EXTSAV
1010 0304  AND K0070 /MASK 0-8
1011 1775  TAD KCDF
1012 3213  DCA BINFLD /FIELD OF BINARY LOADER
1013 0000  BINFLD, 0 /MODIFIED BY TEST
1014 1700  TAD I BCNT2 /GET BINARY WORD
1015 0000  SETFLD, 0 /MODIFIED BY TEST
1016 3677  DCA I BCNT1 /STORE IN BUFFER AREA
1017 2277  ISE BCNT1
1020 2300  ISE BCNT2
1021 5213  JMP BINFLD /MORE WORDS TO GO
1022 5600  JMP I MOVBIN /EXIT

/Routine to check halt after bootstrap
1023 0000  CHKHLT, 0
1024 1774  TAD HLTLOC /GET HALT LOCATION
1025 3301  DCA BCNT10
1026 1701  TAD I BCNT10 /GET HALT
1027 7041  CIA
1030 1773  TAD KHLT /GET EXPECTED VALUE
1031 7650  SNA CLA /WERE THEY THE SAME
1032 5623  JMP I CHKHLT /YES EXIT

/Routine to display error for bad hlt location
/IF AN ERROR OCCURS THE MACHINE WILL
/HLT WITH BAD ADDRESS IN HQ AND BAD DATA IN AC,
/HIT CONTINUE,
/THE MACHINE WILL HLT WITH EXPECTED DATA IN AC,
1033 1301  ACERR2, TAD BCNT10 /GET BAD ADDRESS
1034 7421  HQL /LOAD H.Q.
1035 1701  TAD I BCNT10 /GET BAD DATA
    
```

```

1036 7402 ERHLT2, HLT /BAD DATA IN AC
1037 7300 CLA CLL
1040 1773 TAD KHLT /GET EXPECTED DATA
1041 7402 HLT /EXPECTED DATA IN AC
1042 5241 JMP ,=1

/Routine to COMPARE BOOTSTRAP TO BUFFER
1043 0000 COMPAR, 0
1044 1772 TAD STRBUF /GET START OF BUFFER AREA
1045 3301 DCA BCNT10
1046 1771 TAD STRBLK /GET START OF BOOTSTRAP AREA
1047 3302 DCA BCNT11
1050 1770 TAD AMOUNT /GET AMOUNT TO BE CHECKED
1051 7041 CIA
1052 3303 DCA BCNT12 /STORE IN COUNTER
1053 1701 COMSTR, TAD I BCNT10 /GET EXPECTED DATA
1054 7041 CIA
1055 1702 TAD I BCNT11 /GET UNKNOWN VALUE
1056 7640 SZA CLA
1057 5267 JMP ACERR3 /ERROR VALUES NOT THE SAME
1060 2301 ISZ BCNT10
1061 7000 NOP
1062 2302 ISZ BCNT11
1063 7000 NOP
1064 2303 ISZ BCNT12
1065 5253 JMP COMSTR /CONTINUE CHECKING
1066 5643 JMP I COMPAR /EXIT

/Routine to DISPLAY BOOTSTRAP DATA ERRORS
/IF AN ERROR OCCURES THE MACHINE WILL
/HLT WITH BAD ADDRESS IN MQ AND BAD DATA IN AC;
/HIT CONTINUE;
/THE MACHINE WILL HLT WITH EXPECTED DATA IN AC;

1067 1302 ACERR3, TAD BCNT11
1070 7421 MQL /LOAD AC WITH BAD ADDRESS
1071 1702 TAD I BCNT11 /GET BAD DATA
1072 7402 ERHLT3, HLT /AC IS BAD DATA
1073 7300 CLA CLL
1074 1701 TAD I BCNT10 /GET EXPECTED DATA
1075 7402 HLT /AC IS EXPECTED DATA
1076 5275 JMP ,=1

BCNT1, 0
BCNT2, 0
BCNT10, 0
BCNT11, 0
BCNT12, 0
K0070, 0070
K7400, 7600
BUFBIN, SAVBIN

/TEXT FOR "M18-E BOOTSTRAP DATA"
"ADRS" "DATA"

```

```

1107 0215 BOTMES, 215
1110 0212 212
1111 0315 315
1112 0311 311
1113 0270 270
1114 0255 255
1115 0305 305
1116 0240 240
1117 0302 302
1120 0317 317
1121 0317 317
1122 0324 324
1123 0323 323
1124 0324 324
1125 0322 322
1126 0301 301
1127 0320 320
1130 0240 240
1131 0304 304
1132 0301 301
1133 0324 324
1134 0301 301
1135 0215 215
1136 0212 212
1137 0301 301
1140 0304 304
1141 0322 322
1142 0323 323
1143 0240 240
1144 0240 240
1145 0240 240
1146 0240 240
1147 0304 304
1150 0301 301
1151 0324 324
1152 0301 301
1153 0215 215
1154 0212 212

1170 0340
1171 0335
1172 1390
1173 0343
1174 0341
1175 0342
1176 0337
1177 0213
1200 PAGE

/Routine to REPLACE BINARY LOADER,
/BINARY LOADER MAY BE REPLACED BY
/LOAD AND STARTING ADDRESS "RPLBIN"
1200 1777 RPLBIN, TAD EXTSAV /GET OPERATION SWITCHES

```

```

1201 0776' AND K0070
1202 1775' TAD K0DF /MASK BITS 6-8
1203 3213 DCA TOFLD
1204 1774' TAD PRFLD /MAKE BINARY FIELD CDP
1205 3215 TAD OPRFLD /GET PRESENT FIELD CDP
1206 1773' DCA OPRFLD
1207 3331 TAD K7000 /GET START OF BINARY LOADER
1210 1772' DCA BCNT13
1211 3392 TAD BUFBIN /GET START OF BUFFER AREA
1212 1792 DCA BCNT14
1213 0000 TAD I BCNT14 /GET INFORMATION
1214 3791 TOFLD, 0 /MODIFIED BY TEST
1219 0000 OPRFLD, 0 /STORE BINARY WORD
1216 2392 /MODIFIED BY TEST
1217 2391 ISZ BCNT14
1220 5212 ISZ BCNT13
1221 7402 JMP TOFLD -1 /MORE WORDS TO GO
1222 5221 HLT /BINARY LOADER DONE
JMP .-1

```

/ROUTINE TO PUNCH OBJECT TAPE FOR OPERATOR

```

1223 0000 BPUN, 0
1224 7300 CLA CLL
1225 3336 DCA CKSM /CLEAR CHECK SUM
1226 1777' TAD EXTSAV /GET OPERATION SWITCHES
1227 7006 RTL
1230 7700 SMA CLA /SWR2=1 IS LOW SPEED PUNCH
1231 5262 JMP HIPUN /HIGH SPEED PUNCH SELECTED
1232 6046 TLS /LOW SPEED PUNCH SELECTED
1233 4264 GOLEAD, JMS PLOT /GO PUNCH LEADER TRAILER
1234 1350 TAD STRBUP /GET START OF BUFFER
1235 3337 DCA IA /STORE INITIAL ADDRESS TO BE PUNCHED
1236 1350 TAD STRBUP
1237 1771' TAD AMOUNT /GET AMOUNT OF BOOTSTRAP LOCATIONS
1240 3340 DCA FA
1241 1337 TAD IA
1242 7120 STL
1243 4276 PUNL, JMS BINP /TO PUNCH IA AS ORIGIN
1244 1337 TAD IA /GO PUNCH WORD AS TWO LINES OF TAPE
1245 7041 CIA
1246 1340 TAD FA
1247 7650 SMA CLA
1250 5245 JMP .-3 /HAS IT LAST WORD
1251 1737 TAD I IA /IT HAS LAST WORD
1252 7100 CLL /GET WORD TO PUNCH
1253 2337 ISZ IA
1254 5243 JMP PUNL
1255 1336 TAD CKSM
1256 7100 CLL
1257 4276 JMS BINP /GO PUNCH CHECK SUM
1260 4264 JMS PLOT /GO PUNCH LEADER TRAILER
1261 5623 JMP I BPUN /EXIT AND DONE WITH TAPE

1262 6026 HIPUN, PLS /HIGH SPEED PUNCH SELECTED
1263 5233 JMP GOLEAD /GO PUNCH LEADER TRAILER

```

```

1264 0000 PLOT, 0
1265 7300 CLA CLL
1266 1341 TAD M212 /TO PUNCH 212 OCTAL LEADER TRAILER
1267 3342 DCA CTR1
1270 1343 TAD C200 /LEADER TRAILER CODE
1271 4316 JMS PUN /PUNCH
1272 2342 ISZ CTR1
1273 5271 JMP .-2
1274 7300 CLA CLL
1275 5664 JMP I PLOT /EXIT

1276 0000 BINP, 0
1277 3344 DCA TEM1
1300 1344 TAD TEM1
1301 7012 RTR
1302 7012 RTR
1303 7012 RTR
1304 0345 AND SL7
1305 4316 JMS PUN /FIRST TO OCTAL DIGITS IN AC 5-11
1306 1336 TAD CKSM /PUNCH
1307 3336 DCA CKSM
1310 1344 TAD TEM1
1311 0346 AND SL6 /LAST TWO OCTAL DIGITS IN AC 6-11
1312 4316 JMS PUN /PUNCH
1313 1336 TAD CKSM
1314 3336 DCA CKSM
1315 5076 JMP I BINP /EXIT

1316 0000 PUN, 0
1317 3347 DCA ACPAV2
1320 1777' TAD EXTSAV /SAVE CODE TO BE PUNCHED
1321 7006 RTL /GET OPERATION SWITCHES
1322 7700 SMA CLA /SWR2=1 IS LOW SPEED PUNCH
1323 5331 JMP HISPED /HIGH SPEED PUNCH SELECTED
1324 1347 TAD ACPAV2 /GET CODE TO BE PUNCHED
1325 6041 YSF
1326 5325 JMP .-1
1327 6046 TLS
1330 5716 JMP I PUN /PUNCH
1331 1347 HISPED, TAD ACPAV2 /EXIT
1332 6021 PSF /GET CODE TO BE PUNCHED
1333 5332 JMP .-1
1334 6026 PLS
1335 5716 JMP I PUN /PUNCH
/EXIT

1336 0000 CKSM, 0
1337 0000 IA, 0
1340 0000 FA, 0
1341 7566 M212, -212
1342 0000 CTR1, 0
1343 0200 C200, 200
1344 0000 TEM1, 0
1345 0177 SL7, 177
1346 0077 SL6, 77

```

1347 0000 ACSAV2, 0
1350 1600 STRBUF, INBUF
1351 0000 BCNT13, 0
1352 0000 BCNT14, 0
/
1371 0340
1372 1106
1373 1105
1374 0213
1375 0342
1376 1104
1377 0337
1400

PAGE
/
/THE NEXT PAGE IS RESERVED FOR BINARY LOADER
/STORAGE AREA WHILE THE DIAGNOSTIC IS BEING RUN;
/

1400 1400 SAVBIN, SAVBIN
/
1600 PAGE

/
/THE NEXT 40 OCTAL LOCATIONS ARE RESERVED
/FOR THE BOOTSTRAP BUFFER
/
1600 1600 INBUF, INBUF
/

5

2000
2100

0200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0300 11111111 11111111 11111111 11111111 11111111 00000000 00000000 11111111
0400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 00111111
0600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 10111111

1000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1100 11111111 11111111 11111111 11111111 11111111 11110000 00000000 11111111
1200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1300 11111111 11111111 11111111 11111111 11111111 11100000 00000000 01111111
1400 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
1500 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
1600 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
1700 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

2000
2100
2200
2300
2400
2500
2600
2700

3000
3100
3200
3300
3400
3500
3600
3700

4000
4100
4200
4300
4400
4500
4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

PAL10	V141	24-JAN-72	23136	PAGE 1-17	
ACERR1	0750	HIPUN	1262	TEM1	1344
ACERR2	1033	HISPED	1331	TEST1	0223
ACERR3	1067	HLTLOC	0341	TEST2	0240
AGSAV1	0563	IA	1337	TEST3	0254
AGSAV2	1347	INBUF	1600	TEST4	0311
ADDHLT	0545	INHDMF	0274	TEXTLG	0361
AMOUNT	0340	K0007	0765	TEXTMS	0360
BCNT1	1077	K0010	0764	THSFLO	0462
BCNT10	1101	K0070	1104	TOPLO	1213
BCNT11	1102	K0207	0952	TOTOP	0745
BCNT12	1103	K0212	0951	TYPE	0421
BCNT13	1351	K0215	0950	TYPE	0505
BCNT14	1352	K0240	0954	WATHLT	0334
BCNT2	1100	K0260	0953	XBEGIN	0937
BCNT3	0766	K7600	1105	XCHK	0345
BCNT4	0767	K7774	0947	XINBUF	0942
BCNT5	0770	KCDF	0342	XLOD	0346
BCNT6	0564	KCHK	0760		
BCNT7	0565	KHLT	0343		
BCNT8	0566	KLOD	0761		
BCNT9	0567	LASTLC	0762		
BEGIN	0200	LENGTH	0555		
HELL	0514	LODCHK	0743		
BINFLO	1013	LODMEM	0643		
BINP	1276	M212	1341		
BOYNES	1107	MEMGO	0600		
BPUN	1223	MOVBIN	1000		
BUPBIN	1106	MOVBUF	0400		
C200	1343	HQL	7421		
CAF	0907	NEWFLO	0660		
CHKADD	0920	OGTEL	0464		
CHKHLT	1023	OPRFLO	1215		
CHKMEM	0666	PLOT	1264		
CHKTH0	0710	PRFLO	0213		
CKSH	1336	PUN	1316		
COMPAR	1043	PUNL	1243		
COMSTR	1053	RESTR3	0266		
CTR1	1342	RESTR4	0323		
ENDTST	0714	RPLBIN	1200		
ERHLT1	0753	SAPADD	0506		
ERHLT2	1036	SAVBIN	1400		
ERHLT3	1072	SETFLO	1015		
EXIT	0640	SL6	1346		
EXTCHK	0615	SL7	1345		
EXTSAV	0337	SPECHK	0725		
FA	1340	STRADD	0336		
FIRPAS	0344	STRBLK	0335		
FLOAMN	0763	STRBUF	1350		
FLOCHK	0742	STRCHK	0623		
FLODD	0746	STRTUP	0934		
FRHFLD	0703	STRTYP	0441		
GOLEAD	1233	TBCGIN	0744		

ERRORS DETECTED: 0
 LINKS GENERATED: 63
 RUN-TIME: 7 SECONDS
 2K CORE USED

```

/
/M18-E BOOTSTRAP DIAGNOSTIC
/
/COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION
/
6007 CAF=6007
7421 MQL=7421
/
/THE LOW VERSION STARTING ADDRESS IS 0200,
/THE HIGH VERSION STARTING ADDRESS IS 4200;
/
/OPERATION SWITCH REGISTER FOR M18-E DIAGNOSTIC
/
/SWR0=0 VERIFICATION BY OCTAL DUMP TYPE OUT
/SWR0=1 VERIFICATION BY BINARY OBJECT TAPE
/SWR1=1 PUNCH BINARY OBJECT TAPE
/SWR2=1 LOW SPEED PUNCH
/SWR2=0 HIGH SPEED PUNCH
/SWR6=8 FIELD OF BINARY LOADER
/SWR9=11 AMOUNT OF EXTENDED FIELDS
/
/ROUTINE TO SAVE THE INITIAL BLOCK ADDRESS
/AND THE STARTUP ADDRESS OF THE BOOTSTRAP;
/SAVE BINARY LOADER IN PROGRAM BUFFER AREA;
/CHECK TO MAKE SURE THAT ADDRESSES DO NOT INTERFERE
/WITH THE LOCATIONS OF THE DIAGNOSTIC;
/ALSO SAVE OPERATION SWITCHES;
/
4200 *4200
/
4200 7604 BEGIN, LAS
4201 3335 DCA STRBLK /SAVE START OF BOOTSTRAP BLOCK
4202 7402 HLT
4203 7604 LAS
4204 3336 DCA STRADD /SAVE STARTUP ADDRESS OF BOOTSTRAP
4205 7402 HLT
4206 7604 LAS
4207 3337 DCA EXTSAV /SAVE OPERATING SWITCHES
4208 6224 RIF
4209 1342 TAD KCDF /MAKE PRESENT FIELD CDF
4210 3213 DCA PREFLD
4211 3213 DCA PREFLD, 0
4212 0000
4213 1335 TAD STRBLK /GET START OF BLOCK
4214 1340 TAD AMOUNT /GET LENGTH OF BLOCK +1
4215 3341 DCA HLTLOC /MAKE HALT LOCATION
4216 1344 TAD FIRPAS /GET PASS FLAG
4217 7640 SEA CLA /IS IT FIRST PASS
4218 4777 JMS MOVBIN /YES, MOVE THE BINARY LOADER
4219 4776 JMS CHKADD /CHECK THAT SWITCHES DO NOT CONFLICT
/
/LOAD MEMORY WITH DATA PATTERN 2525 AND SIGNAL
/OPERATOR TO TOGGLE SWITCH AS MEMORY IS BEING CHECKED;
/SWITCH SHOULD BE TOGGLED AT LEAST 10 TIMES
/OPERATOR MUST HIT TTY KEY TO EXIT TO NEXT TEST;
/THIS WILL VERIFY THAT THE PROCESSOR AND THE MEMORY
    
```


/WILL NOT BE AFFECTED BY BOOTSTRAP WHEN THE COMPUTER
/IS RUNNING.

```

4223 6007 TEST1, CAF /CLEAR THE WORLD
4224 3344 DCA FIRPAS /GET JMS FOR LOAD
4225 1746 TAD I XL0D /SETUP FOR LOAD
4226 3775' DCA LODCHK /LOAD MEMORY WITH DATA PATTERN
4227 4774' JMS MEMGO /DATA PATTERN TO BE USED
4230 2525 2525 /SIGNAL OPERATOR
4231 4773' JMS BELL /GET JMS FOR CHECK
4232 1745 TAD I XCHK /SETUP FOR CHECK MEMORY
4233 3775' DCA LODCHK /CHECK MEMORY
4234 4774' JMS MEMGO /COMPARE TO THIS PATTERN
4235 2525 2525
4236 6031 KSF
4237 5234 JMP ,=3 /WAIT FOR OPERATOR TO CONTINUE
    
```

/LOAD MEMORY WITH DATA PATTERN 5252 AND SIGNAL
/OPERATOR TO TOGGLE SWITCH AS MEMORY IS BEING CHECKED;
/SWITCH SHOULD BE TOGGLED AT LEAST 10 TIMES
/OPERATOR MUST HIT TTY KEY TO EXIT THIS TEST
/THIS WILL VERIFY THAT THE PROCESSOR AND THE MEMORY
/WILL NOT BE AFFECTED BY BOOTSTRAP WHEN THE COMPUTER
/IS RUNNING.

```

4240 7300 TEST2, CLA CLL
4241 1746 TAD I XL0D /GET JMS FOR LOAD
4242 3775' DCA LODCHK /SETUP FOR LOAD MEMORY
4243 4774' JMS MEMGO /LOAD MEMORY WITH DATA PATTERN
4244 5252 5252 /DATA PATTERN TO BE USED
4245 4773' JMS BELL /SIGNAL OPERATOR
4246 1745 TAD I XCHK
4247 3775' DCA LODCHK /SETUP FOR CHECK MEMORY
4250 4774' JMS MEMGO /CHECK MEMORY
4251 5252 5252 /COMPARE TO THIS PATTERN
4252 6031 KSF
4253 5250 JMP ,=3 /WAIT FOR OPERATOR TO CONTINUE
    
```

/LOAD MEMORY WITH DATA PATTERN 2525,
/LOAD A HALT INTO BOOTSTRAP BUFFER #1 JUST
/IN CASE THE BOOTSTRAP DOESN'T HANG,
/THEN GO HALT AND WAIT FOR THE OPERATOR TO
/TOGGLE THE BOOTSTRAP SWITCH ONCE,
/THE OPERATOR MUST THEN RESTART THE PROGRAM AT
/LOCATION 0266/4266.

```

4254 7300 TEST3, CLA CLL
4255 1746 TAD I XL0D /GET JMS FOR LOAD
4256 3775' DCA LODCHK /SETUP FOR LOAD
4257 4774' JMS MEMGO /LOAD MEMORY WITH DATA PATTERN
4260 2525 2525
4261 1343 TAD KHLT
4262 3734 DCA I WATHLT /STORE WAIT HALT AT END OF DIAG,
4263 1343 TAD KHLT
4264 3741 DCA I HLTLOC /STORE HLT IN BOOTSTRAP BLOCK #1
    
```

```

4265 5734 JMP I WATHLT /GO HALT AND WAIT FOR OPERATOR TO
/TOGGLE SWITCH AND RESTART PROGRAM
    
```

/CHECK MEMORY NOT OCCUPIED BY THE BOOTSTRAP
/FOR CORRECT PATTERN OF 2525, IF OPERATOR HAS
/INHIBITED VISUAL PRINT OUT ASSUME THAT THE
/BINARY OBJECT TAPE OF THE BOOTSTRAP INFORMATION
/HAS BEEN LOADED INTO CORE AND COMPARE THIS OBJECT
/TO THE INFORMATION BOOTSTRAPED INTO CORE,
/IF VISUAL CHECK PRINT OUT BOOTSTRAP BUFFER INFORMATION,
/IF OPERATOR HAS SELECTED TO PUNCH NEW OBJECT TAPE
/HALT AND WAIT FOR OPERATOR TO PREPARE THE
/PAPER TAPE PUNCH SELECTED,
/OPERATOR MUST HIT CONTINUE TO PUNCH TAPE.

```

4266 7300 RESTR3, CLA CLL
4267 1337 TAD EXTSAV /GET OPERATION SWITCHES
4270 7710 SPA CLA /SHR0=1 IS INHIBIT OCTAL DUMP
4271 5274 JMP INHDMP /INHIBIT OCTAL DUMP OF BOOTSTRAP
4272 4772' JMS MOVBUF /MOVE BOOTSTRAP TO BUFFER AREA
4273 4771' JMS TYPBUF /TYPE OCTAL DUMP OF BOOTSTRAP
4274 1745 INHDMP, TAD I XCHK /GET JMS FOR CHECK
4275 3775' DCA LODCHK /SETUP FOR CHECK MEMORY
4276 7320 CLA CLL CML
4277 4774' JMS MEMGO
4300 2525 2525 /CHECK MEMORY OTHER THAN BOOTSTRAP
4301 4770' JMS CMKHLT
4302 4767' JMS COMPAR /CHECK HLT STORED IN BLOCK #1
4303 1337 TAD EXTSAV /COMPARE BOOTSTRAP TO BUFFER AREA
4304 7004 RAL /GET OPERATION SWITCHES
4305 7700 SNA CLA
4306 5311 JMP TEST4 /SHR1=1 IS PUNCH NEW TAPE
4307 7402 HLT /INHIBIT TAPE AND GO TO NEXT TEST
4310 4766' JMS BPUN /WAIT FOR OPERATOR TO PREPARE PUNCH
/OPERATOR MUST HIT CONTINUE TO PUNCH
    
```

/LOAD MEMORY WITH DATA PATTERN 5252,
/LOAD A HALT INTO BOOTSTRAP BUFFER #1 JUST
/IN CASE THE BOOTSTRAP DOESN'T HANG,
/THEN GO HALT AND WAIT FOR THE OPERATOR TO
/TOGGLE THE BOOTSTRAP SWITCH ONCE,
/THE OPERATOR MUST THEN RESTART THE PROGRAM AT
/LOCATION 0323/4323.

```

4311 7300 TEST4, CLA CLL
4312 1746 TAD I XL0D /GET JMS FOR LOAD
4313 3775' DCA LODCHK /SETUP FOR LOAD
4314 4774' JMS MEMGO /LOAD MEMORY WITH DATA PATTERN
4315 5252 5252
4316 1343 TAD KHLT
4317 3734 DCA I WATHLT /STORE WAIT HALT AT END OF DIAG,
4320 1343 TAD KHLT
4321 3741 DCA I HLTLOC /STORE A HLT IN BLOCK #1
4322 5734 JMP I WATHLT /GO HALT AND WAIT FOR OPERATOR TO
/TOGGLE SWITCH AND RESTART PROGRAM
    
```

/CHECK MEMORY NOT OCCUPIED BY THE BOOTSTRAP
 /FOR A CORRECT PATTERN OF 9252, THEN COMPARE
 /THE BOOTSTRAP INFORMATION TO THAT FOUND IN
 /TEST 3.

4323	7300	RESTR4, CLA CLL	
4324	1745	TAD I XCHK	/GET JMS FOR CHECK
4325	3775	DCA LODCHK	/SETUP FOR CHECK MEMORY
4326	7320	CLA CLL CML	
4327	4774	JMS MEMGO	/CHECK MEMORY OTHER THAN BOOTSTRAP
4328	5252	9252	
4331	4770	JMS CHKHLT	/CHECK HLT IN BLOCK +1
4332	7482	HLT	/END OF TEST
4333	5223	JMP TEST1	/LOOP ON PROGRAM

4334	5640	/
4335	0000	WATHLT, INBUF +40
4336	0000	STRBLK, 0
4337	0000	STRADD, 0
4340	0040	EXTSAV, 0
4341	0000	AMOUNT, 0040
4342	6201	HLTLOC, 0
4343	7402	KCDF, CDF
4344	7777	KHLT, HLT
4345	4760	FIRPAS, 7777
4346	4761	XCHK, KOHK
		XLOD, KLOD

4366	5223	
4367	5043	
4370	5023	
4371	4421	
4372	4400	
4373	4514	
4374	4600	
4375	4743	
4376	4520	
4377	5000	
	4400	

PAGE

/ROUTINE TO MOVE BOOTSTRAP INFORMATION FROM
 /BOOTSTRAP AREA TO PROGRAM BUFFER AREA,

4400	0000	MOVBUF, 0
4401	1777	TAD STRBLK
4402	3364	DCA BCNT6
4403	1776	TAD AMOUNT
4404	7041	CIA
4405	3365	DCA BCNT7
4406	1362	TAD XINBUF
4407	3366	DCA BCNT8
4410	1764	TAD I BCNT6
4411	3766	DCA I BCNT8
4412	2364	ISE BCNT6
4413	7000	NOP

4414	2366	ISE BCNT8
4415	7000	NOP
4416	2365	ISE BCNT7
4417	5210	JMP MOVBUF +10
4420	5600	JMP I MOVBUF

/ROUTINE TO TYPE BOOTSTRAP DATA INFORMATION

4421	0000	TYPBUF, 0
4422	1360	TAD TEXTMS
4423	3364	DCA BCNT6
4424	1361	TAD TEXTLG
4425	3365	DCA BCNT7
4426	1764	TAD I BCNT6
4427	4305	JMS TYPE
4430	2364	ISE BCNT6
4431	7000	NOP
4432	2365	ISE BCNT7
4433	5226	JMP TYPBUF +5
4434	1777	TAD STRBLK
4435	3364	DCA BCNT6
4436	1776	TAD AMOUNT
4437	7041	CIA
4440	3365	DCA BCNT7
4441	1344	STRTYP, TAD BCNT6
4442	4244	JMS OCTEL
4443	1347	TAD K7774
4444	3366	DCA BCNT8
4445	1354	TAD K0240
4446	4305	JMS TYPE
4447	2366	ISE BCNT8
4450	5245	JMP :=3
4451	1764	TAD I BCNT6
4452	4244	JMS OCTEL
4453	1350	TAD K0215
4454	4305	JMS TYPE
4455	1351	TAD K0212
4456	4305	JMS TYPE
4457	2364	ISE BCNT6
4460	7000	NOP
4461	2365	ISE BCNT7
4462	5241	JMP STRTYP
4463	5621	JMP I TYPBUF

/MAKE A SPACE

/ROUTINE TO TYPE OCTAL INFORMATION,

4464	0000	OCTEL, 0
4465	7106	RTL CLL
4466	7006	RTL
4467	3363	DCA ACSAV1
4470	1347	TAD K7774
4471	3367	DCA BCNT9
4472	1363	TAD ACSAV1
4473	0775	AND K0007
4474	1353	TAD K0260

```

4475 4305 JMS TYPE
4476 1363 TAD ACSAV1
4477 7006 RTL
4500 7004 RAL
4501 3363 DCA ACSAV1
4502 2367 ISE BCNT9
4503 5272 JMP ,=11
4504 5664 JMP I OCTEL
    
```

```

/TYPE, 0
4505 0000 TYPE, 0
4506 6046 TLS
4507 6041 TSF
4510 5307 JMP ,=1
4511 6042 TCF
4512 6032 KCC
4513 5705 JMP I TYPE
    
```

```

/BELL, 0
4514 0000 BELL, 0
4515 1392 TAD K0207
4516 4305 JMS TYPE
4517 5714 JMP I BELL
    
```

```

/ROUTINE TO CHECK THAT ADDRESSES SUBMITTED BY OPERATOR
/DO NOT CONFLICT WITH DIAGNOSTIC, IF SWITCH ERROR OCCURES
/THE COMPUTER SHOULD HALT, RE-SET SWITCH FOR STARTING
/ADDRESS OF BLOCK AND HIT CONTINUE TO TRY AGAIN,
    
```

```

4520 0000 CHKADD, 0
4521 7300 CLA CLL
4522 1356 TAD SAFADD /LAST LOCATION USED
4523 7040 CMA
4524 1777 TAD STRBLK /GET START OF BOOTSTRAP BLOCK
4525 7630 SZL CLA /DOES BLOCK INTERFER WITH DIAG,
4526 5334 JMP STRTUP /OK, CHECK STARTUP ADDRESS
4527 1357 TAD XBEGIN /GET FIRST LOCATION USED
4530 7041 CIA
4531 1774 TAD HLTLOC /COMPARE TO THIS VALUE
4532 7630 SZL CLA /DOES BLOCK INTERFER WITH DIAG,
4533 5345 JMP ADDHLT /STARTING BLOCK ADDRESS ERROR
4534 1777 STRTUP, TAD STRBLK /GET START OF BLOCK
4535 7041 CIA
4536 1773 TAD STRADD /GET STARTUP ADDRESS
4537 7510 SPA /WAS ADDRESS OK
4540 5345 JMP ADDHLT /NO, ERROR
4541 7161 CIA STL
4542 1355 TAD LENGTH /LENGTH OF BLOCK
4543 7620 SNL CLA /WAS STARTUP ADDRESS OK
4544 5720 JMP I CHKADD /YES, START TEST
4545 7602 ADDHLT, HLT CLA /SWITCH SETTING ERROR
4546 5772 JMP BEGIN /RESET SWITCH REGISTER TO START OF
/BLOCK AND HIT CONTINUE TO TRY AGAIN

4547 7774 K7774, 7774
4550 0215 K0215, 0215
4551 0212 K0212, 0212
    
```

```

4552 0207 K0207, 0207
4553 0260 K0260, 0260
4554 0240 K0240, 0240
4555 0037 LENGTH, 0037
4556 5777 SAFADD, INBUF +177
4557 4177 XBEGIN, BEGIN -1
4560 5107 TEXTMS, BOTMES
4561 7732 TEXTLG, 7732
4562 5600 XINBUF, INBUF
4563 0000 ACSAV1, 0
4564 0000 BCNT6, 0
4565 0000 BCNT7, 0
4566 0000 BCNT8, 0
4567 0000 BCNT9, 0
    
```

```

/
4572 4200
4573 4336
4574 4341
4575 4765
4576 4340
4577 4335
4600
    
```

```

PAGE
/ROUTINE TO DETERMINE FIELDS TO BE CHECKED
    
```

```

4600 0000 MEMGO, 0
4601 1777 TAD PREFLD /GET PRESENT FIELD CDF
4602 3346 DCA FLDGO
4603 7630 SZL CLA
4604 5325 JMP SPECHK /CHECK FOR ALL BUT PROG. + BOOTSTRAP
4605 1776 TAD XBEGIN /GET STARTING ADDRESS OF CHECK
4606 3345 DCA TSTOP
4607 1362 TAD LASTLC /GET ENDING ADDRESS OF CHECK
4610 3344 DCA TBEGIN
4611 4342 JMS FLDCHK /ENTER ROUTINE TO LOAD OR CHECK
4612 7340 CLA CLL CMA
4613 3345 DCA TSTOP /START AT 0
4614 3344 DCA TBEGIN /END AT 0
4615 1775 EXTCHK, TAD EXTSAV
4616 0365 AND K0007
4617 7450 SNA /IS IT TEST EXTENDED MEMORY
4620 5240 JMP EXIT /NO DO NOT TEST EXTENDED MEMORY
4621 7040 CMA
4622 3363 DCA FLDAMN /SETUP FOR FIELDS TO TEST
4623 3366 DCA BCNT3 /START WITH 0
4624 1366 TAD BCNT3
4625 1774 TAD KCDF
4626 3346 DCA FLDGO /FIELD TO BE CHECKED
4627 6224 RIF
4630 7041 CIA
4631 1366 TAD BCNT3
4632 7640 SZA CLA /ARE WE IN THIS FIELD
4633 4342 JMS FLDCHK /NO, ENTER ROUTINE TO LOAD OR CHECK
4634 1366 TAD BCNT3
4635 1364 TAD K0010
    
```

24-JAN-72 23:37 PAGE 10-4

```

4636 2363   ISE FLDAMN OCCUPIED BY THE BOOTSTRAP
4637 5223   JMP STRCHK CLM CLL PATTERN OF 9292, THEN COMPARE
4640 7300   EXIT, ISE MEMGO INFORMATION TO THAT FOUND IN
4641 2200   JMP I MEMGO /EXIT
4642 5600

/Routine to load memory with data pattern
4643 0000   LODMEM, 0 /GET JMS FOR CHECK
4644 1777   TAD PREFLD /SETUP FOR CHECK MEMORY
4645 3262   DCA THSFLD /SETUP FOR PRESENT FIELD
4646 1643   TAD I LODMEM /CHECK MEMORY OTHER THAN BOOTSTRAP
4647 3367   DCA BCNT4 /GET STARTING ADDRESS
4648 2243   ISE LODMEM /CHECK HLT IN BLOCK #1
4649 1643   TAD I LODMEM /END OF TEST
4650 3370   DCA BCNT5 /GET ENDING ADDRESS
4651 2243   ISE LODMEM
4652 1643   TAD I LODMEM /GET FIELD TO 00
4653 3260   DCA NEWFLD
4654 2243   ISE LODMEM
4655 1600   TAD I MEMGO /GET DATA PATTERN
4656 0000   NEWFLD, 0 /MODIFIED BY TEST
4657 3767   DCA I BCNT4
4658 0000   THSFLD, 0 /MODIFIED BY TEST
4659 4314   JMS ENDTST /IS IT END OF TEST
4660 9257   JMP NEWFLD -1 /NO, CONTINUE
4661 5643   JMP I LODMEM /YES, EXIT

/Routine to check memory for correct data pattern;
4666 0000   CHKMEM, 0
4667 1777   TAD PREFLD /GET PRESENT FIELD
4668 3310   DCA CHKTHS
4669 1666   TAD I CHKMEM /GET STARTING ADDRESS
4670 3367   DCA BCNT4
4671 2266   ISE CHKMEM
4672 1666   TAD I CHKMEM /GET ENDING ADDRESS
4673 3370   DCA BCNT5
4674 2266   ISE CHKMEM
4675 1666   TAD I CHKMEM /GET FIELD TO TEST
4676 3303   DCA FRMFLD /GET STRAP INFORMATION FROM
4677 2266   ISE CHKMEM PROGRAM BUFFER AREA,
4678 1600   TAD I MEMGO /GET EXPECTED DATA
4679 0000   FRMFLD, 0
4680 7041   CIA
4681 1767   TAD I BCNT4 /GET DATA PATTERN
4682 7640   SZA CLA /ARE THEY THE SAME
4683 5350   JMP ACERR1 /NO, INDICATE
4684 0000   CMKTHS, 0 /MODIFIED BY TEST
4685 4314   JMS ENDTST /IS IT END OF TEST
4686 5302   JMP FRMFLD -1 /NO, CONTINUE
4687 5666   JMP I CHKMEM /YES EXIT

/Routine to check for end of test

```

PAL17 V141 24-JAN-72 23:37 PAGE 10-4

```

4714 0000   ENDTST, 0
4715 1367   TAD BCNT4 /GET ENDING ADDRESS
4716 7041   CIA
4717 1370   TAD BCNT5 /GET PRESENT ADDRESS
4718 7650   SNA CLA /IS IT LAST ADDRESS TO TEST
4719 2314   ISE ENDTST /YES, EXIT
4720 2367   ISE BCNT4 BOOTSTRAP DATA INFORMATION
4721 5714   JMP I ENDTST /EXIT
4722 5714   JMP I ENDTST /EXIT

/Routine to test all but prog, + bootstrap
4725 1362   SPECHK, TAD LASTLC /GET START OF CHECK
4726 3344   DCA TBEGIN
4727 7340   CLA CLL CMA
4728 1773   TAD STRBLK /GET END OF CHECK
4729 3345   DCA TSTOP
4730 7301   JMS FLDCHK /ENTER ROUTINE TO LOAD OR CHECK
4731 7301   CLA CLL IAO
4732 1772   TAD HLTLOC
4733 3344   DCA TBEGIN
4734 1776   TAD XBEGIN /GET END OF CHECK
4735 3345   DCA TSTOP
4736 4342   JMS FLDCHK /ENTER ROUTINE TO LOAD OR CHECK
4737 5215   JMP EXTCHK /EXIT

/Routine to load or check memory
4742 0000   FLDCHK, 0
4743 0000   LODCHK, 0 /MODIFIED BY TEST
4744 0000   TBEGIN, 0 /MODIFIED BY TEST
4745 0000   TSTOP, 0 /MODIFIED BY TEST
4746 0000   FLDGO, 0 /MODIFIED BY TEST
4747 5742   JMP I FLDCHK /EXIT

/Routine to display core pattern errors,
/IF AN ERROR OCCURS THE MACHINE WILL
/HLT WITH BAD ADDRESS IN MQ AND BAD DATA IN AC,
/NOT CONTINUE,
/THE MACHINE WILL HLT WITH EXPECTED DATA IN AC,
4750 1367   ACERR1, TAD BCNT4 /GET BAD ADDRESS
4751 7421   MQL /LOAD AC TO MQ
4752 1767   TAD I BCNT4 /GET BAD DATA PATTERN
4753 7402   ERHLT, HLT /AC = EXPECTED DATA PATTERN FOUND
4754 7300   CLA CLL
4755 1600   TAD I MEMGO /GET GOOD NUMBER
4756 7402   HLT /AC = EXPECTED DATA PATTERN
4757 5356   JMP -1

4760 4266   KCHK, JMS CHKMEM
4761 4243   KLOD, JMS LODMEM
4762 5641   LASTLC, INBUF +41
4763 0000   FLDAMN, 0
4764 0010   K0010, 0010

```

4765 0007 K0007, 0007
 4766 0000 BCNT3, 0
 4767 0000 BCNT4, 0
 4770 0000 BCNT5, 0

4772 4341
 4773 4335
 4774 4342
 4775 4337
 4776 4557
 4777 4213
 5000

PAGE

/ROUTINE TO MOVE THE BINARY LOADER

5000 0000 MOVBIN, 0
 5001 1306 TAD BUFBIN
 5002 3277 DCA BCNT1 /SETUP BINARY LOADER BUFFER
 5003 1777 TAD PREFLD
 5004 3215 DCA SETFLD
 5005 1305 TAD K7400
 5006 3300 DCA BCNT2 /SETUP FOR BINARY LOADER
 5007 1776 TAD EXTSAV
 5010 0304 AND K0070 /MASK 0-8
 5011 1775 TAD KCDF
 5012 3213 DCA BINFLD /FIELD OF BINARY LOADER
 5013 0000 BINFLD, 0 /MODIFIED BY TEST
 5014 1700 TAD I BCNT2 /GET BINARY WORD
 5015 0000 SETFLD, 0 /MODIFIED BY TEST
 5016 3677 DCA I BCNT1 /STORE IN BUFFER AREA
 5017 2277 ISZ BCNT1
 5020 2300 ISZ BCNT2
 5021 5213 JMP BINFLD /MORE WORDS TO GO
 5022 5600 JMP I MOVBIN /EXIT

/ROUTINE TO CHECK HALT AFTER BOOTSTRAP

5023 0000 CHKHLT, 0
 5024 1774 TAD HLTLOC /GET HALT LOCATION
 5025 3301 DCA BCNT10
 5026 1701 TAD I BCNT10 /GET HALT
 5027 7041 CIA
 5030 1773 TAD KHLT /GET EXPECTED VALUE
 5031 7650 SNA CLA /HERE THEY THE SAME
 5032 5623 JMP I CHKHLT /YES EXIT

/ROUTINE TO DISPLAY ERROR FOR BAD HLT LOCATION
 /IF AN ERROR OCCURES THE MACHINE WILL
 /HLT WITH BAD ADDRESS IN MQ AND BAD DATA IN AC;
 /HIT CONTINUE,
 /THE MACHINE WILL HLT WITH EXPECTED DATA IN AC;

5033 1301 ACERR2, TAD BCNT10 /GET BAD ADDRESS
 5034 7421 MQL /LOAD M.O.
 5035 1701 TAD I BCNT10 /GET BAD DATA

5036 7402 ERHLT2, HLT /BAD DATA IN AC
 5037 7300 CLA CLL
 5040 1773 TAD KHLT /GET EXPECTED DATA
 5041 7402 HLT /EXPECTED DATA IN AC
 5042 5241 JMP .-1

/ROUTINE TO COMPARE BOOTSTRAP TO BUFFER

5043 0000 COMPAR, 0
 5044 1772 TAD STRBUF /GET START OF BUFFER AREA
 5045 3301 DCA BCNT10
 5046 1771 TAD STRBLK /GET START OF BOOTSTRAP AREA
 5047 3302 DCA BCNT11
 5050 1770 TAD AMOUNT /GET AMOUNT TO BE CHECKED
 5051 7041 CIA
 5052 3303 DCA BCNT12
 5053 1701 COMSTR, TAD I BCNT10 /STORE IN COUNTER
 5054 7041 CIA /GET EXPECTED DATA
 5055 1702 TAD I BCNT11 /GET UNKNOWN VALUE
 5056 7640 SZA CLA
 5057 5267 JMP ACERR3 /ERROR VALUES NOT THE SAME
 5060 2301 ISZ BCNT10
 5061 7000 NOP
 5062 2302 ISZ BCNT11
 5063 7000 NOP
 5064 2303 ISZ BCNT12
 5065 5253 JMP COMSTR /CONTINUE CHECKING
 5066 5643 JMP I COMPAR /EXIT

/ROUTINE TO DISPLAY BOOTSTRAP DATA ERRORS
 /IF AN ERROR OCCURES THE MACHINE WILL
 /HLT WITH BAD ADDRESS IN MQ AND BAD DATA IN AC;
 /HIT CONTINUE,
 /THE MACHINE WILL HLT WITH EXPECTED DATA IN AC;

5067 1302 ACERR3, TAD BCNT11
 5070 7421 MQL /LOAD AC WITH BAD ADDRESS
 5071 1702 TAD I BCNT11 /GET BAD DATA
 5072 7402 ERHLT3, HLT /AC IS BAD DATA
 5073 7300 CLA CLL
 5074 1701 TAD I BCNT10 /GET EXPECTED DATA
 5075 7402 HLT /AC IS EXPECTED DATA
 5076 5275 JMP .-1

5077 0000 BCNT1, 0
 5100 0000 BCNT2, 0
 5101 0000 BCNT10, 0
 5102 0000 BCNT11, 0
 5103 0000 BCNT12, 0
 5104 0070 K0070, 0070
 5105 7600 K7400, 7600
 5106 3400 BUFBIN, SAVBIN

/TEXT FOR "M18-E BOOTSTRAP DATA"
 / "ADRS" "DATA"

5107 0215 BOTMES, 215
 5110 0212 212
 5111 0315 315
 5112 0311 311
 5113 0270 270
 5114 0255 255
 5115 0305 305
 5116 0240 240
 5117 0302 302
 5120 0317 317
 5121 0317 317
 5122 0324 324
 5123 0323 323
 5124 0324 324
 5125 0322 322
 5126 0301 301
 5127 0320 320
 5130 0240 240
 5131 0304 304
 5132 0301 301
 5133 0324 324
 5134 0301 301
 5135 0215 215
 5136 0212 212
 5137 0301 301
 5140 0304 304
 5141 0322 322
 5142 0323 323
 5143 0240 240
 5144 0240 240
 5145 0240 240
 5146 0240 240
 5147 0304 304
 5150 0301 301
 5151 0324 324
 5152 0301 301
 5153 0215 215
 5154 0212 212

5170 4340
 5171 4335
 5172 5350
 5173 4343
 5174 4341
 5175 4342
 5176 4337
 5177 4213
 5200

PAGE

/ROUTINE TO REPLACE BINARY LOADER,
 /BINARY LOADER MAY BE REPLACED BY
 /LOAD AND STARTING ADDRESS "RPLBIN"

5200 1777' RPLBIN, TAD EXTSAV /GET OPERATION SWITCHES

5201 0776' AND K0070 /MASK BITS 6-8
 5202 1775' TAD K0DF
 5203 3213 DCA TOFLD /MAKE BINARY FIELD CDF
 5204 1774' TAD PREFLD /GET PRESENT FIELD CDF
 5205 3215 DCA OPRFLD
 5206 1773' TAD K7600 /GET START OF BINARY LOADER
 5207 3351 DCA BCNT13
 5210 1772' TAD BUFBIN /GET START OF BUFFER AREA
 5211 3352 DCA BCNT14
 5212 1752 TAD I BCNT14 /GET INFORMATION
 5213 0000 TOFLD, 0 /MODIFIED BY TEST
 5214 3751 DCA I BCNT13 /STORE BINARY WORD
 5215 0000 OPRFLD, 0 /MODIFIED BY TEST
 5216 2352 ISE BCNT14
 5217 2351 ISE BCNT13
 5220 5212 JMP TOFLD -1 /MORE WORDS TO GO
 5221 7402 HLT /BINARY LOADER DONE
 5222 5221 JMP ,=1

/ROUTINE TO PUNCH OBJECT TAPE FOR OPERATOR

5223 0000 BPUN, 0
 5224 7300 CLA CLL
 5225 3336 DCA CKSM /CLEAR CHECK SUM
 5226 1777' TAD EXTSAV /GET OPERATION SWITCHES
 5227 7006 RTL
 5230 7700 SMA CLA /SWR2=1 IS LOW SPEED PUNCH
 5231 5202 JMP HIPUN /HIGH SPEED PUNCH SELECTED
 5232 6046 TLS /LOW SPEED PUNCH SELECTED
 5233 4264 GLEAD, JMS PLOT /GO PUNCH LEADER TRAILER
 5234 1350 TAD STRBUF /GET START OF BUFFER
 5235 3337 DCA IA /STORE INITIAL ADDRESS TO BE PUNCHED
 5236 1350 TAD STRBUF
 5237 1771' TAD AMOUNT /GET AMOUNT OF BOOTSTRAP LOCATIONS
 5240 3340 DCA FA
 5241 1337 TAD IA
 5242 7120 STL /TO PUNCH IA AS ORIGIN
 5243 4276 PUNL, JMS BINP /GO PUNCH WORD AS TWO LINES OF TAPE
 5244 1337 TAD IA
 5245 7041 CIA
 5246 1340 TAD FA
 5247 7050 SNA CLA /WAS IT LAST WORD
 5250 5255 JMP ,+5 /IT WAS LAST WORD
 5251 1737 TAD I IA /GET WORD TO PUNCH
 5252 7100 CLL
 5253 2337 ISE IA
 5254 5243 JMP PUNL
 5255 1336 TAD CKSM
 5256 7100 CLL
 5257 4276 JMS BINP /GO PUNCH CHECK SUM
 5260 4264 JMS PLOT /GO PUNCH LEADER TRAILER
 5261 5023 JMP I BPUN /EXIT AND DONE WITH TAPE
 5262 6026 HIPUN, PLS /HIGH SPEED PUNCH SELECTED
 5263 5233 JMP GLEAD /GO PUNCH LEADER TRAILER

```

5264 0000 /
5265 0000 PLOT, 0
5266 1341 CLA CLL
5267 3342 TAD M212 /TO PUNCH 212 OCTAL LEADER TRAILER
5270 1343 DCA CTR1
5271 4316 TAD C200 /LEADER TRAILER CODE
5272 2342 JMS PUN /PUNCH
5273 5271 ISZ CTR1
5274 7300 JMP .+2
5275 5664 CLA CLL /EXIT
      JMP I PLOT

/
5276 0000 RINP, 0
5277 3344 DCA TEM1
5302 1344 TAD TEM1
5301 7012 RTR
5302 7012 RTR
5303 7012 RTR
5304 7345 AND SL7 /FIRST TO OCTAL DIGITS IN AC 5=11
5305 4316 JMS PUN /PUNCH
5306 1336 TAD CKSM
5307 3336 DCA CKSM
5310 1344 TAD TEM1
5311 2346 AND SL6 /LAST TWO OCTAL DIGITS IN AC 6=11
5312 4316 JMS PUN /PUNCH
5313 1336 TAD CKSM
5314 3336 DCA CKSM
5315 5676 JMP I RINP /EXIT

/
5316 0000 PUN, 0
5317 3347 DCA ACSAV2 /SAVE CODE TO BE PUNCHED
5320 1777 TAD EXTSAV /GET OPERATION SWITCHES
5321 7006 RTL
5322 7700 SMA CLA /SWR2=1 IS LOW SPEED PUNCH
5323 5331 JMP HISPED /HIGH SPEED PUNCH SELECTED
5324 1347 TAD ACSAV2 /GET CODE TO BE PUNCHED
5325 6041 TSF
5326 5325 JMP .-1
5327 6046 TLS /PUNCH
5330 5716 JMP I PUN /EXIT
5331 1347 HISPED, TAD ACSAV2 /GET CODE TO BE PUNCHED
5332 6021 PSF
5333 5332 JMP .-1
5334 6026 PLS /PUNCH
5335 5716 JMP I PUN /EXIT

/
5336 0000 CKSM, 0
5337 0000 IA, 0
5340 0000 FA, 0
5341 7566 M212, -212
5342 0000 CTR1, 0
5343 0200 C200, 200
5344 0000 TEM1, 0
5345 0177 SL7, 177
5346 0077 SL6, 77
    
```

```

5347 0000 ACSAV2, 0
5350 5600 STRBUF, INBUF
5351 0000 RCNT13, 0
5352 0000 RCNT14, 0
/
5371 4340
5372 5106
5373 5105
5374 4213
5375 4342
5376 5104
5377 4337
5400 5400 PAGE
/
/ THE NEXT PAGE IS RESERVED FOR BINARY LOADER
/ STORAGE AREA WHILE THE DIAGNOSTIC IS BEING RUN,
/
5400 5400 SAVBIN, SAVBIN
/
5600 5600 PAGE
/
/ THE NEXT 40 OCTAL LOCATIONS ARE RESERVED
/ FOR THE BOOTSTRAP BUFFER
/
5600 5600 INBUF, INRUF
/
      S
    
```

0000
0100
0200
0300
0400
0500
0600
0700

1000
1100

1200
1300

1400
1500

1600
1700

2000
2100

2200
2300

2400
2500

2600
2700

3000
3100

3200
3300

3400
3500

3600
3700

4000
4100

4200
4300

4400
4500

4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

4200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4300	11111111	11111111	11111111	11111111	11111111	11111110	00000000	00000000	11111111
4400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4500	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	00111111
4600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4700	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	10111111
5000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5100	11111111	11111111	11111111	11111111	11111111	11111111	11111000	00000000	11111111
5200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5300	11111111	11111111	11111111	11111111	11111111	11111111	11110000	00000000	01111111
5400	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
5500	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
5600	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
5700	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

ACERR1	4790	HIPUN	5262	TCM1	5354
ACERR2	5033	HISPED	5331	TCM2	4203
ACERR3	5067	MLTLOC	4341	TCM3	4840
ACSAV1	4563	IA	5337	TCM4	4254
ACSAV2	5347	INBUF	5000	TCM5	4351
ADDHLT	4545	INHDMF	4274	TEXTG	4961
AMOUNT	4340	K0007	4745	TEXTM	4960
BCNT1	5077	K0010	4764	THSFLD	4662
BCNT10	5101	K0070	5104	TOPLD	5053
BCNT11	5102	K0207	4552	TSTOP	4745
BCNT12	5103	K0212	4551	TYPBUF	4421
BCNT13	5351	K0215	4550	TYPE	4905
BCNT14	5352	K0240	4554	WATHLT	4334
BCNT2	5100	K0260	4553	XBEGIN	4957
BCNT3	4766	K7600	5105	XCHK	4345
BCNT4	4767	K7774	4547	XINBUF	4362
BCNT5	4770	KCDF	4342	XLOD	4346
BCNT6	4564	KCHK	4760		
BCNT7	4565	KHLT	4343		
BCNT8	4566	KLOD	4761		
BCNT9	4567	LASTLC	4762		
BEGIN	4200	LENGTH	4555		
BELL	4514	LODCHK	4743		
BINFLD	5013	LODMEM	4643		
BINP	5276	M212	5341		
BOTMES	5107	MEMGO	4600		
BPUN	5223	NOVBIN	5000		
BUFBIN	5106	NOVBUF	4400		
C200	5343	NQL	7421		
CAF	4007	NEWFLD	4660		
CHKADD	4520	OCTEL	4464		
CHKHLT	5023	OPRFLD	5215		
CHKMEM	4666	PLOT	5264		
CHKTHS	4710	PREFLD	4213		
CKSM	5336	PUN	5316		
COMPAR	5043	PUNL	5243		
COMSTR	5053	RESTR3	4266		
CTR1	5342	RESTR4	4323		
ENDTST	4714	RPLBIN	5200		
ERHLT1	4753	SAFADD	4556		
ERHLT2	5036	SAVBIN	5400		
ERHLT3	5072	SETFLD	5015		
EXIT	4640	SL6	5346		
EXTCHK	4615	SL7	5345		
EXTSAV	4337	SPECHK	4725		
FA	5340	STRADD	4336		
FIRPAS	4344	STRBLK	4335		
FLDAMN	4763	STRBUF	5350		
FLDCHK	4742	STRCHK	4623		
FLJGO	4746	STRTUP	4534		
FRMFLD	4703	STRTYP	4441		
GOLEAD	5233	TBEGIN	4744		

ERRORS DETECTED: 0
 LINKS GENERATED: 63
 RUN-TIME: 7 SECONDS
 2K CORE USED