

7-12-72
digital

MAINDEC CHANGE

276 MCN NOTICE

Ø8-DHTMB-A-1
CHANGE NO.

Sheet 1 of 1

AUTHOR
Len Beyersdorfer
DATE **EXT.**
1/26/73 2537

PROGRAM DATE
12/4/72

PRODUCT LINE
8 Family
V-96C-06806

MAINDEC NUMBER
Ø8-DHTMB-A

PROGRAM NAME TM8-E Control Test Part 2 **DEVICE** TM8-E DECmagtape

ITEM

Ø. Release announcement--new release. This program continues testing the TM8-E control and verifies proper handling of all transports in the system.

1. PROBLEM: Test 3Ø (T3ØC,D,E) currently uses WRITE, WEOF, and RDCOMP to test the error detection logic for File Protect; however the software never sets the "GO" bit. The latest changes to the TM8-E require the "GO" bit set to a 1 for this error detection to work properly.

CORRECTION: Change the program as indicated:

<u>LOCATION</u>	<u>OLD CONTENTS</u>	<u>NEW CONTENTS</u>	<u>SYMBOLIC</u>
3664	4000	4100	WRITE + GO
3677	5000	5100	WEOF + GO
3712	3000	3100	RDCOMP + GO

OK
1/27/73

516-864-4002

IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DHTMB-A-D
PRODUCT NAME: TM8-E CONTROL TEST PART 2 **TM CONZ**
DATE CREATED: DECEMBER 4, 1972
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: LEONARD E. BEYERSDORFER

~~XXXXXXXXXX~~

COPYRIGHT (C) 1972

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS, 01754

**MAIN DEC CHANGE NOTICE
MAY BE REQUIRED FOR
PROGRAM TO OPERATE**



NOTE

THERE ARE SIX DIAGNOSTIC PROGRAMS ASSOCIATED WITH THE TM8-E DECMAGTAPE CONTROL AND ITS TRANSPORT SYSTEM. ALTHOUGH PHYSICALLY SEPARATE, THESE PROGRAMS MUST BE TREATED AS A LARGE INTEGRATED TEST, AND TO ENSURE PROPER SYSTEM OPERATION, THESE TESTS MUST BE EXECUTED IN THE ORDER DELINEATED BELOW.

IF A GIVEN TEST SHOULD FAIL AND IT APPEARS THAT A FIX HAS BEEN FOUND, ALL PROGRAMS MUST ONCE AGAIN BE RUN. ONLY WHEN ALL TESTS HAVE RUN WITHOUT ANY UNACCEPTABLE ERRORS CAN THE TM8-E SYSTEM BE CONSIDERED UP.

TM8-E DIAGNOSTIC PROGRAMS' ORDER OF EXECUTION

1. TM8-E CONTROL TEST PART 1 (MAINDEC-08-DHTMA)
2. TM8-E CONTROL TEST PART 2 (MAINDEC-08-DHTMB)
3. TM8-E DRIVE FUNCTION TIMER (MAINDEC-08-DHTMC)
4. TM8-E DATA RELIABILITY 9 TRACK (MAINDEC-08-DHTMD)
5. TM8-E DATA RELIABILITY 7 TRACK (MAINDEC-08-DHTME)
6. TM8-E RANDOM EXERCISER (MAINDEC-08-DHTMF)

TABLE OF CONTENTS

(1.	ABSTRACT
(2.	REQUIREMENTS
(2.1	HARDWARE
(2.2	MEMORY
(2.3	PRELIMINARY PROGRAMS
(3.	CONVENTIONS
(4.	PROGRAM LOADING PROCEDURE
(5.	PROGRAM STARTING PROCEDURE
(6.	STANDARD TEST PROCEDURE
(6.1	DRIVE SELECTION
(6.2	TEST PROCEDURE
(7.	PROGRAM CONTROLS
(7.1	PROGRAM MONITOR CONTROL VIA KEYBOARD COMMANDS
(8.1.1	TEST SELECTION COMMANDS
(7.1.2	TEST CONTINUATION COMMANDS
(7.1.3	TEST INTERRUPT COMMANDS
(7.1.4	MISCELLANEOUS COMMANDS
(7.2	SWITCH REGISTER CONTROL OPTIONS
(8.	ERRORS
(8.1	ERROR HALTS
(8.2	ERROR REPORTS
(8.3	STANDARD ERROR RECOVERY PROCEDURE
(8.3.1	SUBTEST LOOPS
(8.3.2	MINI SCOPE LOOPS
(9.	RESTRICTIONS
(10.	EXECUTION TIME
(11.	TEST ABSTRACTS
(12.	LISTING

1. ABSTRACT

THE TMB-E CONTROL TEST PART 2 IS AN INTEGRATED SERIES OF SUB-TESTS DESIGNED TO AID IN THE CHECKOUT AND MAINTENANCE OF THE TMB-E DECMAGTAPE CONTROL AND TU10 MASTER/SLAVE TRANSPORT SYSTEM. VERSATILITY OF USAGE IS AFFORDED THROUGH A MODEST KEYBOARD MONITOR AND SWITCH REGISTER CONTROL OPTIONS.

THIS PROGRAM CONSISTS OF 12 MAJOR TESTS (TEST 15 THROUGH TEST 30) EACH OF WHICH CONSISTS OF A NUMBER OF SUBTESTS DESIGNATED BY THE LETTERS A THROUGH Z. THESE TESTS PROGRESS FROM THE FUNCTIONS TESTED IN THE TMB-E CONTROL TEST PART 1 AND REQUIRE A TU10 TRANSPORT SYSTEM TO BE ON LINE FOR PROPER EXECUTION.

2. REQUIREMENTS

2.1 HARDWARE

- PDP-8/E, 8/M, 8/F
- TELETYPE OR COMPATIBLE DEVICE (TTY)
- TMB-E DECMAGTAPE CONTROL
- TU10 MASTER/SLAVE TRANSPORT SYSTEM

2.2 MEMORY

THIS PROGRAM REQUIRES 4K OF MEMORY AND MAY RESIDE IN ANY MEMORY FIELD.

2.3 PRELIMINARY PROGRAMS

- ALL PROCESSOR/MEMORY DIAGNOSTICS
- TMB-E CONTROL TEST PART 1

3. CONVENTIONS

IN THE DESCRIPTION OF ANY KEYBOARD COMMANDS GIVEN IN THIS MANUAL, THE BACK ARROW (←) CORRESPONDS TO DEPRESSING THE RETURN KEY, AND THE NUMBER SIGN (#) CORRESPONDS TO DEPRESSING THE LINE-FEED KEY. A LETTER PRECEDED BY AN UP ARROW (↑) SHOULD BE TYPED WITH THE "CTRL" KEY DEPRESSED. THE PROGRAM PRINTS A LEFT BRACKET (C) WHEN IT IS READY TO ACCEPT A KEYBOARD COMMAND.

WHEN THE PROGRAM PRINTS:

SETUP DRV N (N TRK)

THE OPERATOR MUST ENSURE THAT THE FOLLOWING STEPS ARE CARRIED OUT.

- A. MOUNT A SPARE REEL OF INDUSTRY COMPATIBLE MAGNETIC TAPE ON THAT DRIVE WITH THE FILE PROTECT RING IN PLACE (WRITE ENABLED).
- B. LOAD THE TAPE AND POSITION AT BOT.
- C. SET THE DRIVE SELECTOR SWITCH TO THE CORRECT DRIVE POSITION.
- D. SWITCH THAT DRIVE ON LINE.
- E. ENSURE THAT ALL OTHER DRIVES ARE OFF LINE.

4. PROGRAM LOADING PROCEDURE

LOAD THE PROGRAM INTO ANY DESIRED MEMORY FIELD USING THE STANDARD BINARY LOADER TECHNIQUE.

5. PROGRAM STARTING PROCEDURE

- A. LOAD ADDRESS 2200.
- B. LOAD THE EXTENDED ADDRESS WITH THE PROGRAM FIELD.
- C. CLEAR ALL SWITCHES.
- D. DEPRESS CLEAR, THEN CONTINUE.
- E. THE PROGRAM WILL PRINT ITS TITLE AND MAINDEC NUMBER, THEN ASK FOR DRIVE SELECTION. PRIOR TO MAKING DRIVE SELECTION, GO TO THE STANDARD TEST PROCEDURE IN PARAGRAPH 6.

NOTE: THE PROGRAM MAY BE RESTARTED AT ANY TIME AT ADDRESS 0201. IN THIS CASE THE PROGRAM BYPASSES ALL PROGRAM INITIALIZATION AND GOES DIRECTLY TO THE PROGRAM MONITOR TO ACCEPT KEYBOARD COMMANDS.

6. STANDARD TEST PROCEDURE

USE OF THE STANDARD TEST PROCEDURE ENSURES PROPER TM8-E/TU10 CHECKOUT, ANY ERROR OCCURRENCE RESULTS IN AN ERROR REPORT ON THE TTY AND IN A RETURN TO THE PROGRAM MONITOR. ALL OPERATIONAL TEST PROCEDURES AFFORDED BY PROGRAM MONITOR AND SWITCH REGISTER CONTROL ARE DESCRIBED IN PARAGRAPH 7. ERROR RECOVERY PROCEDURES AND RELATED INFORMATION ARE GIVEN IN PARAGRAPH 8.

6.1 DRIVE SELECTION

- TO SPECIFY THE DRIVE TO BE TESTED, CARRY OUT THE FOLLOWING STEPS.
- A. EITHER START THE PROGRAM AT 0200 AS DESCRIBED IN PARAGRAPH 5, OR WITH THE PROGRAM MONITOR IN CONTROL TYPE "I".
- B. RESPOND TO "DRIVE?" BY TYPING THE DRIVE NUMBER (0-7).
- C. RESPOND TO "7 OR 9 TRACK?" BY TYPING "7" OR "9".
- D. TAKE THE ACTION DESCRIBED IN PARAGRAPH 3 TO "SETUP DRIVEN (N TRK)".

6.2 TEST PROCEDURE

THE FOLLOWING STEPS ARE TO BE ACCOMPLISHED FOR EACH DRIVE IN THE SYSTEM UNDER TEST.

- A. SELECT THE DRIVE UNDER TEST AS DRIVE 0 AND ACCOMPLISH THE SETUP (REFER TO 6.1).
- B. SET THE SWITCH REGISTER (SR) TO 0000.
- C. TYPE "TA*" WHICH RESULTS IN THE EXECUTION OF TEST 15 THROUGH TEST 30, SEVERAL PASSES WILL BE MADE OF EACH TEST WITH THE EXCEPTION OF TEST 30. DURING TEST 30 FOLLOW THE DIRECTIONS AS PRINTED BY THE PROGRAM, WHEN EACH STEP HAS BEEN CARRIED OUT, TYPE "C*" TO CONTINUE IN THE TEST.
- D. WHEN ALL TESTS ARE COMPLETED, "SELECTED TESTS DONE" WILL BE PRINTED AND THE PROGRAM MONITOR WILL BE READY TO ACCEPT A NEW COMMAND.
- E. USING THE SAME DRIVE, SELECT THAT DRIVE AS DRIVE 1 (REFER TO 6.1).
- F. SET THE SR TO 4000.
- G. TYPE "TA*" WHICH RESULTS IN THE EXECUTION OF TEST 15 THROUGH TEST 27. ONLY ONE PASS WILL BE MADE OF EACH TEST.
- H. WHEN ALL TESTS ARE COMPLETED, "SELECTED TESTS DONE" WILL BE PRINTED AND THE PROGRAM MONITOR WILL BE READY TO ACCEPT A NEW COMMAND.
- I. USING THE SAME DRIVE, EXECUTE STEPS E THROUGH H WITH THAT DRIVE SELECTED AS DRIVE 2, 3, 4, 5, 6 AND 7.
- J. FOR EACH ADDITIONAL DRIVE ON THE SYSTEM REPEAT STEPS A THROUGH I.

7. PROGRAM CONTROLS

THE FOLLOWING SUBPARAGRAPHS DESCRIBE THOSE CONTROLS WHICH THE USER HAS OVER THE PROGRAM.

THERE ARE TWO MAIN SOURCES OF PROGRAM CONTROL: A) PROGRAM MONITOR CONTROL VIA KEYBOARD COMMANDS; AND B) SWITCH REGISTER CONTROL.

7.1 PROGRAM MONITOR CONTROL VIA KEYBOARD COMMANDS

THE PROGRAM MONITOR KEYBOARD COMMANDS ARE DESCRIBED BELOW, IF A COMMAND ERROR IS DETECTED, MONITOR PRINTS "?" AND THE COMMAND MUST BE RETYPED. COMMANDS MAY BE INPUT AFTER MONITOR HAS PRINTED A LEFT BRACKET (L). TEST INTERRUPT COMMANDS MAY BE TYPED AT ANY TIME.

THE PROGRAM MONITOR IS ENTERED UNDER ANY OF THE FOLLOWING CONDITIONS.

- A. AFTER PROGRAM STARTUP.
- B. AN ERROR OCCURS AND SR2=0.
- C. ALL SELECTED TESTS ARE DONE.
- D. A TEST INTERRUPT COMMAND (REFERENCE PARAGRAPH 7.1.3) IS TYPED BY THE USER.

7.1.1 TEST SELECTION COMMANDS

THE FOLLOWING COMMANDS ARE USED TO SELECT FROM ONE TO TWELVE TESTS FOR EXECUTION. REGARDLESS OF THE ORDER IN WHICH A TEST SELECTION IS MADE, THOSE TESTS ARE EXECUTED IN NUMERICAL ORDER. IF A GIVEN TEST IS SPECIFIED TWICE IN THE SAME SELECTION, IT WILL BE DELETED FROM THAT SELECTION.

COMMAND	RESULT
-----	-----
TA*	RUN ALL TESTS (TEST 15 THROUGH TEST 30)
TAX-	RUN ALL TESTS EXCEPT TEST 30 (MANUAL INTERVENTION TEST)
T15T22TNN+	RUN THE TEST(S) INDICATED; T15 THROUGH T30 IN OCTAL ARE THE VALID SELECTIONS. IF THE USER SELECTS OTHER THAN T15 THROUGH T30 THE PROGRAM WILL EXECUTE ONE OF THE VALID TESTS (UNDETERMINED).

7.1.1.2 TEST CONTINUATION COMMANDS

THE FOLLOWING COMMANDS ARE USED TO CONTINUE IN THE TEST SEQUENCE IF THAT SEQUENCE HAS BEEN INTERRUPTED BY AN ERROR WITH SR2=0, OR AS IN THE CASE OF TEST 30 WHERE THE USER MUST CARRY OUT SOME MANUAL INTERVENTION.

COMMAND RESULT

C+ CONTINUE IN THE TEST SEQUENCE. IF NO TESTS ARE SELECTED, MONITOR WILL PRINT "?"; IF THIS OCCURS A NEW TEST SELECTION MUST BE MADE.

EX+ CONTINUE WITH THE NEXT SUBTEST IF THE CURRENT SUBTEST IS FAILING AND SEVERAL SETS OF DATA REMAIN TO BE USED IN THAT SUBTEST, THIS COMMAND ENABLES THE USER TO EXIT A FAILING DATA SUBTEST AND CONTINUE WITH THE NEXT SUBTEST. AFTER THE CURRENT "EX+" COMMAND HAS BEEN UTILIZED FOR A FAILING DATA SUBTEST EXIT, IT IS NO LONGER EFFECTIVE. IF NO TESTS ARE SELECTED, MONITOR WILL PRINT "?". IN THIS CASE A NEW TEST SELECTION MUST BE MADE.

SNN+ MINI SCOPE LOOP. THIS COMMAND SHOULD BE USED ONLY WHEN AN ERROR HAS OCCURRED IN A SUBTEST WHICH TESTS TIME SEQUENCED OPERATIONS. "NN" IS THE BIT NUMBER (IN OCTAL) IN THE ERROR STATUS WORD (ERSTAT) WHICH INDICATES A FAILURE. SR5 MUST BE SET TO A 1 TO UTILIZE THIS FEATURE CORRECTLY. REFER TO PARAGRAPH 6 FOR MORE INFORMATION.

7.1.3 TEST INTERRUPT COMMANDS

THE FOLLOWING COMMANDS MAY BE USED TO INTERRUPT TESTING AND RETURN TO THE PROGRAM MONITOR. IN MOST CASES, ALL TEST SELECTIONS WILL BE DELETED AND THE USER MUST RESELECT THE TESTS TO BE EXECUTED. THE MOST COMMON USES OF THESE COMMANDS ARE TO REWIND THE SELECTED DRIVE, AND TO EXIT A MINI SCOPE LOOP.

COMMAND RESULT

ALTMODE KEY INTERRUPT TEST EXECUTION, RESPONDS TO "ALT" AND "ESC" KEYS.

*R INTERRUPT TEST EXECUTION, DELETE ALL TEST SELECTIONS, AND REWIND THE SELECTED DRIVE TO BOT.

*C INTERRUPT TEST EXECUTION, DELETE ALL TEST SELECTIONS, AND FORCE A DUMP OF ALL TMB-E REGISTERS ON THE TTY. (THE REGISTER DUMP FORMAT IS A MODIFIED ERROR REPORT FORMAT.)

7.1.4 MISCELLANEOUS COMMANDS

COMMAND RESULT

I+ INITIALIZE AND ALLOW NEW DRIVE SELECTION,

TR TRACE THE PROGRAM FLOW USING THE ERROR REPORT FORMAT (MODIFIED). THIS COMMAND IS A PREFIX TO ALL "TEST SELECTION" AND "TEST CONTINUATION" COMMANDS, AND RESULTS IN A MODIFIED ERROR REPORT FOR EACH SUBTEST THAT PASSES. TRACE REPORTS ARE CONTROLLED VIA SR BITS 3 AND 4 IN THE SAME MANNER AS ERROR REPORTS. "TR" REMAINS IN EFFECT UNTIL THE NEXT ENTRY TO THE PROGRAM MONITOR.

DMINNNN+ DUMP THE CONTENTS OF MEMORY LOCATION NNNN IN FIELD M ON THE TTY. LINE-FEED DUMPS THE CONTENTS OF THE NEXT LOCATION; CARRIAGE RETURN RETURNS TO THE PROGRAM MONITOR, TEST SELECTION IS UNAFFECTED.

7.2 SWITCH REGISTER CONTROL OPTIONS

SR BIT	STATE	FUNCTION
0	0	RUN ALL TESTS NORMALLY (SEVERAL PASSES OF EACH SELECTED TEST EXCEPT TEST 30), QUICK VERIFY (ONE PASS IS MADE OF EACH SELECTED TEST).
1	0	ENABLE TEST IN PROGRESS REPORTS, AS SOON AS A TEST IS STARTED THE TEST NUMBER (TESTNN) IS PRINTED ON THE TTY.
1	1	DISABLE TEST IN PROGRESS REPORTS.
2	0	RETURN TO PROGRAM MONITOR ON ANY ERROR OCCURRENCE.
1	1	RETURN TO PROGRAM MONITOR ONLY IF A FATAL ERROR HAS OCCURRED.
3	0	ENABLE ERROR AND TRACE REPORTS.
1	1	DISABLE
4	0	ERROR AND TRACE REPORTS INCLUDE APPLICABLE DATA ONLY.
1	1	FORCE ERROR AND TRACE REPORTS TO INCLUDE ALL POSSIBLE DATA.
5	0	DISABLE SUBTEST LOOP.
1	1	LOOP ON CURRENT SUBTEST, ALL VARIABLE DATA REMAINS CONSTANT, THIS SWITCH MUST ALSO BE SET TO ENTER A "MINI SCOPE LOOP".
6-9	0	INHIBIT LOOPS 6-9 RESPECTIVELY.
1	1	ENABLE LOOPS 6-9 RESPECTIVELY, LOOPS 6-9 ALLOW THE USER TO LOOP ON A SMALL GROUP OF SUBTESTS WITHIN THE CURRENT TEST THEREBY ENABLING INCREASING THE TESTING FREQUENCY OF A SMALL SET OF FUNCTIONS, THE LISTING MUST BE REFERENCED TO DETERMINE WHICH SUBTESTS ARE INCLUDED IN EACH LOOP, THE END OF EACH LOOP IS MARKED AS "*****LOOP N*****".
10	0	INHIBIT LOOP ON CURRENT TEST (TEST 15 - TEST 30).
1	1	ENABLE LOOP ON CURRENT TEST.
11	0	TERMINATE TESTING UPON COMPLETION OF ALL SELECTED TESTS AND DELETE TEST SELECTION.
1	1	RUN ALL SELECTED TESTS CONTINUOUSLY, AT THE COMPLETION OF EACH PASS THROUGH THE EXTREME TEST SELECTION, THE NUMBER OF PASSES EXECUTED (IN OCTAL) IS PRINTED (PASS NNNN).

12

8. ERRORS

8.1 ERROR HALTS

THERE ARE NO ERROR HALTS IN THIS PROGRAM. IF AN ERROR SHOULD OCCUR AND SR2=0, THE PROGRAM WILL STOP TESTING AND RETURN TO THE PROGRAM MONITOR TO AWAIT A USER COMMAND. REFER TO PARAGRAPH 7.1 FOR AVAILABLE COMMANDS.

8.2 ERROR REPORTS

WHEN SR4=0, ERROR REPORTS INCLUDE ONLY THAT INFORMATION WHICH APPLIES DIRECTLY TO THE ERROR. IF SR4=1, ALL POSSIBLE INFORMATION IS PRINTED WHETHER APPLICABLE OR NOT, AN EXAMPLE OF A MAXIMUM INFORMATION ERROR REPORT IS SHOWN BELOW.

```
*ER15D PC:0250 I1:6706 I2:6716 GD:0000 BD:7777 DD:7777
WC:1234 CA:2345 CM:0000 FS:0000 MS:0000 DB:0000 AC:0000
```

THE SYMBOLS USED IN THE ERROR REPORTS ARE DEFINED BELOW.

```
SYMBOL DEFINITION
-----
```

```
*ERNXX ERROR OCCURRED IN TEST NN, SUBTEST X. (IF NN=31, THE
ERROR OCCURRED OUTSIDE OF A FORMAL TEST,)
```

```
*FENXX FATAL ERROR
```

```
TR*NNX TRACE REPORT INDICATOR, NO ERROR OCCURRED BUT TRACE
IS ENABLED OR THE "C" COMMAND WAS TYPED BY THE USER,
```

```
PC:NNNN ADDRESS IN PROGRAM AT WHICH ERROR WAS DETECTED.
```

```
I1:NNNN OCTAL CODE FOR IOT1 IN A VARIABLE SUBTEST,
```

```
I2:NNNN OCTAL CODE FOR IOT2 IN A VARIABLE SUBTEST,
```

```
THE FOLLOWING THREE SYMBOLS ARE FURTHER DEFINED IN THE PROGRAM LISTING IN THE SUBTEST COMMENTS.
```

```
GD:NNNN GOOD TEST VALUE
```

```
BD:NNNN REAL TEST VALUE (BAD)
```

```
OD:NNNN PREVIOUS GOOD TEST VALUE (OLD)
```

THE FOLLOWING SYMBOLS INDICATE THE CONTENTS OF THE SPECIFIED REGISTER AT THE TIME THE ERROR WAS DETECTED,

```
WC:NNNN WORD COUNT REGISTER
```

```
CA:NNNN CURRENT ADDRESS REGISTER
```

```
CM:NNNN COMMAND REGISTER
```

```
FS:NNNN FUNCTION/STATUS REGISTER
```

```
MS:NNNN MAIN STATUS REGISTER
```

```
DB:NNNN DATA BUFFER REGISTER
```

```
AC:NNNN PROCESSOR ACCUMULATOR (VALID ONLY FOR ILLEGAL SKIP ERRORS.)
```

8.3 STANDARD ERROR RECOVERY PROCEDURE

THE STANDARD ERROR RECOVERY PROCEDURE ASSUMES THAT THE STANDARD TEST PROCEDURE IS BEING USED; THAT IS, THAT ALL ERROR OCCURRENCES RESULT IN AN ERROR REPORT AND IN A RETURN TO THE PROGRAM MONITOR.

WHEN AN ERROR OCCURS, USE THE FOLLOWING STEPS AS A GUIDE FOR RECOVERY REFERRING TO PARAGRAPH 8.2 FOR ERROR REPORT SYMBOL DEFINITIONS.

- A. REFERENCE THE POINT IN THE PROGRAM LISTING INDICATED BY THE "PC:" NUMBER.
- B. THE ERROR CODE IN THE LISTING SHOULD MATCH THE CODE IN THE ERROR REPORT (ERNX OR FENX).
- C. COMMENTED IMMEDIATELY BELOW THE ERROR CODE IN THE LISTING IS AN EXPLANATION OF THE NUMBERS PRINTED AFTER THE GD, BD AND OD SYMBOLS.
- D. IN THE IMMEDIATE VICINITY OF THE ERROR CODE AND RELATED SUBTEST A DESCRIPTION OF THE SUBTEST WILL BE FOUND.
- E. IF THE SYMBOLS "GD" AND "BD" ARE DEFINED AS THE GOOD AND REAL (BAD) VALUES OF ERSTAT (ERROR STATUS WORD), THAT SUBTEST TESTS TIME SEQUENCED OPERATIONS. IN THIS CASE, ANY BITS WHICH DIFFER BETWEEN GD AND BD INDICATE WHICH TIME SEQUENCED OPERATION(S) FAILED. THE TIME SEQUENCED OPERATIONS ARE COMMENTED WITH THE BIT POSITION TO WHICH THEY CORRESPOND AND EXACTLY WHAT IS BEING TESTED AT THAT TIME.
- F. IF THE ERROR IS FATAL (FENX), THERE IS NO STANDARD RECOVERY. THE CAUSE OF THE FAILURE MUST BE DETERMINED THROUGH STATIC MEANS.
- G. IF THE ERROR IS NON-FATAL (ERNX), THE USER MAY ELECT TO CONTINUE IN THE TEST SEQUENCE (C+), OR TO ENTER A SUBTEST OR MINI SCOPE LOOP AS DESCRIBED BELOW.
- H. IF THE FAILING SUBTEST DOES NOT TEST TIME SEQUENCED OPERATIONS, ENTER A SUBTEST LOOP AS DESCRIBED IN 8.3.1 BELOW.
- I. IF THE FAILING SUBTEST DOES TEST TIME SEQUENCED OPERATIONS, THE USER MAY ENTER A SUBTEST LOOP AS DESCRIBED IN 8.3.1 OR A MINI SCOPE LOOP AS DESCRIBED IN 8.3.2.

8.3.1 SUBTEST LOOPS

TO ENTER A SUBTEST LOOP, CARRY OUT THE FOLLOWING STEPS,

- A. SET SR BITS 2, 3 AND 5=1.
- B. TYPE "C←",

TO EXIT A SUBTEST LOOP, RESTORE THE SWITCHES TO THEIR NORMAL POSITION,

8.3.2 MINI SCOPE LOOPS

NOTE: MINI SCOPE LOOPS WILL WORK ONLY FOR SUBTESTS WHICH TEST TIME SEQUENCED OPERATIONS,

TO ENTER A MINI SCOPE LOOP, CARRY OUT THE FOLLOWING STEPS,

- A. SET SR BITS 2, 3 AND 5=1.
- B. TYPE "SNN←" WHERE NN IS THE BIT NUMBER (IN OCTAL) OF THE FIRST FAILING TIME SEQUENCED OPERATION, THE PROGRAM WILL LOOP CONTINUOUSLY FROM THE LAST "SET" COMMAND THROUGH THE TIME SEQUENCED OPERATION TEST SPECIFIED IN "SNN←". A "CLEAR TRANSPORT" (CLT) IS ISSUED AFTER THE COMPLETION OF EACH LOOP.

TO EXIT A MINI SCOPE LOOP, TYPE "←R" WHICH WILL INTERRUPT THE LOOP, DELETE ALL TEST SELECTIONS, REWIND THE SELECTED DRIVE, AND RETURN TO THE PROGRAM MONITOR,

9. RESTRICTIONS

ONLY ONE DRIVE MAY BE TESTED AT ONE TIME.
ALL OTHER DRIVES MUST BE OFF LINE.

10. EXECUTION TIME

EXECUTION TIME VARIES DEPENDENT UPON THE TYPE OF DRIVE BEING TESTED. ONE LONG PASS OF A FULL TEST SELECTION FOR A 9 TRACK DRIVE TAKES APPROXIMATELY 15 MINUTES.

11. TEST ABSTRACTS

TEST 15 CHECKS BASIC TMB-E CONTROL FUNCTIONS THAT DO REQUIRE A DRIVE TO BE ON LINE. NO TAPE MOTION, HOWEVER, IS INITIATED.

TEST 16 CHECKS TIME SEQUENCED CONDITIONS INCLUDING TAPE MOTION FOR WRITE, READ, READ-COMPARE, SPACE FORWARD, SPACE REVERSE AND REWIND FUNCTIONS. ALL OPERATIONS ARE DONE AT 800 BPI WITH 9 TRACK DRIVES IN CORE DUMP MODE.

TEST 17 VERIFIES THAT RECORDS WRITTEN IN ONE PARITY MODE, THEN READ IN THE OTHER MODE CAUSE LATERAL (VERTICAL) PARITY ERRORS.

TEST 20 VERIFIES THE ABILITY OF THE DRIVE TO WRITE A FILE MARK AND ALSO THE ABILITY TO RECOGNIZE A FILE MARK.

TEST 21 EXERCISES VARIOUS COMBINATIONS OF WRITE, WRITE FILE MARK AND SPACE REVERSE.

TEST 22 CONTINUES VARIOUS POSITIVE AND NEGATIVE TESTING SEQUENCES ON THE TM8-E CONTROL UTILIZING TAPE MOTION.

TEST 23 VERIFIES THE PROPER OPERATION OF THE CONTINUOUS MODE OF SINGLE DRIVE OPERATION UTILIZING VARIOUS COMBINATIONS OF DRIVE FUNCTIONS.

TEST 24 TESTS THE CHANGE DIRECTION MODE OF OPERATION UTILIZING VARIOUS COMBINATIONS OF FUNCTIONS WHICH CAUSE DIRECTION REVERSAL.

TEST 25 IS A BASIC TEST OF THE ABILITY OF THE TM8-E/TU10 TO PASS DATA CORRECTLY.

TEST 26 VERIFIES THE PROPER GENERATION OF THE CYCLIC REDUNDANCY CHECK CHARACTER (CRCC) BY COMPARING SIMULATED VALUES TO THOSE VALUES GENERATED BY THE HARDWARE AND WRITTEN ON TAPE. THIS TEST IS RUN ON 9 TRACK DRIVES ONLY.

TEST 27 VERIFIES THE UNIQUENESS OF CORE DUMP AND COMPATIBLE MODES OF OPERATION ON 9 TRACK DRIVES ONLY.

TEST 30 IS A MANUAL INTERVENTION TEST WHICH TESTS SUCH THINGS AS FILE PROTECT, THE OFF LINE FUNCTION AND THE PROPER EFFECT ON THE CONTROL BY THE PROCESSOR "INITIALIZE" SIGNAL.

12. LISTING (ATTACHED)

/TM8E CONTROL TEST PART 2 MAINDEC-08-DHTMB-A=L
/COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS,

.....
/ WARNING: THIS PROGRAM SHOULD NOT BE STOPPED UNLESS
/ MONITOR HAS TYPED "E".
.....

/THIS DIAGNOSTIC PROGRAM TESTS THE TM8E CONTROL AND THE TRANSPORT SYSTEM
/CONNECTED, AND MAY RESIDE IN ANY EXISTING MEMORY FIELD,

/BASIC STARTING PROCEDURE:
/LOAD THE PROGRAM INTO THE DESIRED MEMORY FIELD WITH THE BINARY LOADER;
/LOAD THE INSTRUCTION AND DATA FIELDS WITH THE PROGRAM FIELD,
/LOAD ADDRESS 220,
/CLEAR ALL SWITCHES,
/SET ANY SR OPTIONS DESIRED,
/DEPRESS "CLEAR", THEN "CONTINUE",
/ANSWER THE QUESTIONS ASKED BY THE PROGRAM, WHEN READY, SELECT
/TESTS USING THE MONITOR,

/SWITCH REGISTER CONTROL OPTIONS,

/SR BIT STATE	CONTROL
/0! 0	RUN ALL TESTS NORMALLY;
/ 1	RUN QUICK VERIFY,
/ 1	(1 PASS OF EACH SELECTED TEST);
/1! 0	TEST PROGRESS REPORTS,
/ 1	NO TEST PROGRESS REPORTS,
/2! 0	GO TO MONITOR ON ANY ERROR OCCURRENCE,
/ 1	GO TO MONITOR ONLY UPON FATAL ERROR,
/3! 0	PRINT ERROR MESSAGES AND IF "TR" IS PENDING ENABLE TRACE MESSAGES,
/ 1	PRINT ONLY FATAL ERROR MESSAGES,
/4! 0	ERROR MESSAGES DISPLAY NECESSARY DATA ONLY,
/ 1	ERROR MESSAGES DISPLAY ALL DATA WHETHER APPLICABLE OR NOT,

/NOTE: IF SR5=1 WHEN EXITING MONITOR TO RUN A NEW TEST SELECTION,
/THE PROGRAM MAY APPEAR TO HANG, MOMENTARILY RELEASE SP5 TO CONTINUE,

/5! 0	NO SUBTEST LOOP,
/ 1	LOOP ON SUBTEST. (SCOPE LOOP)
/6-9: 0	NO LOOP 6-9 RESPECTIVELY,

/ 1	LOOP 6-9 RESPECTIVELY. (LOOP ON SMALL GROUP OF SUBTESTS),
/10! 0	NO LOOP ON CURRENT TEST (TC1-T14),
/ 1	LOOP ON CURRENT TEST (TC1-T14),
/11! 0	NO LOOP ON COMPLETE TEST AS SELECTED,
/ 1	LOOP ON COMPLETE TEST AS SELECTED (OCTAL PASS NUMBER TYPED AT END OF EACH PASS),

/TTY KEYBOARD COMMAND CONTROL,

/1. THE FOLLOWING COMMANDS MAY BE ENTERED ANYTIME THAT THE PROGRAM IS RUNNING AND AUTOMATICALLY KILL ALL TESTS SELECTED, WHEN THE COMMAND ACTION IS COMPLETED, THE PROGRAM GOES TO MONITOR,
 /1.1

COMMAND	RESULT
/*C* (CONTROL C)	GENERATES CLT AND STOPS THE TRANSPORT AT PRESENT POSITION,
/*R* (CONTROL R)	CARRIES OUT /*C* THEN REWINDS TO BOT,

/2. IN ADDITION TO /*C* AND /*R* THE FOLLOWING COMMANDS MAY BE USED AS INDICATED,
 /NOTES! 1, THE FOLLOWING COMMANDS MAY BE ENTERED WHEN "C" IS TYPED.
 /2. /** INDICATES "RETURN" KEY; "RUROUT" ABORTS A COMMAND STRING PRIOR TO /**; "SPACE" MAY BE INPUT AT ANY POINT;
 /3. "ALTMODE" INTERRUPTS THE MAIN PROGRAM AND RETURNS TO MONITOR ONLY DURING TESTS NOT INVOLVING TAPE MOTION,

COMMAND	RESULT
/*T15116T17TNN*	RUN THE TESTS INDICATED, IF A TEST NUMBER IS INPUT 2 TIMES, THAT TEST IS NOT RUN,
/*TA*	RUN ALL TESTS,
/*TAX*	RUN ALL TESTS EXCEPT MANUAL INTERVENTION TESTS,
/*C*	CONTINUE FROM POINT OF INTERRUPTION, IF NO TESTS ARE PENDING OR A FATAL ERROR OCCURS, /*C* RESULTS IN SYNTAX ERROR,
/*EX*	IF A SUBTEST CONTAINS A LOOP TO CHECK OUT VARYING DATA PATTERNS, AND IF ERRORS KEEP OCCURRING PREVENTING CONTINUING ON QUICKLY WITH THE NEXT SUBTEST, DEPRESS "ALTMODE", THEN "EX*", THIS WILL EXIT THE CURRENT SUBTEST AFTER THE NEXT ERROR ALLOWING THE TEST TO CONTINUE AT THE NEXT SUBTEST. THE "EXITFL" WHICH IS SET BY "EX*" IS CLEARED AS SOON AS IT IS UTILIZED FOR AN ESCAPE. IF NO TESTS ARE PENDING OR A FATAL ERROR OCCURS, "EX*" RESULTS IN SYNTAX ERROR,
/*TR*	TRACE USING ERROR MESSAGE FORMAT WHETHER ERROR OCCURS OR NOT, /*TR* MAY PRECEED /*TA*, /*TNN*, /*C*, OR /*EX* COMMANDS, TRACE REMAINS IN EFFECT UNTIL "C" IS DISPLAYED AGAIN, IT RESULTS IN A MODIFIED ERROR MESSAGE BEING DISPLAYED FOR EACH SUBTEST RUN IF THAT TEST PASSES, THESE PRINTOUTS ARE CONTROLLED BY THE SAME SR OPTIONS AS ERROR MESSAGES,
/*I*	REPEAT OPENING DIALOGUE TO ALLOW USER TO SELECT NEW DRIVE, ETC., THEN REINITIALIZE,
/*SNN*	ENABLE SCOPE LOOP "N" WHERE "N" IS THE BIT NUMBER IN OCTAL IN THE ERROR WORD WHICH IS FAILING, THE PROCEDURE TO FOLLOW IS: WHEN AN ERROR OCCURS ENABLE RETURN TO MONITOR, SET SR0 TO A 1 (SUBTEST LOOP) AND TYPE "S(THEN THE BIT NUMBER)*", THE RESULT WILL BE A LOOP STARTING AT THE BEGINNING OF THE SUBTEST THROUGH THE BIT N CHECK, THE ONLY WAY TO EXIT FROM A LOOP OF THIS TYPE IS BY /*C* OR /*R*, A SYNTAX ERROR WILL OCCUR WILL OCCUR

/ IF NO SUBTESTS ARE PENDING;
 /*DN:MMMM* DISPLAY THE CONTENTS OF MEMORY LOCATION MMMM IN FIELD N, IF A LINE FEED IS TYPED THE NEXT SEQUENTIAL MEMORY LOCATION IS TYPED, WHEN THE ADDRESS OVERFLOWS THE NEXT MEMORY FIELD WILL BE REFERENCED AND THE CONTENTS DISPLAYED, A CARRIAGE RETURN ALLOWS THE INPUT OF A NEW COMMAND,

/ERRORS:

/IF AN ERROR OCCURS, REFER TO THE ERROR PRINTOUT AND THE PROGRAM LISTING FOR THE SUBTEST FAILURE DESCRIPTION,

/ERROR PRINTOUTS:

/HEADER	DEFINITION
/-----	-----
/•ERNXX	ERROR OCCURRED IN TEST NN SUBTEST X(TNXX),
/•FENXX	SAME AS ABOVE EXCEPT FATAL ERROR,
/TR•NXX	TRACE PRINTOUT (FROM "TR" MONITOR COMMAND),
/PC•NXXX	ADDRESS IN PROGRAM AT WHICH ERROR OCCURRED,
/I1•NXXX	OCTAL CODE FOR IOT1,
/I2•NXXX	OCTAL CODE FOR IOT2,
/GD•NXXX	GOOD TEST VALUE, DESCRIBED IN SUBTEST LISTING,
/BD•NXXX	BAD OR ACTUAL TEST VALUE, DESCRIBED IN SUBTEST LISTING;
/DD•NXXX	USUALLY PREVIOUS GOOD, DESCRIBED IN SUBTEST LISTING,
/THE FOLLOWING	DISPLAY THE CONTENTS OF THE INDICATED REGISTER
/WC•NXXX	WORD COUNT,
/CA•NXXX	CURRENT ADDRESS,
/CH•NXXX	COMMAND,
/FS•NXXX	FUNCTION - STATUS,
/MS•NXXX	MAIN STATUS,
/DB•NXXX	DATA BUFFER,
/AC•NXXX	ACCUMULATOR (USED FOR SKIP ERRORS ONLY),

/PROGRAM DIRECTORY:

7600 *7600

/REFER TO THE ASSEMBLY LISTING JUST TO THE LEFT OF THE MNEMONIC FOR THE ABSOLUTE STARTING ADDRESS OF THAT PARTICULAR PROGRAM SEGMENT. (THIS IS FOR REFERENCE ONLY, DO NOT USE THESE STARTING ADDRESSES.)

/MNEMONIC - PROGRAM SEGMENT,

7600	0203	TEST15	/TEST SECTION;
7601	0600	TEST16	/BASIC CONTROL TESTS NOT INVOLVING TAPE MOTION,
7602	1200	TEST17	/BASIC TAPE MOTION TEST,
7603	1427	TEST18	/LATERAL PARITY TEST,
7604	1600	TEST19	/HEOF TEST,
7605	2000	TEST21	/COMBINED FUNCTION TEST,
7606	2400	TEST22	/ADDITIONAL TESTS FOR IF AND EF,
7607	2600	TEST23	/CONTINUE MODE TEST,
7610	3000	TEST24	/CHANGE DIRECTION TEST,
7611	3200	TEST25	/BASIC DATA TEST,
7612	3400	TEST26	/CMCC GENERATION TEST (9 TRACK ONLY),
7613	3600	TEST27	/CORE DUMP MODE TEST (9 TRACK ONLY),
		TEST30	/MANUAL INTERVENTION TEST,
7614	4400	EXEC	/PROGRAM CONTROL;
7615	4600	MONIT	/EXECUTIVE,
7616	5600	INTSEV	/KEYBOARD MONITOR,
			/INTERRUPT SERVICE,
7617	5200	ERRORS	/ERROR HANDLERS;
7620	6473	COMP	/ERROR HANDLING ROUTINE,
7621	6451	FE31A	/ERROR DETECTOR,
			/ERROR TABLE,
7622	4016	SLWCH	/UTILITIES;
			/TM8E IOT SUBROUTINES,
			/IF THE DEVICE CODES ARE NOT 70-72 IN YOUR SYSTEM, CHANGE
			/ONLY THE IOT CODES IN EACH OF THESE 10 SUBROUTINES,
7623	7413	WAIT1R	/WAIT ROUTINES,
7624	4237	TSKEP	/LITTLE TEST ROUTINES,
7625	6270	SETUP	/SET UP ROUTINES INCLUDING CONTNU MODE;
7626	2737	AMGR	/TEXT HANDLERS,
7627	6105	EDIT	
7630	6665	ERMSG	/DATA SECTION;
7631	6776	MSG1	/ERROR MESSAGE INSERT BLOCK,
		ENPUNCH	/MESSAGE TEXTS,

```

0000 *0000
0000 0000 0
0001 5001 JMP 1
0002 0002 2
0003 0003 3
0004 3541 SRBSWP, SRBSW /BSW
0005 3554 SRMQLP, SRMQL /MQL
0006 3557 SRMQAP, SRMQA /MQA
0007 4400 EXECOP, EXEC

0010 *0010
/AUTO INDEXERS:
0010 0000 AUTO10, 0
0011 0000 AUTO11, 0
0012 0000 AUTO12, 0
0013 0000 AUTO13, 0
0014 0000 AUTO14, 0
0015 0000 AUTO15, 0
0016 0000 AUTO16, 0
0017 0000 AUTO17, 0

0020 *0020
/WIDE USAGE NUMERIC CONSTANTS, (POSITIVE)
0020 0002 K2, 2
0021 0004 K4, 4
0022 0007 K7, 7
0023 0010 K10, 10
0024 0014 K14, 14
0025 0017 K17, 17
0026 0040 K40, 40
0027 0077 K77, 77
0030 0100 K100, 100
0031 0200 K200, 200
0032 0215 K215, 215
0033 0400 K400, 400
0034 1000 K1000, 1000
0035 4040 K4040, 4040

/WIDE USAGE NUMERIC CONSTANTS, (NEGATIVE)
0036 7776 M2, -2
0037 7774 M4, -4
0040 7773 M5, -5
0041 7764 M14, -14
0042 7740 M40, -40
0043 7735 M43, -43
0044 7575 M203, -203
0045 7556 M222, -222
0046 7520 M240, -240
0047 7510 M270, -270

/WIDE USAGE INSTRUCTIONAL CONSTANTS:
0050 6201 KCDF, CDF

/TRIAL BUFFER AREA.
    
```

```

0051 7342 XBUF, T29BUF
0052 7343 T29BUF+1
0053 7344 T29BUF+2
0054 7345 T29BUF+3
0055 7346 T29BUF+4
0056 7347 T29BUF+5

/WIDE USAGE POINTERS
0057 4375 SCDFP, SCDF
0060 2737 AMG8EP, AMG8E
0061 4600 MON1P, MON1T
0062 6105 ED1P, ED1T
0063 6132 EDTEMP, EDTEM
0064 2571 BELLP, BELL
0065 6167 CRLFP, CRLF
0066 6665 ERMSGP, ERMSG
0067 2553 TYPEP, TYPE
0070 6140 CNTCP, CNTC
0071 6377 CNTRP, CNTR

/TMBE IOT SUBROUTINE POINTERS,
0072 4016 SLWCRP, SLWCR /LWCR
0073 4022 SLCARP, SLCAR /LCAR
0074 4026 SLCMRP, SLCMR /LCMR
0075 4032 SLFGRP, SLFGR /LFG
0076 4036 SLDBRP, SLDBR /LDBR
0077 4042 SRWCRP, SRWCR /RWCR
0100 4046 SRCARP, SRCAR /RCAR
0101 4052 SRMSHP, SRMSR /RMSR
0102 4056 SRCMRP, SRCMR /RCMR
0103 4062 SRFSHP, SRFSR /RFSR
0104 4066 SRDBRP, SRDBR /RDBR
0105 4072 SSKEFP, SSKEF /SKEF
0106 4077 SSKCBP, SSKCB /SKCB
0107 4104 SSK1DP, SSK1D /SK1D
0110 4111 SSKTRP, SSKTR /SKTR
0111 4116 SCLFP, SCLF /CLF
0112 4122 SCLTP, SCLT /CLT
0113 4126 SSDLEP, SSDLE /SDLE
0114 4132 SSBRRP, SSBRR /SBRM

/PSEUDO MNEMONIC SUBROUTINE POINTERS,
0115 4332 CLR1P, CLR1 /CLEAN1
0116 4353 CLR4P, CLR4 /CLEAN4
0117 4357 CLR5P, CLR5 /CLEAN5
0120 1141 LAS5P, LAS5 /LOOP5
0121 1147 LAS6P, LAS6 /LOOP6
0122 1155 LAS7P, LAS7 /LOOP7
0123 1163 LAS8P, LAS8 /LOOP8
0124 1171 LAS9P, LAS9 /LOOP9
0125 2473 COMPP, COMP /COMPAN
0126 9200 ERRORP, ERRORS /ERRORM
0127 1742 LDPTP, LDPT /LOADPT
0130 6200 SET1P, SET1 /SET1
    
```

```

0131 6207 SET2RP, SET2R /SET2
0132 6216 SET3RP, SET3R /SET3
0133 6224 SET4RP, SET4R /SET4
0134 6331 CONTNP, CONTNR /CONTNU
0135 4200 TSKEFP, TSKEFR /TSKEF
0136 4206 TSKTDP, TSKTDR /TSKTD
0137 4214 TSKCBP, TSKCBR /TSKCB
0140 4222 TSKTRP, TSKTRR /TSKTR
0141 4230 TMRP, TMRR /TMR
0142 4243 TFRP, TFRR /TFR
0143 4256 TWCP, TWCR /TWC
0144 7413 WAIT1P, WAIT1R /WAIT1
0145 7420 WAIT2P, WAIT2R /WAIT2

/ WIDE USAGE TEMPORARIES,
2146 0000 GOOD, 0 /GOOD DATA;
0147 0000 BAD, 0 /BAD DATA;
0150 0000 OLD, 0 /OLD DATA;
0151 0000 TXXTM1, 0 /THREE TEMPORARIES FOR
0152 0000 TXXTM2, 0 /FORMAL TEST USAGE ONLY,
0153 0000 TXXTM3, 0
0154 0000 SAVEAC, 0 /AC SAVED;
0155 0000 SAVED, 0 /LINK SAVED;
0156 0000 PRGFLO, 0 /MEMORY FIELD WHICH PROGRAM OCCUPIES,
0157 0000 SLKNST, 0 /BIT NUMBER FOR SCOPE LOOP;
0160 0000 SLADDR, 0 /SUBTEST SET UP ADDRESS FOR SCOPE LOOP;
0161 0000 TRKY, 0 /SET IF SELECTED DRIVE IS X TRACK;
0162 0000 ERSTAT, 0 /CONTAINS CONDITION BITS FROM LITTLE TESTS;
0163 0000 KCMD, 0 /CORRECTION TO CH CONSTANT FOR DRIVE SELECTED;
0164 0000 ERTAL, 0 /CURRENT ERSTAT BIT POSITION;
0165 0000 ALTEA, 0 /SET IF ALTMODF INTERRUPT ENABLED;
0166 0000 IOT1, 0 /IOT'S IN USE IN GENERALIZED TESTS;
0167 0000 IOT2, 0
0170 0000 ACLOC, 0 /CONTENTS OF AC FOR ILLEGAL SKIP,
0171 0000 EXPEOF, 0 /0=EOF, -1=NO EOF

/ WIDE USAGE SOFTWARE FLAGS AND INDICATORS,
0172 0000 TRACE, 0 /SET IF "TR" PENDING,
0173 0000 TIOFLG, 0 /SET IF TIO FLAG GETS SET,
0174 0000 EXITFL, 0 /SET IF AUTO-EXIT "EX=M",
0175 0000 ACTFLG, 0 /SET IF TEST IN PROGRESS,
0176 0000 TSTAT, 0 /BIT N SET MEANS RUN TEST
0177 0000 TSTNUM, 0 /N*15 (IN OCTAL),
/CURRENT TEST NUMBER,

```

/THIS IS WHERE THE PROGRAM STARTS,
/-----

```

0200 *000
0200 5407 START, JMP I EXEC /START HERE, GO TO EXEC,
0201 7120 STL
0202 5407 JMP I EXEC

```

5

/TEST SECTION:

/THE TEST SECTION IS COMPOSED OF 12 TESTS (T15-T30) EACH OF WHICH IS
/MADE UP OF SUBTESTS DESIGNATED BY THE LETTERS A-Z;

/THE SUBTEST FORMAT IS DESCRIBED BELOW,

```

/T15Z, INST /HOUSEKEEP TO SET UP LOOPS AND SPECIAL LOCATIONS
/ INST /SUCH AS "GOOD" AND "OLD",
/ RUN TEST
/ AT THE END OF THE TEST, AC SHOULD CONTAIN SOME PREDETERMINED
/ VALUE,
/ COMPAR /COMPARE THE VALUE IN AC WITH THE VALUE IN "GOOD",
/ /IF EQUAL, CHECK FOR SUBTEST LOOP, IF NOT EQUAL, GO TO
/ /ERROR HANDLER AND TAKE THE REQUIRED
/ /ACTION,
/ ER15Z /ADDRESS OF ERROR STATUS WORDS;
/ JMP T15Z /SUBTEST LOOP JUMP
/ INST /POINT OF CONTINUATION OR GET MORE DATA, IF THIS IS
/ /A DATA SUBTEST, "EX-" ENABLES SKIPPING THIS INSTRUCTION,
    
```

/SOME SUBTESTS CHECK MANY CONDITIONS IN A TIME ORDERED SEQUENCE,
/TO FACILITATE SUCH A TESTING PROCEDURE, A DICTIONARY OF LITTLE TEST
/ROUTINES HAS BEEN INCLUDED, IN GENERAL, THESE SUBTESTS MOVE TAPE AND HAVE
/THE FOLLOWING FORMAT,

```

/ SETN /CALL TO ONE OF FOUR SET UP ROUTINES;
/ NNNN /CONSTANT TO STORE IN GOOD,
/ NNNN /WC CONSTANT,
/ NNNN /CA CONSTANT,
/ NNNN /DATA TO BE STORED IN CA+1,
/ NNNN /CM CONSTANT(USUALLY INCLUSIVELY OR'ED WITH KCMD),
/ NNNN /FR CONSTANT(USUALLY INCLUSIVELY OR'ED WITH KFUNC);
/ TMS /N:M /LITTLE TEST ROUTINES FOLLOW, THE METHOD HERE
/ IS TO PRESET GOOD WITH THE EXPECTED VALUE OF ERSTAT AT THE END OF
/ THE SUBTEST, THE FIRST LITTLE TEST ROUTINE STORES ITS RESULT IN
/ BIT 0 OF ERSTAT; THE 2ND IN BIT 1, ETC., UP TO 12 BITS, CONSULT
/ THE LITTLE TEST ROUTINES FOR INFORMATION CONCERNING BIT SETTING AND CLEARING,
/ THEN ERSTAT AND GOOD ARE COMPARED, EACH LITTLE TEST IS COMMENTED
/ "N:M" WHICH INDICATES THE ERSTAT BIT NUMBR(N) AND EXPECTED BIT
/ SETTING(M=1 OR 0),
/ TAD ERSTAT
/ COMPAR
/ ETC,
    
```

/TEST 15. BASIC CONTROL TEST NOT INVOLVING TAPE MOTION,

/TESTS ARE RUN ON DRIVES THAT SHOULD BE OFF LINE
/THEN SOME MORE CONTROL TESTS ARE MADE ON THE SELECTED DRIVE
/WITHOUT ISSUING "GO", (THE AVAILABILITY OF BOT ENABLES
/CHECKING OUT A FEW MORE GATES IN THE CONTROL,)

/LOOPS 6-9:
/LOOP 6 CYCLES ON THE OFFLINE DRIVES TESTS (T15A-T15C),
/LOOP 7 CYCLES ON T15D-T15H,
/LOOP 8 CYCLES ON T15I-T15J,

0203 0000 TEST15, 0

/VERIFY SKTR DOES NOT SKIP WHEN AN OFF LINE
/DRIVE IS SELECTED.

```

0204 7240 T15A, STA
0205 3146 DCA GOOD
0206 4767 JMS I T15X1 /SELECT AN OFF LINE DRIVE
0207 4510 SKTR /SHOULD NOT SKIP,
0210 7240 STA
0211 4525 COMPAR
0212 6463 ER15A
0213 5770 JMP I T15X2 /SUBTEST LOOP
0214 4771 JMS I T15X3 /SELECT ANOTHER DRIVE,
    
```

/VERIFY NO BOT BIT ON ANY OFF LINE DRIVE,

```

0215 3146 T15B, DCA GOOD
0216 4767 JMS I T15X1 /SELECT AN OFF LINE DRIVE;
0217 4501 RMSR /BOT SHOULD NOT BE SET,
0220 0034 AND K1000
0221 4525 COMPAR
0222 6465 ER15B
    
```

/GO=GOOD BOT BIT; BD=REAL BOT BIT

```

0223 5770 JMP I T15X2 /SUBTEST LOOP,
0224 4771 JMS I T15X3 /SELECT ANOTHER OFF LINE DRIVE,
    
```

/VERIFY SELECT REMOTE BIT SET FOR OFF LINE DRIVES,

```

0225 1033 T15C, TAD K400
0226 3146 DCA GOOD
0227 4767 JMS I T15X1 /SELECT AN OFF LINE DRIVE
0230 4501 RMSR /SELECT REMOTE SHOULD BE SET
0231 0033 AND K400
0232 4525 COMPAR
0233 6467 ER15C
    
```

/GO=GOOD SELECT REMOTE BIT; BD=REAL SELECT REMOTE BIT

```

0234 5770 JMP I T15X2 /SUBTEST LOOP
0235 4771 JMS I T15X3 /SELECT ANOTHER DRIVE
    
```

```

0236 4521 LOOP6
0237 5204 JMP T15A /*****LOOP 6****
    
```

/VERIFY SKTR AND SKCB SKIP FOR ON LINE DRIVE

```

0240 3146 T15D, DCA GOOD
0241 4772 JMS I T15X4 /GET AN IOT
0242 0445 T15LS1
0243 7776 -2
0244 3245 DCA ,*1 /SKTR OR SKCR (11) SHOULD SKIP
0245 0000 0
0246 7240 STA
0247 4525 COMPAR
0250 6471 ER15D
/11=IOT CURRENTLY BEING TESTED
0251 5773 JMP I T15X5 /SUBTEST LOOP
0252 4774 JMS I T15X6 /GET ANOTHER IOT

/VERIFY BOT BIT IS ONLY MAIN STATUS BIT SET FOR ON LINE DRIVE
0253 1034 T15E, TAD K1000
0254 3146 DCA GOOD
0255 4515 CLEAR1 /CLEAR ALL AND SELECT ON LINE DRIVE
0256 4501 RCSR /BOT BIT IS ONLY BIT THAT SHOULD BE SET
0257 4525 COMPAR
0258 6473 ER15E
/GD=GOOD MAIN STATUS; BD=REAL MAIN STATUS
0261 5255 JMP T15E+2 /SUBTEST LOOP

/VERIFY 9 TRACK BIT SET FOR 9 TRACK DRIVE; CLEAR FOR 7 TRACK
/DRIVE! ON LINE DRIVE SELECTED.
0262 1161 T15F, TAD TRK9
0263 7640 SEA CLA
0264 1026 TAD K40
0265 3146 DCA GOOD
0266 4515 CLEAR1 /CLEAR AND SELECT ON LINE DRIVE
0267 4502 RCSR /ENSURE DENSITY 3 IS SELECTED
0270 7001 IAC
0271 4474 LCSR
0272 4503 RFSR /FUNCTION/STATUS =40 FOR 9, 00 FOR 7
0273 0027 AND K77
0274 4525 COMPAR
0275 6473 ER15F
/GD=GOOD FUNCTION/STATUS; RD=REAL FUNCTION/STATUS
0276 5266 JMP T15F+4 /SUBTEST LOOP

/VERIFY SKEF, SKTD, SBRM, SDLE AND CLF DO NOT SKIP WHEN
/ON LINE DRIVE IS SELECTED.
0277 7240 T15G, STA
0300 3146 DCA GOOD
0301 4772 JMS I T15X4 /GET AN IOT
0302 0447 T15LS1+2
0303 7773 -5
0304 3305 DCA ,*1 /IOT IN "11" SHOULD NOT SKIP
0305 0000 0
0306 7240 STA
0307 4525 COMPAR
0310 6477 ER15G
/11=IOT BEING TESTED
0311 5773 JMP I T15X5 /SUBTEST LOOP

```

```

0312 4774 JMS I T15X6 /GET ANOTHER IOT

/VERIFY CLF CLEANS NON-STATUS REGISTERS SINCE TAPE UNIT IS READY
/CURRENT ADDRESS USED.
0313 3146 T15H, DCA GOOD
0314 4515 CLEAR1 /CLEAR ALL AND SELECT ON LINE DRIVE
0315 7240 STA
0316 4473 LCAR /SET UP CURRENT ADDRESS
0317 4511 CLF /SHOULD CLEAR CURRENT ADDRESS
0320 4500 RCSR
0321 4525 COMPAR
0322 6501 ER15H
/GD=GOOD CURRENT ADDRESS
0323 5314 JMP T15H+1 /SUBTEST LOOP

0324 4522 LOOP7
0325 5236 JMP T15D-2 /****LOOP 7****

/VERIFY SPCREV AT BOT SETS IF AND EF
0326 4532 T15I, SET3
0327 4000 /GD
0330 0000 0 /NC
0331 0000 0 /CM
0332 7100 SPCREV+GO/FS
0333 4541 TMS /011 EF AND IF SHOULD BE SET
0334 4001 4001 /110 SKTD SHOULD NOT SKIP
0335 4536 TSKTD
0336 1162 TAD ERSTAT
0337 4525 COMPAR
0340 6503 ER15I
/GD=GOOD ERSTAT; BD=REAL ERSTAT
0341 5326 JMP T15I /SUBTEST LOOP

/VERIFY BOT AND A FUNCTION OTHER THAN SPCREV DOES NOT SET IF OR EF
T15J, JMS I T15X4 /GET A FUNCTION
0342 4772 T15LS2
0343 0457 -7
0344 7771 DCA ,*5
0345 3352 SET3
0346 4532 0 /GD
0347 0000 0 /NC
0350 0000 0 /CM
0351 0000 0 /FS /FUNCTION SUBSTITUTED
0352 0000 0 THERE SHOULD BE NO ERROR
0353 4535 TSKEF /010 IF SHOULD NOT BE SET
0354 4541 TMS /110
0355 0001 1 /210 MTF SHOULD NOT BE SET
0356 4536 TSKTD
0357 1162 TAD ERSTAT
0360 4525 COMPAR
0361 6505 ER15J
/GD=GOOD ERSTAT; BD=REAL ERSTAT
0362 5773 JMP I T15X5 /SUBTEST LOOP
0363 4774 JMS I T15X6 /GET ANOTHER FUNCTION

```

```

0364 4523          LOOP8
0365 5324          JMP    T151-2      /*****LOOP 8*****/

2366 5603          JMP    I TEST15
0367 2400          T15X1, T15EX1
0370 2413          T15X2, T15EX2
0371 2417          T15X3, T15EX3
0372 2423          T15X4, T15EX4
0373 2434          T15X5, T15EX5
0374 2437          T15X6, T15EX6

0400              PAGE
0400 0000          T15EX1, 0
0401 1163          TAD    KCMD          /GET CM CONSTANT WITH CORRECT
0402 4405          MQL          /DRIVE NUMBER, ETC.;
0403 1272          TAD    K3          /ENSURE BITS 10 AND 11 ARE SET TO AVOID
0404 4406          MQA          /YCH ERRORS, AND SAVE;
0405 3151          DCA    TXXTM1
0406 1270          TAD    M7
0407 3153          DCA    TXXTM3          /THEN SET UP FOR 7 OFFLINE DRIVES
0410 1151          TAD    TXXTM1          /GET CM CONSTANT AND UPDATE
0411 1034          TAD    K1000          /DRIVE NUMBER BY 1 DISREGARDING
0412 3151          DCA    TXXTM1          /OVERFLOW, AND SAVE;
0413 4515          T15EX2, CLEAR1          /GENERATE "INITIALIZE";
0414 1151          TAD    TXXTM1          /LOAD CM WITH UNSELECTED DRIVE,
0415 4474          LCHR          /HUN TEST;
0416 5000          JMP    I T15EX1          /ENTER AT END OF TEST;
0417 0000          T15EX3, 0          /ANY MORE DRIVES ON THIS MASTER?
0420 2153          ISZ   TXXTM3          /YES, RETURN AND GET NEXT DRIVE,
0421 5210          JMP    T15EX2-3          /END OF SUBTEST, EXIT;
0422 5617          JMP    I T15EX3

0423 0000          T15EX4, 0          /
0424 1623          TAD    I T15EX4          /GET LIST STARTER AND
0425 3151          DCA    TXXTM1          /SAVE;
0426 2223          ISZ   T15EX4
0427 1623          TAD    I T15EX4          /GET NUMBER OF ITEMS AND
0430 3152          DCA    TXXTM2          /SAVE;
0431 2223          ISZ   T15EX4          /UPDATE TO RETURN,
0432 1551          TAD    I TXXTM1          /GET A LIST ITEM;
0433 3166          DCA    IOT1          /STASH FOR AN "I11" PRINTOUT;
0434 4515          T15EX5, CLEAR1          /GENERATE "INITIALIZE";
0435 1166          TAD    IOT1          /GET ITEM AND PUT IN AC;
0436 5623          JMP    I T15EX4          /GO TO SUBTEST;
0437 0000          T15EX6, 0          /ENTER AT END OF SUBTEST;
0440 2151          ISZ   TXXTM1          /UPDATE LIST POINTER;
0441 2152          ISZ   TXXTM2          /ALL ITEMS USED?
0442 5232          JMP    T15EX5-2          /NO, GET NEXT ITEM;
0443 5637          JMP    I T15EX6          /YES, EXIT;
0444 0000          T15TM1, 0
0445 4510          T15LS1, SKTR
0446 4506          SKCB
0447 4505          SKCF
0450 4507          SKTD
0451 4514          SBRM

```

```

2452 4513          SOLE
2453 4511          CLF
2454 4474          LCHR
2455 4475          LFGR
2456 4476          LDBR
2457 0000          T15LS2, OFFLIN
2460 1000          REWIND
2461 2000          READ
2462 3000          RDCOMP
2463 4000          WRITE
2464 5000          WEOF
2465 6000          SPCFWD
2466 2000          READ
2467 3000          RDCOMP
2470 7771          M7,    =7
2471 7770          M10,   =10
2472 3003          K3,    3

```

```

/COMPARE AND ERROR DETECTION ROUTINE.

/ENTER WITH "BAD" IN AC.
/USE THE FOLLOWING CALLING SEQUENCE.

/COMPAR          CALL COMPARE,
/ERNXX          ADDRESS OF ERROR, (ERROR CODE),
/JMP ADDR      LOOP JUMP
/.....CONTINUATION      TEST CONTINUE.

/COMPARE DOES THE FOLLOWING:
/1. COMPARE "GOOD" AND "BAD".
/2. IF NOT EQUAL APPEARS TO GO TO ERRORS FROM POINT IN TEST
/   WHERE ERROR OCCURRED.
/3. IF EQUAL, CHECKS CURRENT SUBTEST LOOP.

0473 0000  COMP,  0
0474 3147  DCA   BAD           /SAVE AC IN BAD.
0475 1146  TAD   GOOD          /COMPARE GOOD AND BAD,
0476 7041  CIA
0477 1147  TAD
0500 7650  SNA  CLA  BAD
0521 5305  JMP           ,++
0502 1273  TAD   COMP
0503 3526  DCA  I  ERRORP     /ERROR, SET UP AND GO TO
0504 5732  JMP  I  ERR1P     /ERRORS AND DON'T
0505 1172  TAD   TRACE  /RETURN HERE,
0506 7700  SMA  CLA  /TRACE REQUESTED?
0507 5324  JMP   COMPF
0510 7604  LAS
0511 0033  AND   K400        /YES, TRACE INHIBITED?
0512 7640  S&A  CLA
0513 5324  JMP   COMPF
0514 1273  CONTCB, TAD  COMP
0515 3526  DCA  I  ERRORP     /NO, GET ERROR POINTER
0516 1673  TAD  I  COMP      /AND PUT IN ERRORS;
0517 3734  DCA  I  TRPP      /GET STAT1 ADDRESS AND
0520 1336  TAD   K4324    /PUT IN PRNT.
0521 3466  DCA  I  ERMSGP
0522 1335  TAD   K2252
0523 4733  JMS  I  PRNTP
0524 2273  COMPF, IS&  COMP      /FAKE AN ERROR PRINTOUT,
0525 4520  LOOP&          /UPDATE RETURN JUMP TO SUBTEST LOOP JUMP,
0526 7410  SKP
0527 2273  IS&  COMP      /SUBTEST LOOP?
0530 4515  CLEAR1
0531 5673  JMP  I  COMP      /NO, UPDATE ONE MORE.
0532 5201  ERR1P, ERRORS=1  /EXIT.
0533 5254  PRN1P, PRNT
0534 5347  TRPP,  ERRPTR
0535 2252  K2252, 2252
0536 4324  K4324, 4324
    
```

```

/TEST 16.      BASIC MOTION TEST,

/TESTS ALL TIME SEQUENCED CONDITIONS FOR WRITE, READ, READ-COMPARE,
/SPACE FORWARD AND REVERSE, AND REWIND. ALL TESTS ARE BASED UPON 2 WORD
/RECORD (4 CHARC FRAMES IN 7 TRK OR 9TRK CORE DUMP) WRITTEN FROM BOT BY T16A.

0600 0000  PAGE
TEST16, 0

/WRITE A 2 WORD RECORD FROM BOT AND CHECK BASIC TIME SEQUENCED
/OPERATIONS.
T16A,  SET1
0601 4530  0570 /GD
0602 0570  -2 /WC
0603 7776  T2&BUF /CA
0604 7342  0 /MEM
0605 0000  ODD /CM
0606 0400  WRITE+GO/FS
0607 4100  TSKEF /010 /WRITE RECORD
0610 4535  TSKCB /110 SKCB SHOULD NOT SKIP
0611 4537  TSKTR /210 SKTR SHOULD NOT SKIP
0612 4940  TWC /311 WC SHOULD INCREMENT TO 7777
0613 4543  -1
0614 7777  WAIT2 /WAIT FOR ROT TO GO AWAY
0615 4545  0
0616 3030  TMS /410 BOT SHOULD GO AWAY BY NOW
0617 4541  1000
0620 1000  WAIT2 /WAIT FOR HTTF
0621 4545  1
0622 3001  TSKTD /511 HTTF SHOULD BE SET BY NOW
0623 4536  NOP
0624 7000  TSKCB /611 SKCB SHOULD SKIP
0625 4537  WAIT1 /WAIT FOR TUR
0626 4544  1
0627 0001  TSKTR /711 SKTR SHOULD SKIP BY NOW
0630 4540  NOP
0631 7000  TWC /811 WC SHOULD BE 0000
0632 4543  0000
0633 0000  TSKEF /910 THERE SHOULD BE NO ERROR
0634 4535  ERSTAT
0635 1162  TAD
0636 4525  COMPAR
0637 6507  ER16A

/GO=GOOD ERSTAT; BD=REAL ERSTAT
0640 5201  JMP T16A /SUBTEST LOOP

/SPACE REVERSE 2 RECORDS TO BOT, REFER TO "TSPREV" ROUTINE BELOW
/FORM DEFINITIONS OF ERSTAT BITS.
T16B, JMS I TSPREP /GO TO TSPREV ROUTINE
0641 4756  COMPAR
0642 4525  ER16B
0643 6511

/GO=GOOD ERSTAT; BD=REAL ERSTAT
0644 5241  JMP T16B /SUBTEST LOOP
    
```

```

/READ-COMPARE A RECORD FROM BOT, REFER TO "TMOFWD" ROUTINE BELOW
/FOR DEFINITIONS OF ALL ERSTAT BITS,
T16C, JMS I TMOFWP /GO TO TMOFWD ROUTINE
RDCOMP+GO /TO DO READ-COMPARE
COMPAR
ER16C
/GO=GOOD ERSTAT; BD=REAL ERSTAT
0651 5245 JMP T16C /SUBTEST LOOP

/SPACE REVERSE 2 RECORDS TO BOT, REFER TO "TSPREV" ROUTINE BELOW
/FOR ERSTAT BIT DEFINITIONS,
T16D, JMS I TSPREP /GO TO TSPREV ROUTINE
COMPAR
ER16D
/GO=GOOD ERSTAT; BD=REAL ERSTAT
0652 4756 JMP T16D /SUBTEST LOOP
0653 4525
0654 6515

LOOP6 /STATUS BIT DEFINITIONS,
0656 4521 JMP T16A /*****LOOP 6****

/READ A 2 WORD RECORD FROM BOT, REFER TO "TMOFWD" ROUTINE BELOW
/FOR ERSTAT BIT DEFINITIONS,
T16E, JMS I TMOFWP /GO TO TMOFWD ROUTINE
READ+GO /TO READ A RECORD,
COMPAR
ER16E
/GO=GOOD ERSTAT; BD=REAL ERSTAT
0660 4755 JMP T16E /SUBTEST LOOP
0661 2100
0662 4525
0663 6517

/SPACE REVERSE 2 RECORDS TO BOT, REFER TO "TSPREV" ROUTINE BELOW
/FOR ERSTAT BIT DEFINITIONS,
T16F, JMS I TSPREP /GO TO TSPREV ROUTINE
COMPAR
ER16F
/GO=GOOD ERSTAT; BD=REAL ERSTAT
0665 4756 JMP T16F /SUBTEST LOOP
0666 4525
0667 6521

LOOP7
0670 5265 JMP T16E-2 /*****LOOP 7****

/SPACE FORWARD 1 RECORD FROM BOT, REFER TO "TSPFWD" ROUTINE BELOW
/FOR ERSTAT DEFINITIONS,
T16G, JMS I TSPFWP /TO TSPFWD ROUTINE
COMPAR
ER16G
/GO=GOOD ERSTAT; BD=REAL ERSTAT
0673 4757 JMP T16G /SUBTEST LOOP
0674 4525
0675 6523

/SPACE REVERSE 2 RECORDS TO BOT, REFER TO "TSPREV" ROUTINE BELOW
/FOR ERSTAT DEFINITIONS,
T16H, JMS I TSPREP /TO TSPREV ROUTINE
COMPAR
ER16H
/GO=GOOD ERSTAT; BD=REAL ERSTAT
0677 4756 JMP T16H /SUBTEST LOOP
0700 4525
0701 6525
0702 5277

```

```

LOOP8
0703 4523 JMP T16G-2 /*****LOOP 8****
0704 5271

/SPACE FORWARD 1 RECORD FROM BOT, REFER TO "TSPFWD" ROUTINE BELOW
/FOR ERSTAT BIT DEFINITIONS,
T16I, JMS I TSPFWP /TO TSPFWD ROUTINE
COMPAR
ER16I
/GO=GOOD ERSTAT; BD=REAL ERSTAT
0705 4757 JMP T16I /SUBTEST LOOP
0706 4525
0707 6527

/REWIND OVER 1 RECORD TO BOT,
T16J, SET3
0711 4532 3524 /GO
0712 3524 -1 /WC
0713 7777 ODD /CM
0714 0400 REWIND+GO/FS
0715 1100
0716 4535 TSKEF /010 SKEF SHOULD NOT SKIP
0717 4541 TMS /111 REWIND STATUS SHOULD BE SET
0720 2000
0721 4536 TSKTD /211 MTF SHOULD BE SET AND SKTD SHOULD SKIP
/SINCE REWIND STAT,REWIND FUNCTION
0722 4537 TSKCB /311 SKCB SHOULD NOT SKIP
0723 4511 CLF /CLEAR MTF TO CHECK SETTING AT BOT
0724 4540 TSKTR /410 SKTR SHOULD NOT SKIP YET
0725 4545 WAIT2 /WAIT FOR BOT
0726 0001 1
0727 4541 TMS /511 BOT SHOULD BE SET BY NOW
0730 1000 1000
0731 4545 WAIT2 /WAIT FOR LOSS OF REWIND STATUS
0732 0000 0
0733 4541 TMS /610 REWIND STATUS SHOULD GO AWAY BY NOW
0734 2000 2000
0735 4544 WAIT1 /WAIT FOR TUR
0736 0001 1
0737 4540 TSKTR /711 SKTR SHOULD SKIP BY NOW
0740 7000 NOP
0741 4536 TSKTD /810 MTF SHOULD NOT SET AT BOT WHEN TUR;
/THEREFORE, SKTD SHOULD NOT SKIP,
BOT SHOULD STILL BE SET
0742 4541 TMS /911
0743 1000 1000
0744 4535 TSKEF /1010 THERE SHOULD BE NO ERROR
0745 1162 TAD ERSTAT
0746 4525 COMPAR
0747 6531 ER16J
/GO=GOOD ERSTAT; BD=REAL ERSTAT
0750 5311 JMP T16J /SUBTEST LOOP

LOOP9
0751 4524 JMP T16I-2 /*****LOOP 9****
0752 5303

LOADPT
0753 4527 JMP I TEST16
0754 5000

```

```

0755 1000 TMOFWD, TMOFWD
0756 1046 TSPREP, TSPREV
0757 1102 TSPFWD, TSPFWD

1000 PAGE
/ " T M O F W D " ROUTINE
/USED FOR READ-COMPARE AND READ FUNCTIONS,
TMOFWD, 0 /TEST MOTION FORWARD FOR READ AND READ-COMPARE
1001 1000 TAD I TMOFWD /GET AND SAVE FUNCTION
1002 3212 DCA ,+10
1003 2200 ISZ TMOFWD /SETUP RETURN
1004 4530 SET1
1005 0574 0574 /GD
1006 7776 -2 /WC
1007 7342 T29BUF /CA
1010 0000 0 /MEM
1011 0400 ODD /CH
1012 0000 0 /FR
1013 4935 TSKEF /010 /HDCOMP+GO OR READ+GO
1014 4937 TSKCB /110 THERE SHOULD BE NO ERROR
1015 4940 TSKTR /210 CONTROL SHOULD BE BUSY
1016 4543 TWC /311 TAPE UNIT SHOULD NOT BE READY
1017 7776 -2 /311 WC SHOULD STILL BE 7776
1020 4545 WAIT2 /WAIT FOR BOT TO GO AWAY
1021 0000 0
1022 4541 TMS /410 BOT SHOULD BE GONE BY NOW
1023 1000 1000
1024 4545 WAIT2 /WAIT FOR WC TO INCREMENT TO 0000
1025 0001 1
1026 4943 TWC /511 WC SHOULD = 0002 BY NOW
1027 0000 0000
1030 4545 WAIT2 /WAIT FOR HTTF
1031 0001 1
1032 4936 TSKTD /611 HTTF SHOULD BE SET BY NOW
1033 7000 NOP
1034 4937 TSKCB /711 CONTROL SHOULD NOT BE BUSY
1035 4544 WAIT1 /WAIT FOR TUR
1036 0001 1
1037 4540 TSKTR /811 TAPE UNIT SHOULD BE READY BY NOW
1040 7000 NOP
1041 4543 TWC /911 WC SHOULD STILL BE 0000
1042 0000 0000
1043 4935 TSKEF /1010 THERE SHOULD BE NO ERROR
1044 1162 TAD ERSTAT /ERSTAT IN AC
1045 5600 JMP I TMOFWD /EXIT FOR COMPAR,

```

```

/ " T S P R E V " ROUTINE
/USED TO SPACE REVERSE 2 RECORDS TO BOT OVER ONE RECORD WRITTEN FROM
/BOT.
1046 0000 TSPREV, 0 /SPACE REVERSE 2 RECORDS TO BOT
1047 4932 SET3
1050 0714 0714 /GD
1051 7776 -2 /WC
1052 0400 ODD /CH
1053 7100 SPCREV+GO/FS

```

```

1054 4935 TSKEF /010 THERE SHOULD BE NO ERROR
1055 4937 TSKCB /110 CONTROL SHOULD BE BUSY
1056 4540 TSKTR /210 TAPE UNIT SHOULD NOT BE READY
1057 4545 WAIT2 /WAIT FOR WC TO INCREMENT TO 7777
1060 0001 1
1061 4543 TWC /311 WC SHOULD = 7777 BY NOW
1062 7777 -1
1063 4545 WAIT2 /WAIT FOR BOT TO SET
1064 0001 1
1065 4541 TMS /411 BOT SHOULD BE SET BY NOW
1066 1000 1000
1067 4935 TSKEF /511 BOT AND SPCREV SHOULD SET ERROR
1070 4936 TSKTD /610 HTTF SHOULD NOT BE SET
1071 4937 TSKCB /710 CONTROL SHOULD STILL BE BUSY
1072 4544 WAIT1 /WAIT FOR TUR
1073 0001 1
1074 4540 TSKTR /811 TAPE UNIT SHOULD BE READY BY NOW
1075 7000 NOP
1076 4543 TWC /911 WC SHOULD STILL BE 7777,
1077 7777 7777
1100 1162 TAD ERSTAT /ERSTAT IN AC
1101 5646 JMP I TSPREV /EXIT

```

```

/ " T S P F W D " ROUTINE,
/SPACE FORWARD 1 RECORD FROM BOT,
1102 0000 TSPFWD, 0 /SPACE FORWARD 1 RECORD FROM BOT
1103 4932 SET3
1104 0370 0370 /GD
1105 7777 -1 /WC
1106 0400 ODD /CH
1107 6100 SPCFWD+GO/FS
1110 4935 TSKEF /010 THERE SHOULD BE NO ERROR
1111 4937 TSKCB /110 CONTROL SHOULD BE BUSY
1112 4940 TSKTR /210 TRANSPORT SHOULD NOT BE READY
1113 4545 WAIT2 /WAIT FOR BOT TO GO AWAY
1114 0000 0
1115 4541 TMS /310 BOT SHOULD BE GONE BY NOW
1116 1000 1000
1117 4545 WAIT2 /WAIT FOR AC TO INCREMENT TO 0000
1120 0001 1
1121 4543 TWC /411 WC SHOULD BE 0002 BY NOW
1122 0000 0
1123 4544 WAIT1 /WAIT FOR HTTF TO SET
1124 0001 1
1125 4936 TSKTD /511 HTTF SHOULD BE SET BY NOW
1126 7000 NOP
1127 4937 TSKCB /611 CONTROL SHOULD NOT BE BUSY
1130 4544 WAIT1 /WAIT FOR TUR
1131 0001 1
1132 4940 TSKTR /711 TAPE UNIT SHOULD BE READY BY NOW
1133 7000 NOP
1134 4543 TWC /811 WC SHOULD STILL BE 0000
1135 0000 0
1136 4935 TSKEF /910 THERE SHOULD BE NO ERROR
1137 1162 TAD ERSTAT /ERSTAT IN AC

```

```

/LOOP SWITCH SENSORS.
1141 0000 LAS9, 0 /ENTER BY "LOOP5",
1142 7604 LAS AND K100
1143 0030 SNA CLA
1144 7650 ISZ LAS5
1145 2341 JMP I LAS5
1146 5741 LAS6, 0 /"LOOP6"
1147 0000 LAS AND K40
1148 7604 SNA CLA
1149 0026 ISZ LAS6
1150 2347 JMP I LAS6
1151 5747 LAS7, 0 /"LOOP7"
1152 0000 LAS AND K20
1153 7604 SNA CLA
1154 0377 ISZ LAS7
1155 7650 JMP I LAS7
1156 2355 LAS8, 0 /"LOOP8"
1157 5755 LAS AND K10
1158 7604 SNA CLA
1159 0023 ISZ LAS8
1160 7650 JMP I LAS8
1161 2363 LAS9, 0 /"LOOP9"
1162 5763 LAS AND K4
1163 0000 SNA CLA
1164 7604 ISZ LAS9
1165 0021 JMP I LAS9
1166 7650 K20, 20
1167 2371
1168 5771
1169 0020
    
```

```

/TEST 17, LATERAL PARITY TEST,

/IN GENERAL THIS TEST WRITES 2 WORD RECORDS AT ODD OR EVEN
/PARITY AND CHECKS THAT THERE IS NO PARITY ERROR, THEN THE RECORDS
/ARE READ AT THE OPPOSITE PARITY AND A PARITY ERROR IS
/VERIFIED.

/LOOPS 6-9:
/LOOP 6 CYCLES ON WRITE EVEN, READ ODD (T17A-T17B),
/LOOP 7 CYCLES ON WRITE ODD, READ EVEN (T17C-T17D),

1200 1200 PAGE
1200 0000 TEST17, 0

/WRITE 2 WORD RECORDS (CORE DUMP) USING EVEN PARITY, NO PARITY
/ERROR SHOULD OCCUR, DATA USED IS 01-77, AFTER EACH RECORD IS
/WRITTEN, T17B IS RUN USING THAT RECORD, 9 TRK IS IN CORE DUMP,
T17A, LOADPT /TO BOT
TAD M77
DCA TXXTH1
IAC
DCA T17ATH
T17AL1, SET1 /WRITE 2 WORD RECORD
4000 /GD
-2 /WC
T2>8BUF /CA
T17ATH, 0 /MEM
EVEN /CM
WRITE*GO/FR
WAIT2 /WAIT FOR CONTROL NOT BUSY
1 /011 CONTROL SHOULD BE READY BY NOW
TSKCB /010 THERE SHOULD BE NO PARITY ERROR
NOP /110 THERE SHOULD BE NO ERRORS
TMS /210
TSKEF
TAD T17ATH
DCA OLD
TAD ERSTAT
COMPAR
ER17A
/GO=GOOD ERSTAT; BD=REAL ERSTAT; OD=DATA USED IN THIS RECORD,
JMP T17AL1 /SUBTEST LOOP
SKP
JMP T17LP6

/USING THE RECORD LAST WRITTEN IN T17A, SPREV TO THE BEGINNING
/OF THE RECORD, THEN READ IT USING ODD PARITY, A PARITY ERROR
/SHOULD OCCUR AND EF SHOULD SET,
T17B, SET3 /SPREV ONE RECORD
0 /GD
-1 /WC
EVEN /CM
    
```

```

1240 7100 SPREV*GO/FR
1241 4532 SET1 /READ RECORD
1242 5000 5000 /GD
1243 7776 -2 /WC
1244 7342 T2>8BUF /CA
1245 0000 0 /MEM
1246 0400 ODD /CM
1247 2100 READ*GO /FR
1250 4545 WAIT2 /WAIT FOR CONTROL READY
1251 0001 1
1252 4537 TSKCB /011 CONTROL SHOULD BE READY BY NOW
1253 7000 NOP /110 LATERAL PARITY ERROR BIT SHOULD NOT BE SET
1254 4542 TFS /211 EF AND PARITY ERROR SHOULD BE SET
1255 0004 4
1256 4541 TMS
1257 4200 4200
1260 1162 TAD ERSTAT
1261 4525 COMPAR
1262 6535 ER17B
/GO=GOOD ERSTAT; BD=REAL ERSTAT; OD=DATA USED
JMP T17B /SUBTEST LOOP
SKP
JMP T17LP6
1266 2151 ISZ TXXTH1
SKP
JMP T17LP6
1271 2212 ISZ T17ATH
JMP T17AL1

1273 4521 T17LP6, LOOP6
1274 5201 JMP T17A /*****LOOP6*****/

/WRITE 2 WORD RECORDS USING ODD PARITY, NO PARITY ERROR SHOULD OCCUR,
/DATA IS 00-77, AFTER EACH RECORD IS WRITTEN, THAT RECORD IS READ
/IN EVEN PARITY BY T17D,
T17C, LOADPT /TO BOT
TAD M100
DCA TXXTH1
DCA T17CTH
T17CL1, SET1 /WRITE 2 WORD RECORD
4000 /GD
-2 /WC
T2>8BUF /CA
T17CTH, 0 /MEM
ODD /CM
WRITE*GO/FR
WAIT2 /WAIT FOR CONTROL READY
1 /011 CONTROL SHOULD BE READY BY NOW
TSKCB /110 THERE SHOULD NOT BE A PARITY ERROR,
NOP /210 THERE SHOULD BE NO ERRORS
TMS
TSKEF
TAD T17CTH
DCA OLD
    
```

```

1321 1162      TAD      ERSTAT
1322 4525      COMPAR
1323 6537      ER17C
                /GD=GOOD ERSTAT; BD=REAL ERSTAT; DD=DATA USED
1324 5301      JMP      T17CL1      /SUBTEST LOOP
1325 7410      SKP
1326 5755      JMP I   T17L7P

/USING THE LAST RECORD WRITTEN IN T17C, READ THAT RECORD USING
/EVEN PARITY. A PARITY ERROR SHOULD RESULT AND EF SHOULD SET,
T17D, SEI3      /SPCREV 1 RECORD
1327 4532      0          /GD
1330 0000      -1         /HC
1331 7777      ODD        /CH
1332 0400      SPCREV=GD/FR
1333 7100      SET1
1334 4530      5000      /GD
1335 5000      -2         /HC
1336 7776      T2>BUF   /CA
1337 7342      0          /HEM
1340 0000      EVEN      /CH
1341 0000      READ=GO /FR
1342 2100      WAIT2
1343 4545      1          /WAIT FOR CONTROL READY
1344 3031      TSKCB
1345 4537      NOP
1346 7000      TFS
1347 4542      4
1348 0004      TMS
1350 0004      4200
1351 4541      JMP I   ,*1
1352 4200      T17DCK
1353 5754      T17L7P, T17LP7
1354 1400      M77,    -77
1355 1413      M100,   -100
1356 7701      PAGE
1357 7700      T17DCK, TAD      ERSTAT
                COMPAR
                ER17D
                /GD=GOOD ERSTAT; BD=REAL ERSTAT; DD=DATA USED
1400 1162      JMP I   T17DP      /SUBTEST LOOP
1401 4525      SKP
1402 6541      JMP      T17LP7
                IS2      TXXTM1
1403 5626      SKP
1404 7410      JMP      T17LP7
1405 5213      IS2 I   T17CTP
1406 2151      JMP I   T17CLP
1407 7410      T17LP7, LOOP7
1408 5213      JMP I   T17CP      /*****LOOP7****
1411 2625      LOADPT
1412 5624      TAD I   TST17P
                DCA     ,*2
                JMP I   ,*1

1413 4522      0
1414 5623      TST17P, TEST17
1415 4527      T17CP,  T17LP6
1416 1622      T17CLP, T17CL1
1417 3221      T17CTP, T17CTM
1420 5621      T17DP,  T17D

```

```

1421 0000      0
1422 1200      TST17P, TEST17
1423 1273      T17CP,  T17LP6
1424 1301      T17CLP, T17CL1
1425 1305      T17CTP, T17CTM
1426 1327      T17DP,  T17D

```

```

/TEST 20.      WEOF TEST,
              /WEOF FROM BOT AND VERIFY USING SPCFWD,

1427 0000     TEST20, 0

/WRITE EOF FROM BOT AND VERIFY,
T20A.  LOADPT
1430 4527     SET3
1431 4532
1432 0740     0740 /GD
1433 7777     -1 /WC
1434 0400     ODD /CH
1435 5100     WEOF+GO /FR
1436 3171     DCA EXPEOF
1437 4535     TSKEF /010 THERE SHOULD BE NO INITIAL EHRDR
1440 4537     TSKCB /110 CONTROL SHOULD BE BUSY
1441 4540     TSKTR /210 TRANSPORT SHOULD NOT BE READY
1442 4545     WAIT2 /WAIT FOR TRANSPORT READY
1443 0001     1
1444 4540     TSKTR /311 TRANSPORT SHOULD BE READY BY NOW
1445 7000     NOP
1446 4543     TWC /411 WC SHOULD STILL BE 7777 SINCE NO DATA BREAK
1447 7777     -1
1450 4535     TSKEF /511 ERROR SHOULD BE SET SINCE EOF
1451 4541     TMS /611 EOF SHOULD BE SET
1452 4100
1453 1162     TAD ERSTAT
1454 4525     COMPAR
1455 6543     ER20A

/GD=GOOD ERSTAT; BD=REAL ERSTAT
1456 5231     JMP T20A+1 /SUBTEST LOOP

/WRITE EOF FROM BOT; REWIND TO BOT OVER EOF; THEN TRY TO SPCFWD
/2 RECORDS OVER EOF, MTF SHOULD SET WITH EOF BIT SET AND WC =7776;
T20B.  LOADPT
1457 4527     SET3 /WEOF FROM BOT
1460 4532
1461 0000     0 /GD
1462 7777     -1 /WC
1463 0400     ODD /CH
1464 5100     WEOF+GO /FR
1465 3171     DCA EXPEOF
1466 4527     LOADPT /TO BOT
1467 4532     SET3 /TRY TO SPCFWD 2 RECORDS OVER EOF
1470 7400     7400 /GD
1471 7776     -2 /WC
1472 0400     ODD /CH
1473 6100     SPCFWD+GO/FR
1474 3171     DCA EXPEOF
1475 4545     WAIT2 /WAIT FOR MTF
1476 0001     1
1477 4536     TSKTD /011 MTF SHOULD EVENTUALLY SET
1500 7000     NOP
1501 4541     TMS /111 EOF SHOULD BE SET
1502 4100
    
```

```

1503 4535     TSKEF /211 THERE SHOULD BE AN ERROR SINCE EOF
1504 4543     TWC /311 WORD COUNT SHOULD STILL BE 7776
1505 7776     -2
1506 1162     TAD ERSTAT
1507 4525     COMPAR
1510 6545     ER20B

/GD=GOOD ERSTAT; BD=REAL ERSTAT
1511 5257     JMP T20B /SUBTEST LOOP

/WRITE EOF FROM BOT; THEN WRITE ONE RECORD; THEN SPACE REVERSE
/2 RECORDS, EXPECT EOF WITH WC=-1,
T20C.  LOADPT
1512 4527     SET3 /WEOF FROM BOT
1513 4532
1514 0000     0 /GD
1515 7777     -1 /WC
1516 0400     ODD /CH
1517 5100     WEOF+GO/FR
1520 3171     DCA EXPEOF
1521 4530     SET1 /WRITE 1 RECORD
1522 0000     0 /GD
1523 7776     -2 /WC
1524 7342     T20BUF /CA
1525 0000     0 /MEM
1526 0400     ODD /CH
1527 4100     WRITE+GO/FR
1530 4532     SET3 /TRY TO SPCREV 2 RECORDS OVER EOF
1531 7400     7400 /GD
1532 7776     -2 /WC
1533 0400     ODD /CH
1534 7100     SPCREV+GO/FR
1535 3171     DCA EXPEOF
1536 4545     WAIT2 /WAIT FOR MTF
1537 0001     1
1540 4536     TSKTD /011 MTF SHOULD BE SET BY NOW
1541 7000     NOP
1542 4541     TMS /111 EOF SHOULD BE SET
1543 4100
1544 4535     TSKEF /211 THERE SHOULD BE AN ERROR SINCE EOF
1545 4543     TWC /311 WORD COUNT SHOULD BE -1
1546 7777     -1
1547 1162     TAD ERSTAT
1550 4525     COMPAR
1551 6547     ER20C

/GD=GOOD ERSTAT; BD=REAL ERSTAT
1552 5312     JMP T20C /SUBTEST LOOP

1553 4527     LOADPT
1554 4521     LOOP6
1555 5230     JMP T20A /*****LOOP6****
1556 5627     JMP I TEST20
    
```

/TEST 21. COMBINED FUNCTIONS TEST,
 /WRITE 2-2 WORD RECORDS THEN EOF AND VERIFY,

1600	1000		PAGE	
	0000	TEST21.	0	
		/WRITE A 2 WORD RECORD FROM BOT IN PREPARATION FOR T21B, /DATA IS 7777, 7777.		
1601	4927	T21A.	LOADPT	/TO BOT
1602	4930		SET1	/WRITE 2 WORD RECORD
1603	3400		3400	/GD
1604	7776		-2	/WC
1605	7342		T2>BUF	/CA
1606	7777		7777	/MEM
1607	0400		ODD	/CM
1610	4100		WRITE+GO/FR	
1611	4535		TSKEF	/010 THERE SHOULD BE NO INITIAL ERROR
1612	4545		WAIT2	/WAIT FOR WC=0000
1613	0001		1	
1614	4543		TWC	/111 WC SHOULD BE 0000 BY NOW
1615	0000		0	
1616	4544		WAIT1	/WAIT FOR CONTROL READY
1617	0001		1	
1620	4537		TSKCB	/211 CONTROL SHOULD BE READY BY NOW
1621	7000		NOP	
1622	4544		WAIT1	/WAIT FOR TRANSPORT READY
1623	0001		1	
1624	4540		TSKTR	/311 TRANSPORT SHOULD BE READY BY NOW
1625	7000		NOP	
1626	4535		TSKEF	/410 THERE SHOULD BE NO ERROR
1627	1162		TAD	ERSTAT
1630	4525		COMPAR	
1631	6551		ER21A	
		/GD=GOOD ERSTAT; BD=REAL ERSTAT		
1632	5202		JMP	T21A+1 /SUBTEST LOOP
		/WRITE A 2 WORD RECORD FROM BOT (DATA = 7777, 7777); /THEN SPCREV 1 RECORD TO BOT.		
1633	4527	T21B.	LOADPT	
1634	4530		SET1	/WRITE 2 WORD RECORD
1635	0000		0	/GD
1636	7776		-2	/WC
1637	7342		T2>BUF	/CA
1640	7777		7777	/MEM
1641	0400		ODD	/CM
1642	4100		WRITE+GO/FR	
1643	4532		SET3	/NOW SPCREV WITH WC=0000, SHOULD END UP
1644	7400		7400	/AT BOT WITH WC=1 AND EF SET,
1645	0000		0	/WC
1646	3400		ODD	/CM
1647	7100		SPCREV+GO/FR	
1650	4545		WAIT2	/WAIT FOR TUR

1651	0001		1	
1652	4540		TSKTR	/011 SHOULD BE TUR BY NOW
1653	7000		NOP	
1654	4541		TMS	/111 SHOULD BE AT BOT
1655	1000		1000	
1656	4535		TSKEF	/211 THERE SHOULD BE AN ERROR AT BOT
1657	4543		TWC	/311 WC SHOULD = 0001
1660	0001		1	
1661	1162		TAD	ERSTAT
1662	4525		COMPAR	
1663	6553		ER21B	
		/GD=GOOD ERSTAT; BD=REAL ERSTAT		
1664	5233		JMP	T21B /SUBTEST LOOP
1665	4521		LOOP6	
1666	5201		JMP	T21A /*****LOOP6****
		/WRITE 2 * 2 WORD RECORDS FROM BOT FOLLOWED BY EOF; THEN SPCREV /TO EOF.		
1667	4527	T21C.	LOADPT	
1670	1036		TAD	M2 /WRITE 2-2 WORD RECORDS
1671	3151		DCA	TXXTM1
1672	4530		SET1	
1673	0000		0	/GD
1674	7776		-2	/WC
1675	7342		T2>BUF	/CA
1676	7777		7777	/MEM
1677	0400		ODD	/CM
1700	4100		WRITE+GO/FR	
1701	2151		IS4	TXXTM1
1702	5272		JMP	T21C+3
1703	4532		SET3	/WRITE EOF
1704	0000		0	/GD
1705	0000		0	/WC
1706	0400		ODD	/CM
1707	5100		WEOF+GO	/FR
1710	3171		DCA	EXPEOF
1711	4532		SET3	
1712	7200		7200	/GD /SPCREV WITH WC = -2, SHOULD ENCOUNTER
1713	7776		-2	/WC /SHOULD ENCOUNTER EOF AND TUR WITH
1714	0400		ODD	/CM /WC STILL = -2,
1715	7100		SPCREV+GO/FR	
1716	3171		DCA	EXPEOF
1717	4545		WAIT2	/WAIT FOR TUR
1720	0001		1	
1721	4540		TSKTR	/011 TRANSPORT SHOULD BE READY BY NOW
1722	7000		NOP	
1723	4536		TSKTD	/111 SPACE FUNCTION AND EOF * LPCS SHOULD GIVE HTTF
1724	4543		TWC	/211 WC SHOULD = -2
1725	7776		-2	
1726	4541		TMS	/310 BOT SHOULD NOT BE SET
1727	1000		1000	
1730	4541		TMS	/411 ERROR AND EOF SHOULD BE SET
1731	4100		4100	
1732	1162		TAD	ERSTAT

```
1733 4525 COMPAR
1734 6555 ER21C
/GD=GOOD ERSTAT1 BD=REAL ERSTAT
1735 5267 JMP T21C /SUBTEST LOOP

1736 4922 LOOP7
1737 5265 JMP T21C-2 /*****LOOP 7****
1740 4527 LOADPT /BACK TO BOT,
1741 5000 JMP I TEST21
```

```
/RETURN TO BOT
1742 0000 LDPT, 0
1743 4532 SET3 / REWIND,
1744 0000 0
1745 0000 0
1746 0400 ODD
1747 1100 REWIND+GD
1750 7240 CLA CMA /SET ALTHODE INTERRUPT
1751 3165 DCA ALTENA /ENABLE,
1752 6001 ION
1753 4910 SKTR
1754 5353 JMP ,=1
1755 4915 CLEAR1
1756 5742 JMP I LDPT /EXIT,
```

```

/TEST22, BASIC CONTROL TEST USING TAPE MOTION,
2000 2000 PAGE
2000 0072 TEST22, 0

2001 4527 T22A, LOADPT /IF SELECTED DRIVE IS 9 TRACK,
2002 1161 TAD TRK9 /VERIFY THAT 9CM, CM10
2003 7650 SNA CLA /CLEAN AND PRESET GIVE IF
2004 5221 JMP T22B /AND EF,
2005 4533 SET4
2006 6000 6000 /GD
2007 7776 -2 /WC
2008 0400 ODD /CM
2009 4100 WRITE+GO /FR /USE WRITE;
2010 4541 TMS /0:1 IF SHOULD SET,
2011 0001 1
2012 4535 TSKEF /1:1 EF SHOULD BE SET,
2013 1162 TAD ERSTAT
2014 4525 COMPAR
2015 6557 ER22A

/GO: GOOD ERSTAT; BD: REAL ERSTAT,
2020 5201 JMP T22A /SUBTEST LOOP,

2021 4527 T22B, LOADPT /VERIFY THAT ISSUING LCMR, OR
2022 4734 JMS I T22EX4 /LFGR OR LDBR WHEN THE
2023 0454 T19LS1+7 /CONROL IS BUSY YIELDS IF,
2024 7775 -3
2025 3234 DCA ,*7
2026 4532 SET3
2027 2000 2000 /GD
2028 7776 -2 /WC
2029 0400 ODD /CM
2030 4100 WRITE+GO /FR /USE WRITE;
2031 4535 TSKEF /0:0 INITIALLY THERE SHOULD BE NO ERROR,
2032 0000 0 /LCMR, LFGR OR LDBR, (REF "111"),
2033 4541 TMS /1:1 AFTER THE IOT, IF SHOULD SET,
2034 0001 1
2035 1162 TAD ERSTAT
2036 4525 COMPAR
2037 6561 ER22B

/GO: GOOD ERSTAT; BD: REAL ERSTAT,
2042 5735 JMP I T22EX5 /SUBTEST LOOP,
2043 4736 JMS I T22EX6

2044 4527 T22C, LOADPT /VERIFY THAT ISSUING AN IOT
2045 4532 SET3 /OTHER THAN LCMR, LFGR, OR
2046 0000 0 /LDBR WHEN THE CONTROL IS
2047 7776 -2 /WC /BUSY DOES NOT YIELD IF,
2048 0400 ODD /CM
2049 4100 WRITE+GO /FR
2050 4473 LCMR /LCMR USED;
2051 4535 TSKEF /0:0 THERE SHOULD BE NO ERROR,
2052 4541 TMS /1:0 THERE SHOULD BE NO IF,
2053 0001 1
2054 1162 TAD ERSTAT
2055 4525 COMPAR
2056 6563 ER22C

```

```

2057 4525 COMPAR
2058 6563 ER22C
/GO: GOOD ERSTAT; BD: REAL ERSTAT,
2061 5244 JMP T22C /SUBTEST LOOP,

2062 4521 LOOP6
2063 5201 JMP T22A /*****LOOP6****

2064 4301 T22D, JMS T22DE /VERIFY WRITE+GO FOLLOWED BY
2065 4100 WRITE+GO /READ, RDCOMP, OR SPCFWD GIVES
2066 4525 COMPAR /IF, USE T22DE ROUTINE,
2067 6565 ER22D

/GO: GOOD ERSTAT; BD: REAL ERSTAT,
2070 5311 JMP T22DE1 /SUBTEST LOOP,
2071 4331 JMS T22DE2

2072 4301 T22E, JMS T22DE /VERIFY THAT WEOF+GO FOLLOWED
2073 5100 WEOF+GO /BY READ, RDCOMP OR SPCFWD GIVES
2074 4525 COMPAR /IF,
2075 6567 ER22E /USES T22DE ROUTINE,

/GO: GOOD ERSTAT; BD: REAL ERSTAT,
2076 5311 JMP T22DE1 /SUBTEST LOOP,
2077 4331 JMS T22DE2
2100 5337 JMP T22F

2101 0000 T22DE, 0
2102 1701 TAD I T22DE /GET WRITE OR WEOF AND GO
2103 3317 DCA T22DE1 /AND PUT IN CALLING SEQUENCE,
2104 2301 1S# T22DE /UPDATE TO RETURN,
2105 4734 JMS I T22EX4 /GO TO EXEC TO INSERT READ,
2106 2171 T22LS /RDCOMP, OR SPCFWD IN SET
2107 7775 -3 /CALL,
2110 3322 DCA T22DEJ

2111 4530 T22DE1, SET1 /WRITE -GO OR WEOF+GO,
2112 4000 4000 /GD
2113 7776 -2 /WC
2114 7342 T29BUF /CA
2115 7777 7777 /MEM
2116 0400 ODD /CM

2117 0000 T22DE1, 0 /FR /FUNCTION INSERTED HERE
2120 3171 DCA EXPEOF
2121 4534 CONTNU
2122 0000 T22DEJ, 0 /FR
2123 4544 WAIT1 /READ, RDCOMP, OR SPCFWD AND GO,
2124 0001 1 /WAIT FOR IF AND ERRCR BITS TO SET
2125 4541 TMS /0:1 IF AND ERROR BITS SHOULD BE SET BY NOW,
2126 4001 4001
2127 1162 TAD ERSTAT
2130 5701 JMP I T22DE

2131 0000 T22DE2, 0 /ENTER AT END OF SUBTEST,
2132 4736 JMS I T22EX6 /GET NEXT FUNCTION,
2133 5731 JMP I T22DE2 /DONE,

```

```

2134 0423 T22E*4, T19EX4
2135 0434 T22E*5, T19EX5
2136 0437 T22E*6, T19EX6

2137 4527 T22F, LOADPT /VERIFY THAT 4 MISSING CHARACTERS DO NOT
2140 3453 DCA 1 XBUFF+2 /CAUSE BAD TAPE ERROR AND
/THAT EF SETS, DATA IS 7777,7777
/0000, 0000, WRITE WITH EVEN
/PARITY,

2141 3454 DCA 1 XBUFF+3
2142 4530 SET1
2143 4000 /GD
2144 7774 -4 /WC
2145 7341 T22BUF=1 /CA
2146 7777 -1 /MEM
2147 0000 EVEN /CM
2150 4100 WRITE+GO /FR /WRITE=GO,
2151 4545 WAIT2 /WAIT FOR TRANSPORT READY,
2152 0001 1
2153 4540 TSKTR /0:1 TRANSPORT SHOULD BE READY BY NOW,
2154 7000 NOP
2155 4542 TFS /1:0 BAD TAPE ERROR SHOULD NOT BE SET,
2156 0020 20
2157 4541 TMS /2:0 EF SHOULD NOT BE SET,
2160 4000 4000
2161 1162 TAD ERSTAT
2162 4525 COMPAR
2163 6571 ER22F
/GO: GOOD ERSTAT; BD: REAL ERSTAT,
JMP T22F+4 /SUBTEST LOOP,

2165 4522 T22LP7, LOOP7
2166 5262 JMP T22D-2 /*****LOOP 7*****

2167 5770 JMP I ,+1
2170 2200 T22G, SPCFWD+GO
2171 6100 READ+GO
2172 2100 RDCOMP+GO
2173 3100

2200 2200 PAGE
2200 4527 LOADPT /VERIFY RECORD LENGTH
2201 7240 CLA CMA /INCORRECT WHEN WC SET
2202 3453 DCA 1 XBUFF+2 /TO 1 WORD FOR READING A
2203 4530 SET1 /2 WORD RECORD,
/WRITE A 2 WORD RECORD FROM
/BOT,
2204 0000 0 /GD
2205 7776 -2 /WC
2206 7342 T22BUF /CA
2207 7777 -1 /MEM
2210 0400 ODD /CM
2211 4100 WRITE+GO /FR

2212 4527 T22GL, LOADPT /READ FROM BOT,
2213 4530 SET1
2214 7000 /GD

```

```

2215 7777 =1 /WC /SET UP FOR 1 WORD RECORD,
2216 7342 T22BUF /CA
2217 7777 =1 /MEM
2220 0400 ODD /CM
2221 2100 READ+GO /FR /READ=GO,
2222 4545 WAIT2 /WAIT FOR TRANSPORT READY,
2223 0001 1
2224 4540 TSKTR /0:1 TRANSPORT SHOULD BE READY BY NOW,
2225 7000 NOP
2226 4543 TWC /1:1 WC SHOULD BE 0000,
2227 0000 0
2230 4541 TMS /2:1 REC LENGTH AND EF SHOULD BE SET,
2231 4040 4040
2232 1162 TAD ERSTAT
2233 4525 COMPAR
2234 6573 ER22G
/GO: GOOD ERSTAT; BD: REAL ERSTAT,
JMP T22GL /SUBTEST LOOP,

2236 4527 T22H, LOADPT /VERIFY RECORD LENGTH INCORRECT
2237 4530 SET1 /WHEN WC SET FOR 3 WORD
2240 7000 7000 /GD /RECORD WHEN RECORD IS
2241 7775 -3 /WC /ACTUALLY 2 WORDS,
2242 7342 T22BUF /CA
2243 7777 -1 /MEM
2244 0400 ODD /CM
2245 2100 READ+GO /FR /READ=GO,
2246 4545 WAIT2 /WAIT FOR TRANSPORT READY,
2247 0001 1
2250 4540 TSKTR /0:1 TRANSPORT SHOULD BE READY BY NOW,
2251 7000 NOP
2252 4543 TWC /1:1 WC SHOULD BE 7777,
2253 7777 7777
2254 4541 TMS /2:1 REC LENGTH AND EF SHOULD BE SET,
2255 4040 4040
2256 1162 TAD ERSTAT
2257 4525 COMPAR
2260 6575 ER22H
/GO: GOOD ERSTAT; BD: REAL ERSTAT,
JMP T22H /SUBTEST LOOP,

2261 5236 T22I, LOADPT /VERIFY DATA REQUEST LATE,
2262 4527 SET1 /EF SHOULD ALSO BE SET,
2263 4530 6000 /GD
2264 6000 /WC
2265 0000 /CM
2266 7342 T22BUF /CA
2267 7730 =50 /MEM
2270 0400 ODD /CM
2271 4100 WRITE+GO /FR /USE WRITE=GO,
2272 4472 LWCR
2273 2452 ISX I XBUFF+1
2274 5272 JMP ,=2
2275 2453 ISX I XBUFF+2
2276 5272 JMP ,=4
2277 4513 SDLE /SET DATA LATE ERROR,

```

```

2300 4545      WAIT2      /WAIT FOR TRANSPORT READY,
2301 0001      1
2302 4540      TSKTR /0:1  TRANSPORT SHOULD BE READY BY NOW,
2303 7000      NOP
2304 4541      TMS /1:1  DATA LATE AND EF SHOULD BE SET
2305 4020      4020
2306 1162      TAD      ERSTAT
2307 4525      COMPAR
2310 6577      ER221
/GD! GOOD ERSTAT; BD: REAL ERSTAT,
2311 5262      JMP      T221 /SUBTEST LOOP,

2312 4523      LOOP8
2313 5774      JMP I  T22L7P /*****LOOP 8****

2314 4335      T22J,  JMS  T22JK /VERIFY RDCOMP ERROR OCCURS
2315 0000      0 /DATA FOR TAPE, /WHEN DATA IN MEM IS 7777 AND
2316 4525      COMPAR /DATA ON TAPE SHOULD BE 0000,
2317 6601      ER22J /REFER TO T22JK ROUTINE BELOW FOR
/GD! GOOD ERSTAT; BD: REAL ERSTAT,
2320 5314      JMP      T22J /SUBTEST LOOP, INFORMATION,

2321 4335      T22K,  JMS  T22JK /SAME AS T22J EXCEPT
2322 7777      -1 /DATA FOR TAPE, /DATA IN MEM IS 0000 AND DATA
2323 4525      COMPAR /ON TAPE SHOULD BE 7777,
2324 6003      ER22K
/GD! GOOD ERSTAT; BD: REAL ERSTAT,
2325 5321      JMP      T22K /SUBTEST LOOP,

2326 4524      LOOP9
2327 5312      JMP      T22J-2 /*****LOOP 9****
2330 4527      LOADPT

2331 1745      TAD I  TST22P
2332 3334      DCA  .02
2333 5734      JMP I  .01
2334 0000      0

2335 0000      T22JK,  0 /SERVICE FOR T22J AND T22K,
2336 1735      TAD I  T22JK /GET DATA TO PUT ON TAPE,
2337 3350      DCA  T22JKI /SAVE IN SET CALL,
2338 1350      TAD  T22JKI /COMPLEMENT TAPE DATA AND
2339 7040      CHA  /PUT IN SECOND SET CALL,
2342 3360      DCA  T22JKJ
2343 4527      LOADPT /TO BOT,
2344 4531      SET2 /WRITE DATA ON TAPE,
2345 2000      TST22P, TEST22 /GD (NOT USED AS SUCH, BUT AS A POINTER INSTEAD,)
2346 7776      -2 /MC
2347 7342      T29BUF /CA
2350 0000      T22JKI, 0 /MEM /EITHER 0000 OR 7777,
2351 0402      D0007+0DD/CM /CORE DUMP IF 9 TRACK,
2352 4100      WRITE+GO/FR
2353 4527      LOADPT /BACK TO BOT,
2354 4531      SET2 /RDCOMP WITH COMPLEMENT TAPE DATA IN MEM,
2355 6000      6000 /GD
    
```

```

2356 7776      -2 /MC
2357 7342      T29BUF /CA
2360 0000      T22JKJ, 0 /MEM /EITHER 7777 OR 0000,
2361 0402      D0007+0DD/CM /CORE DUMP IF 9 TRACK,
2362 3100      RDCOMP+GO/FR
2363 4545      WAIT2      /WAIT FOR TRANSPORT READY,
2364 0001      1
2365 4540      TSKTR /0:1  TRANSPORT SHOULD BE READY BY NOW,
2366 7000      NOP
2367 4541      TMS /1:1  R/C AND EF SHOULD BE SET,
2370 4002      4002
2371 1162      TAD      ERSTAT
2372 2335      ISZ  T22JK
2373 5735      JMP I  T22JK /GO TO SUBTEST COMPAR,
2374 2165      T22L7P, T22L7P
    
```

```

/TEST23, CONTINUE MODE TEST,
2400 2400 TEST23, 0 /TEST OF CONTINUE MODE;
2401 4255 T23A, JMS T23EX1 /VERIFY WEOF TO WEOF STARTING
2402 5100 WEOF+GO /FR1 /AT BOT, REF, T23EX1 ROUTINE
2403 5100 WEOF+GO /FR2 /FOR INFORMATION,
2404 4525 COMPAR
2405 6625 ER23A
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
2406 5201 JMP T23A /SUBTEST LOOP,
2407 4310 T23B, JMS T23EX2 /VERIFY WRITE TO WRITE STARTING
2410 4100 WRITE+GO /FR1 /AT BOT, REF, T23EX1 ROUTINE
2411 4100 WRITE+GO /FR2 /FOR INFORMATION,
2412 4525 COMPAR
2413 6607 ER23B
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
2414 5207 JMP T23B /SUBTEST LOOP,
2415 4255 T23C, JMS T23EX1 /VERIFY SPCFWD TO SPCFWD
2416 6100 SPCFWD+GO /FR1 /STARTING AT BOT, PLF, T23EX1
2417 6100 SPCFWD+GO /FR2 /ROUTINE FOR INFORMATION,
2420 4525 COMPAR
2421 6611 ER23C
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
2422 5215 JMP T23C /SUBTEST LOOP,
2423 4521 LOOP6
2424 5201 JMP T23A /*****LOOP 6*****/
2425 4310 T23D, JMS T23EX2 /VERIFY READ TO READ FROM
2426 2100 READ+GO /FR1 /BOT, REF T23EX2 ROUTINE
2427 2100 READ+GO /FR2 /FOR INFORMATION,
2430 4525 COMPAR
2431 6613 ER23D
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
2432 5225 JMP T23D /SUBTEST LOOP,
2433 4310 T23E, JMS T23EX2 /VERIFY READ TO WRITE FROM
2434 2100 READ+GO /FR1 /BOT, REFERENCE T23EX2
2435 4100 WRITE+GO/FR2 /ROUTINE FOR INFORMATION,
2436 4525 COMPAR
2437 6615 ER23E
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
2440 5233 JMP T23E /SUBTEST LOOP,
2441 4522 LOOP7
2442 5223 JMP T23D-2 /*****LOOP 7*****/
2443 4255 T23F, JMS T23EX1 /VERIFY READ TO WEOF FROM
2444 2100 READ+GO /FR1 /BOT, REFERENCE T23EX1
2445 5100 WEOF+GO /FR2 /ROUTINE FOR INFORMATION,
2446 4525 COMPAR
2447 6617 ER23F
/GD: GOOD ERSTAT; BD: REAL ERSTAT,

```

```

2450 5243 JMP T23F /SUBTEST LOOP,
2451 4523 LOOP8
2452 5241 JMP T23F-2 /*****LOOP 8*****/
2453 4527 LOADPT
2454 5600 JMP 1 TEST23
2455 0000 T23EX1, 0 /SERVICE FOR T23A, C, AND F,
2456 1055 TAD I T23EX1 /GET FIRST FUNCTION
2457 3273 DCA T23FR1 /AND SAVE,
2460 2255 IS# T23EX1
2461 1055 TAD I T23EX1 /GET SECOND FUNCTION
2462 3276 DCA T23FR2 /AND SAVE,
2463 2255 IS# T23EX1 /UPDATE TO RETURN,
2464 4527 LOADPT /TO BOT,
2465 4530 SET1 /DO FIRST FUNCTION WHEN
2466 7000 /GD /TRANSPORT READY,
2467 7776 -2 /HC
2470 7342 T23BUF /CA
2471 7777 -1 /MEM
2472 0400 ODD /CM
2473 0000 T23FR1, 0 /FR
2474 3171 DCA EXPEOF
2475 4534 CONTNU /AS SOON AS CONTROL IS READY,
2476 0000 T23FR2, 0 /FR /ISSUE SECOND FUNCTION AND
/GO, WAIT A LITTLE FOR
2477 4545 WAIT2 /TRANSPORT READY,
2500 0001 1 /TRANSPORT SHOULD BE READY BY NOW,
2501 4540 TSKTR /0:1
2502 7000 NOP
2503 4535 TSKEF /1:1 THERE SHOULD BE AN ERROR,
2504 4541 TMS /2:1 EOF BIT SHOULD BE SET,
2505 4100 4100
2506 1162 TAD ERSTAT
2507 5655 JMP 1 T23EX1 /EXIT,
2510 0000 T23EX2, 0 /SERVICE FOR T23R, D AND E,
2511 1710 TAD I T23EX2 /GET FIRST FUNCTION
2512 3326 DCA T23FR3 /AND SAVE,
2513 2310 IS# T23EX2
2514 1710 TAD I T23EX2 /GET SECOND FUNCTION AND
2515 3331 DCA T23FR4 /SAVE,
2516 2310 IS# T23EX2 /UPDATE TO RETURN,
2517 4527 LOADPT /TO BOT,
2520 4530 SET1 /DO FIRST FUNCTION WHEN
2521 3400 /GD /TRANSPORT READY,
2522 7776 -2 /HC
2523 7342 T23BUF /CA
2524 7777 -1 /MEM
2525 0400 ODD /CM
2526 0000 T23FH3, 0 /FR
2527 7000 NOP
2530 4534 CONTNU /AS SOON AS CONTROL IS READY,
2531 0000 T23FR4, 0 /FR /ISSUE SECOND FUNCTION AND GO,
2532 4543 TNC /0:0 HC SHOULD NOT INCREMENT RIGHT AWAY,
2533 7777 -1

```

```

2534 4545      WAIT2      /WAIT FOR WC TO INC,
2535 0001      1
2536 4543      TMC      /111 WC SHOULD INC TO 0 BY NOW,
2537 0000      0
2540 4544      WAIT1      /WAIT FOR MTF,
2541 0001      1
2542 4536      TSKTD     /211 MTF SHOULD BE SET BY NOW,
2543 7000      NOP
2544 4544      WAIT1      /WAIT FOR TRANSPORT READY,
2545 0001      1
2546 4540      TSKTR     /311 TRANSPORT SHOULD BE READY BY NOW,
2547 7000      NOP
2550 4535      TSKEF     /410 THERE SHOULD BE NO ERROR,
2551 1162      TAD      ERSTAT
2552 5710      JMP I      T23EX2      /EXIT,
    
```

```

/TYPE SUBROUTINE,
/ENTER WITH ANSCII IN AC, EXIT WITH AC CLEAR
2553 0000      TYPE, 0
2554 3370      DCA      TYPTM      /SAVE ANSCII,
2555 6041      TSF
2556 7410      SKP
2557 5363      JMP      ,+4
2560 1173      TAD      TTOFLG
2561 7700      SMA CLA
2562 5355      JMP      ,+5
2563 3173      DCA      TTOFLG
2564 1370      TAD      TYPTM
2565 6046      TLA
2566 7200      CLA
2567 5753      JMP I      TYPE      /TYPE CHARACTER IF REAL TTY FLAG
2570 0000      TYPTM, 0      /IS SET OR SOFTWARE FLAG SET,

/RING TTY BELL,
2571 0000      BELL, 0
2572 7200      CLA
2573 1376      TAD      K207
2574 4467      JMS I      TYPEP
2575 5771      JMP I      BELL
2576 0207      K207, 207
    
```

/TEST24. CHANGE DIRECTION MODE TESTS.

```

2600 2600 PAGE
2600 0000 TEST24, 0 /CHANGE DIRECTION TESTS,

2601 4320 T24A. JMS T24EX1 /SPCFWD TO SPCREV, TWO 2 WORD
2602 4527 LOADPT /RECORDS WRITTEN FROM BOT, THEN
2603 4532 SET1 /BACK TO BOT, SPCFWD OVER
2604 5000 5000 /GD /RECORDS, CHANGE DIRECTION=
2605 7776 -2 /WC /SPCREV OVER RECORDS,
2606 7342 T25BUF /CA
2607 7777 -1 /MEM
2610 0400 ODD /CM
2611 6100 SPCFWD+GO /FR /SPCFWD
2612 7000 NOP
2613 4534 CONTNU
2614 7100 SPCREV+GO /FR /SPCREV
2615 4545 WAIT2 /WAIT FOR CONTROL READY,
2616 0001 1
2617 4537 TSKCR /0:1 CONTROL SHOULD BE READY BY NOW,
2620 7000 NOP
2621 4541 TMS /1:0 BOT SHOULD NOT BE SET YET,
2622 1000 1000
2623 4544 WAIT1 /WAIT FOR TRANSPORT READY,
2624 0001 1
2625 4540 TSKTR /2:1 TRANSPORT SHOULD BE READY BY NOW,
2626 7000 NOP
2627 4535 TSKEF /3:0 THERE SHOULD BE NO ERROR,
2630 1162 TAD ERSTAT
2631 4525 COMPAR
2632 6621 ER24A

/GD: GOOD ERSTAT; BD: REAL ERSTAT,
JMP T24A+1 /SUBTEST LOOP,
LOOP6
JMP T24A /*****LOOP6*****,
2637 4530 T24B. JMS T24EX1 /SPCREV TO WEOF, WRITE TWO
2640 7000 SET1 /2 WORD RECORDS FROM BOT,
2641 7776 7000 /THEN SPCREV OVER RECORDS,
2642 7342 -2 /CHANGE DIRECTION = WEOF,
2643 7777 T25BUF /CA
2644 0400 -1 /MEM
2645 7100 ODD /CM
2646 7000 SPCREV+GO /FR /SPCREV,
2647 4534 NOP
2650 5100 CONTNU
2651 4545 WEOF+GO /FR /WEOF,
2652 0001 WAIT2 /WAIT FOR TRANSPORT READY,
2653 4536 1
2654 7000 TSKTD /0:1 MTF SHOULD BE SET BY NOW
2655 4535 NOP
2656 4541 TSKEF /1:1 THERE SHOULD BE AN ERROR,
TMS /2:1 EOF BIT SHOULD BE SET,

```

```

2657 4100 4100
2660 1162 TAD ERSTAT
2661 4525 COMPAR
2662 6623 ER24B

/GD: GOOD ERSTAT; BD: REAL ERSTAT,
JMP T24B /SUBTEST LOOP,
LOOP7
JMP T24B-2 /*****LOOP 7*****,
2663 5236 T24C. JMS T24EX1 /SPCREV TO READ, WRITE TWO
2664 4522 SET1 /2 WORD RECORDS FROM BOT,
2665 5234 6000 /THEN SPCREV, CHANGE DIRECTION=
2666 4320 -2 /HEAD THE RECORDS,
2667 4530 /GD
2670 6000 /WC
2671 7776

2672 7342 T25BUF /CA
2673 7777 -1 /MEM
2674 0400 ODD /CM
2675 7100 SPCREV+GO /FR /SPCREV,
2676 7000 NOP
2677 4534 CONTNU
2678 2100 READ+GO /FR /READ,
2679 4544 WAIT1 /WAIT FOR CONTROL READY,
2682 0001 1
2683 4537 TSKCB /0:1 CONTROL SHOULD BE READY BY NOW,
2684 7000 NOP
2685 4543 TMS /1:1 WC SHOULD INDICATE 2 (READ 2 WORDS,)
2686 0000 0
2687 4535 TSKEF /2:0 THERE SHOULD BE NO ERROR,
2690 1162 TAD ERSTAT
2691 4525 COMPAR
2692 6625 ER24C

/GD: GOOD ERSTAT; BD: REAL ERSTAT,
JMP T24C /SUBTEST LOOP,
LOOP8
JMP T24C-2 /*****LOOP 8*****,
2693 5266 LOADPT
2694 4523 JMS I TEST24
2695 5264

/Routine to write 2 2 word records

2720 0000 T24EX1, 0
2721 4527 LOADPT
2722 7244 CLA STA RAL
2723 3336 DCA T24T1
2724 4530 SET1
2725 0000 0 /GD
2726 7776 -2 /WC
2727 7342 T25BUF /CA
2730 7777 -1 /MEM
2731 0400 ODD /CM
2732 4100 WHITE+GO /FR
2733 2336 152 T24T1
2734 5324 JMP T24EX1+4
2735 5720 JMP I T24EX1
2736 0000 T24T1, 0

```

```

/PDP8-E PACKED ANSCII MESSAGE GENERATOR.
/ENTERED WITH:
/JMS AMG8E
/MSGNP (MESSAGE POINTER)
/
/00 IS TERMINATOR, 43 (#) IS CARRIAGE RETURN, LINEFEED,
/THE FOLLOWING CHARACTERS ARE NOT ALLOWED: # WHICH GIVES
/TERMINATOR CODE, * WHICH RESULTS IN CRLF, AND 204-207,
/212, 219, 375, 377, 233,
/
/EXITS WITH AC CLEAR.
    
```

```

2737 0000 AMG8E, 0
2740 7300 CLA CLL
2741 1737 TAD I AMG8E
2742 3376 DCA AMG8E1
2743 2337 ISZ AMG8E
2744 1776 TAD I AMG8E1
2745 4404 BSW
2746 4353 JMS AMG8E2
2747 1776 TAD I AMG8E1
2750 4353 JMS AMG8E2
2751 2376 ISZ AMG8E1
2752 5344 JMP ,*6
2753 0000 AMG8E2, 0
2754 0027 AND K77
2755 7450 SNA
2756 5737 JMP I AMG8E
2757 3377 DCA AMG8E3
2760 1377 TAD AMG8E3
2761 1043 TAD M43
2762 7640 SZA CLA
2763 5366 JMP ,*3
2764 4465 JMS I CRLFP
2765 5753 JMP I AMG8E2
2766 1377 TAD AMG8E3
2767 1042 TAD M40
2770 7710 SPA CLA
2771 1030 TAD K100
2772 1031 TAD K200
2773 1377 TAD AMG8E3
2774 4467 JMS I TYPEP
2775 5753 JMP I AMG8E2
2776 0000 AMG8E1, 0
2777 0000 AMG8E3, 0
    
```

```

/TEST 25, BASIC DATA TEST. (ALL DATA WRITTEN AT 800 BPI, ODD PARITY,
/ CORE DUMP MODE FORCED IF 9 TRK.)
    
```

```

3000 3000 PAGE
3000 0000 TEST25, 0
3001 4227 JMS T25A /WRITE A 40 WORD RECORD OF ALL 1'S
3002 7777 7777 /FROM BOT, THEN READ AND
3003 0000 0000 /CHECK DATA,
3004 4521 LOOP6
3005 5201 JMP ,*4 /*****LOOP 6*****
3006 4227 JMS T25A /WRITE A 40 WORD RECORD OF ALL 0'S
3007 0000 0000 /FROM BOT, THEN READ AND
3008 0000 0000 /CHECK DATA,
3009 4522 LOOP7
3010 5204 JMP ,*6 /*****LOOP 7*****
3013 4227 JMS T25A /WRITE A 40 WORD RECORD OF 1'S
3014 7777 7777 /AND 0'S (7777,0000,7777,ETC)
3015 4000 4000 /FROM BOT, THEN READ AND
3016 4523 LOOP8 /CHECK DATA,
3017 5211 JMP ,*6 /*****LOOP 8*****
3020 4227 JMS T25A /WRITE A 40 WORD RECORD,
3021 5252 5252 /PATTERN=5252,2525,5252,ETC,
3022 4000 4000 /THEN READ AND CHECK DATA,
3023 4524 LOOP9
3024 5216 JMP ,*6 /*****LOOP 9*****
3025 4527 LOADPT
3026 5600 JMP I TEST25
3027 0000 T25A, 0
3030 7340 CLA CLL CMA
3031 1227 TAD T25A
3032 3150 DCA OLD
3033 1627 TAD I T25A /GET DATA STARTER,
3034 3375 DCA T25TM2 /SAVE IT IN BUFFER STARTER,
3035 2227 ISZ T25A
3036 1627 TAD I T25A /GET DATA SPECIFIER,
3037 7104 CLL RAL /MOVE IT INTO LINK,
3040 1375 TAD T25TM2 /GET DATA STARTER,
3041 7430 SEL /IF LINK SET, MAKE COMPLEMENTING PATTERN,
3042 7040 CMA /IF NOT, DO NOT COMPLEMENT,
3043 3153 DCA TXXTM3 /SAVE SAME OR COMPLEMENT DATA,
3044 2227 ISZ T25A /UPDATE TO RETURN,
3045 1377 T25ASL, TAD M50 /SET UP DATA BUFFER WITH DATA,
3046 3151 DCA TXXTM1
3047 1376 TAD T25BUP
3050 3152 DCA TXXTM2
3051 2152 ISZ TXXTM2
3052 1375 TAD T25TM2
    
```

```

3053 3552      DCA I  TXXTM2
3054 2152      ISZ  TXXTM2
3055 1153      TAD   TXXTM3
3056 3552      DCA I  TXXTM2
3057 2151      ISZ  TXXTM1
3060 2151      ISZ  TXXTM1          /BUFFER FULL?
3061 5251      JMP   ,=10          /NO, KEEP INSERTING DATA,
3062 4527      LOADPT          /YES, TAPE TO BOT,
3063 4531      SET2          /WRITE THE RECORD,
3064 4000      4000 /GD
3065 7726      -52 /WC
3066 7337      T25BUF-3/CA
3067 0000      0 /HEM
3070 0402      ODD+08007/CH
3071 4100      WRITE+GD/FR
3072 4545      WAIT2          /WAIT FOR JOB DONE
3073 0001      1
3074 4536      TSKTD /0:1 SKTD SHOULD SKIP BY NOW,
3075 7000      NOP
3076 4535      TSKEF /1:0 THERE SHOULD BE NO ERROR,
3077 1162      TAD   ERSTAT
3100 4525      COMPAR
3101 6627      ER25A
/GO: GOOD ERSTAT; BD: REAL ERSTAT; OD: DATA ROUTINE ADDRESS,
3102 5245      JMP   T25ASL /SUBTEST LOOP,

3103 7200      T25B, CLA
3104 1377      TAD   M50          /SET BUFFER UP TO COMPLEMENT DATA,
3105 3151      DCA  TXXTM1
3106 1376      TAD   T25BUP
3107 3152      DCA  TXXTM2
3110 2152      ISZ  TXXTM2
3111 1375      TAD   T25TM2
3112 7040      CMA
3113 3552      DCA I  TXXTM2
3114 2152      ISZ  TXXTM2
3115 1153      TAD   TXXTM3
3116 7040      CMA
3117 3552      DCA I  TXXTM2
3120 2151      ISZ  TXXTM1
3121 2151      ISZ  TXXTM1
3122 5310      JMP   ,=12          /DONE? NO, CONTINUE,
3123 4527      LOADPT          /YES, TAPE TO BOT,
3124 4531      SET2          /NOW READ THE RECORD,
3125 4000      4000 /GD
3126 7726      -52 /WC
3127 7337      T25BUF-3/CA
3130 2525      2525 /HEM
3131 0402      ODD+08007/CH
3132 2100      READ+GD/FR
3133 4545      WAIT2          /WAIT FOR JOB DONE
3134 0001      1
3135 4536      TSKTD /0:1 SKTD SHOULD SKIP BY NOW,
3136 7000      NOP
3137 4535      TSKEF /1:0 THERE SHOULD BE NO ERRORS,

```

```

3140 1162      TAD   ERSTAT
3141 4525      COMPAR
3142 6031      ER25B
/GO: GOOD ERSTAT; BD: REAL ERSTAT; OD: DATA ROUTINE ADDRESS,
3143 5303      JMP   T25B /SUBTEST LOOP

3144 7200      T25C, CLA          /NOW CHECK THE DATA READ
3145 1377      TAD   M50          /AGAINST THE DATA SPECIFIERS,
3146 3151      DCA  TXXTM1          /USE ACLOC TO INDICATE THE ADDRESS OF
3147 1376      TAD   T25BUP          /ANY FAILING DATA,
3150 3170      DCA  ACLOC
3151 2170      ISZ  ACLOC
3152 1375      TAD   T25TM2
3153 3146      DCA  GOOD
3154 1970      TAD I  ACLOC
3155 4367      JMS  T25CCK
3156 2170      ISZ  ACLOC
3157 1153      TAD   TXXTM3
3160 3146      DCA  GOOD
3161 1572      TAD I  ACLOC
3162 4367      JMS  T25CCK
3163 2151      ISZ  TXXTM1
3164 2151      ISZ  TXXTM1
3165 5351      JMP   ,=14
3166 5627      JMP I  T25A
T25CCK, 0
3167 0000      0
3170 4525      COMPAR
3171 6033      ER25C
/GO: GOOD DATA; BD: DATA READ; OD: DATA ROUTINE ADDRESS; AC: ADDRESS OF FAILING DATA,
3172 7000      NOP          /THERE IS NO SUBTEST LOOP FOR DATA CHECKING,
3173 5767      JMP I  T25CCK          /CONTINUE CHECKING DATA,
3174 5627      JMP I  T25A          /"EX" EXIT,
3175 0000      T25TM2, 0
3176 7341      T25BUP, T25BUF-1
3177 7730      M50, -50

```

```

/TEST26, CRCC TEST (9 TRACK ONLY),

3200 3200 PAGE
3200 0000 TEST26, 0 /CRCC TEST (9 TRACK ONLY);
3201 1161 TAD TRK9 /4 WORD RECORDS, EACH WORD
3202 7650 SNA CLA /IS IDENTICAL.
3203 5600 JMP 1 TEST26 /IMMEDIATE EXIT FOR 7 TRACK,
3204 4527 LOADPT /TO BDT

3205 1350 T26A, TAD N400 /SET UP FOR 400(8) DATA SETS,
3206 3151 DCA TXXTM1
3207 3253 DCA T26SC1 /DATA: 0 THRU 377 (ODD PARITY),
3210 1253 TAD T26SC1
3211 3150 DCA OLD
3212 3235 DCA PARCAR
3213 1253 TAD T26SC1 /GET PRESENT DATA AND GENERATE
3214 7110 CLL RAR /PARITY BIT,
3215 7430 SEL
3216 2235 ISZ PARCAR
3217 7440 SZA
3220 5214 JMP ,=4
3221 1235 TAD PARCAR
3222 7010 RAR
3223 7620 SNL CLA
3224 1033 TAD K400
3225 1253 TAD T26SC1
3226 3235 DCA PARCAR /SAVE DATA PLUS PARITY BIT,
3227 1235 TAD PARCAR /EFFECTIVELY XOR FIRST WORD
3230 3264 DCA CRC /INTC CRCM,
3231 4310 JMS CRCROT /ROTATE CMCR
3232 1351 TAD M3 /SET UP FOR 3 WORDS,
3233 3153 DCA TXXTM3 /
3234 4324 JMS CRXOR /XOR NEXT DATA WORD INTO CRCM,
3235 0000 PARCAR, 0
3236 4310 JMS CRCROT /ROTATE CRCM,
3237 2153 ISZ TXXTM3 /3 WORDS DONE?
3240 5234 JMP ,=4 /NO,
3241 4324 JMS CRXOR /YES, COMPLEMENT ALL BITS
3242 0727 T26ASL, TAD T26SC1 /EXCEPT 4 AND 8,
3243 1253 DCA 1 YBUFF+3 /PUT DATA IN BUFFER
3244 3741 TAD T26SC1
3245 1253 DCA 1 YBUFF+4
3246 3742 DCA 1 YBUFF+4
3247 4531 SET2 /WRITE 2 WORD RECORD,
3250 0000 0 /GD
3251 7774 =4 /MC
3252 7342 T2>BUF /CA
3253 0000 T26SC1, 0 /MEM
3254 0403 ODD+D8809/CM
3255 4100 WRITE+GD/FR
3256 4533 SET4 /SPACE REVERSE 1 RECORD
3257 0000 0 /GD
3260 7777 -1 /MC
3261 0403 ODD+D8809/CM
    
```

```

3262 7100 SPCREV+GD/FR
3263 4531 SET2 /HEAD THE LAST RECORD,

3264 0000 CRC, 0 /GD
3265 7774 =4 /MC /ENABLE CMCR TO BE READ IN
3266 7342 T2>BUF /CA /TO T2>BUF+5,
3267 0000 0 /MEM
3270 0403 ODD+D8809/CM
3271 2300 READ+ERLPCC+GD/FR
3272 4507 SKTD /WAIT FOR MTF
3273 5272 JMP ,=1
3274 1743 TAD 1 YBUFF+5 /GET CRC + CHECK
3275 4525 COMPAR
3276 6635 ER26A

/GO: GOOD CRC: BDI REAL CRC: ODI DATA USED
3277 5243 JMP T26ASL /SUBTEST LOOP,
3300 5303 JMP ,=3
3301 4527 LOADPT
3302 5600 JMP 1 TEST26
3303 7201 CLA IAC /UPDATE DATA,
3304 1253 TAD T26SC1
3305 2151 ISZ TXXTM1 /ALL DONE?
3306 5207 JMP T26A+2 /NO, DO NEXT DATA SET,
3307 5301 JMP ,=6 /YES, EXIT
3310 0000 CRCROT, 0 /CRC ROTATE,
3311 1264 TAD CRC /SET CRC AND MOVE ONE
3312 7110 CLL RAR /RIGHT,
3313 7420 SNL /IF BIT 1 SHOULD BE A 1,
3314 5322 JMP ,=6 /MAKE IT SO, IF NOT, EXIT,
3315 1033 TAD K400
3316 3264 DCA CRC
3317 4324 JMS CRXOR /IF A 1, COMPLEMENT BITS
3320 0074 74 /4,5,6,7,
3321 7410 SKP /SAVE CRC,
3322 3264 DCA CRC /EXIT,
3323 5710 JMP 1 CRCROT

3324 0000 CRCXOR, 0 /EXCLUSIVE OR CRC WITH CALL+1,
3325 1264 TAD CRC /GET CRC,
3326 0724 AND 1 CRXOR /COMPUTE CARRY,
3327 7104 CLL RAL /SCALE AND NEGATE,
3330 7041 CIA
3331 1264 TAD CRC /COMPUTE PARTIAL SUM,
3332 1724 TAD 1 CRXOR
3333 3264 DCA CRC /SAVE CRC,
3334 2324 ISZ CRXOR
3335 5724 JMP 1 CRXOR /EXIT,
3336 7342 YBUFF, T2>BUF /CRC TEST BUFFER,
3337 7343 T2>BUF+1 /1ST DATA
3340 7344 T2>BUF+2 /2ND DATA
3341 7345 T2>BUF+3 /3RD DATA
3342 7346 T2>BUF+4 /4TH DATA
3343 7347 T2>BUF+5 /CRCC
3344 7350 T2>BUF+6 /LPCC
3345 7351 T2>BUF+7
    
```

```

3346 7352          T2>BUF+10
3347 7353          T2>BUF+11
3350 7400      N400, -400
3351 7775      M3,   -3
    
```

```

/TEST27,          CORE DUMP/COMPATIBLE 7 TRACK TESTS;

3400 0000      PAGE
3401 1161      TEST27, 0          /CORE DUMP MODE TESTS,
3402 7650      TAD          TRK9
3403 5600      SNA CLA
          JMP I          TEST27          /IMMEDIATE EXIT FOR 7 TRACK

3404 4273      T27A,  JMS          T27EX1          /VERIFY WEOF WORKS IN
3405 0403      ODD+D8009          /COMPATIBLE MODE,          REF T27EX1
3406 4525      COMPAR          /FOR INFORMATION,
3407 6637      ER27A

/GD: GOOD ERSTAT; BD: REAL ERSTAT,
3410 5204      JMP          T27A          /SUBTEST LOOP,

3411 4273      T27B,  JMS          T27EX1          /VERIFY EOF WRITTEN IN CORE DUMP
3412 0402      ODD+D8007          /MODE IS NOT RECOGNIZED IN
3413 4317      JMS          T27EX2          /COMPATIBLE MODE,
3414 0403      ODD+D8009
3415 4525      COMPAR
3416 6641      ER27B

/GD: GOOD ERSTAT; BD: REAL ERSTAT,
3417 5211      JMP          T27B          /SUBTEST LOOP,

3420 4273      T27C,  JMS          T27EX1          /VERIFY EOF WRITTEN IN COMPATIBLE
3421 0403      ODD+D8009          /MODE IS NOT RECOGNIZED IN
3422 4317      JMS          T27EX2          /CORE DUMP MODE,
3423 0402      ODD+D8007
3424 4525      COMPAR
3425 6643      ER27C

/GD: GOOD ERSTAT; BD: REAL ERSTAT,
3426 5220      JMP          T27C          /SUBTEST LOOP,
3427 4521      LOOP6
3430 5204      JMP          T27A          /*****LOOP6*****/
3431 4527      T27D,  LOADPT          /VERIFY A 4 WORD RECORD
3432 7240      CLA CHA
3433 3455      DCA I          XBUFFER+4
3434 7240      CLA CHA
3435 3637      DCA I          ,*2          /WRITTEN IN COMPATIBLE
3436 4531      SET2          /MODE, READS 2 WORDS IN
3437 7345      T25BUF+3/GD          /CORE DUMP MODE,
3440 7774      -4          /HC
3441 7342      T25BUF          /CA
3442 7777      -1          /MEM
3443 0403      ODD+D8009/CM
3444 4100      WRITE+GD/FR
3445 4527      LOADPT
3446 4531      SET2
3447 6000      6000          /GD
3450 0000      0          /HC
3451 7342      T25BUF          /CA
3452 0000      0          /MEM
3453 0402      ODD+D8007/CM
3454 2100      READ+GD /FR
    
```

```

3455 4545      WAIT2          /WAIT FOR TRANSPORT READY,
3456 0001      1
3457 4540      TSKTR /0:1    TRANSPORT SHOULD BE READY BY NOW,
3460 7000      NOP
3461 4543      TWC /1:1    WC SHOULD INDICATE 2 WORDS READ,
3462 0002      2
3463 1162      TAD ERSTAT
3464 4525      COMPAR
3465 6645      ER27D
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
3466 5231      JMP T27D /SUBTEST LOOP,
3467 4922      LOOP7
3470 5227      JMP T27D-2 /*****LOOP 7****
3471 4927      LOADPT
3472 5000      JMP I TEST27

3473 0000      T27EX1, 0 /WEF USING DENSITY SELECTION
3474 1673      TAD I T27EX1 /IN CALL,
3475 3302      DCA ,+5
3476 4527      LOADPT
3477 4533      SET4
3500 7000      7000 /GD
3501 0000      0 /WC
3502 0000      0 /CH (DENSITY SELECTION INSERTED,)
3503 5100      WEOF+GO /FR
3504 3171      DCA EXPEOF
3505 4545      WAIT2          /WAIT FOR TRANSPORT READY,
3506 0001      1
3507 4540      TSKTR /0:1    TRANSPORT SHOULD BR READY BY NOW,
3510 7000      NOP
3511 4535      TSKEF /1:1    THERE SHOULD BE AN ERROR,
3512 4541      TMS /2:1    EOF BIT SHOULD BE SET
3513 4100      4100
3514 1162      TAD ERSTAT
3515 2273      ISZ T27EX1
3516 5673      JMP I T27EX1

3517 0000      T27EX2, 0 /SPACE REVERSE HOPEFULLY
3520 7200      CLA /THROUGH EOF TO BOT
3521 1717      TAD I T27EX2 /WHEN EOF WRITTEN IN ONE
3522 3326      DCA ,+4 /MODE AND SPACE REVERSE IN THE
3523 4533      SET4 /OTHER,
3524 6000      6000 /GD
3525 0000      0 /WC
3526 0000      0 /CH (DENSITY SELECTION INSERTED,)
3527 7100      SPCREV+GO /FR
3530 4545      WAIT2          /WAIT FOR TRANSPORT READY
3531 0001      1
3532 4540      TSKTR /0:1    TRANSPORT SHOULD BE READY BY NOW,
3533 7000      NOP
3534 4541      TMS /1:1    BOT SHOULD BE SET (SHOULD PASS EOF,)
3535 1000      1000
3536 1162      TAD ERSTAT
3537 2317      ISZ T27EX2
3540 5717      JMP I T27EX2

```

```

3541 0000      SRBSW, 0 /BYTE SWAP ROUTINE,
3542 7106      CLL RTL
3543 7006      RTL
3544 7006      RTL
3545 3353      DCA SRBSWT
3546 7004      RAL
3547 1353      TAD SRBSWT
3550 0027      AND K77
3551 1353      TAD SRBSWT
3552 5741      JMP I SRBSW
3553 0000      SRBSWT, 0

3554 0000      SRMQL, 0 /MO LOADER,
3555 3366      DCA SRMQLT
3556 5754      JMP I SRMQL

3557 0000      SRMQA, 0 /INCLUSIVE OR (MOA),
3560 3394      DCA SRMQL
3561 1354      TAD SRMQL
3562 7040      CMA
3563 0366      AND SRMQLT
3564 1354      TAD SRMQL
3565 5757      JMP I SRMQA
3566 0000      SRMQLT, 0

```

/TEST30. MANUAL INTERVENTION TESTS;

```

3600 3600 PAGE
      2000 TEST30, 0 /MANUAL INTERVENTION TEST,

3601 4460 T30A, JMS 1 AMG8EP /VERIFY OFFLIN ACTUALLY
3602 7137 MTH2 /PUTS DRIVE OFF LINE,
3603 4357 JMS NSTRUC
3604 7043 MSG61
3605 4527 LOADPT /TO BOT,
3606 4530 SET1 /WRITE A RECORD, (GET
3607 0000 0 /GD /AWAY FROM BOT,
3610 7760 -20 /WC
3611 7342 T29BUF /CA
3612 7777 -1 /MEM
3613 0400 ODD /CH
3614 4100 WRITE+GO/FR
3615 4532 SET3 /COMMAND OFF LINE,
3616 1300 1300 /GD
3617 0000 0 /WC
3620 3400 ODD /CH
3621 3100 OFFLIN+GO/FR
3622 4535 TSKEF /0:0 THERE SHOULD BE NO ERROR,
3623 4541 TMS /1:0 RWSTAT SHOULD NOT BE SET,
3624 2000 2000
3625 4541 TMS /2:1 SELECT REMOTE SHOULD BE SET,
3626 0400 0400
3627 4535 TSKEF /3:0 THERE SHOULD BE NO ERROR,
3630 4536 TSKTD /4:1 MTF SHOULD BE SET,
3631 4545 WAIT2 /STALL,
3632 0000 0
3633 4541 TMS /5:1 SELECT REMOTE SHOULD STILL BE SET,
3634 0400 0400
3635 1162 TAD ERSTAT
3636 4525 COMPAR
3637 6647 ER30A
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
3640 7000 NOP
3641 4357 JMS NSTRUC /ASK FOR VISUAL OFF LINE CHECK,
3642 7143 MTH3
3643 4520 LOOPS
3644 5201 JMP T30A /SUBTEST LOOP,

3645 4357 T30B, JMS NSTRUC /VERIFY TRANSPORT
3646 7166 MTH4 /NOT READY WITH NO VACUUM,
3647 3146 DCA GOOD
3650 4510 SKTR
3651 7610 SKP CLA /SKTR SHOULD NOT SKIP,
3652 7040 CMA
3653 4525 COMPAR
    
```

```

3654 6651 ER30B
/GD: GOOD AC; BD: REAL AC,
3655 5250 JMP T30B+3 /SUBTEST LOOP,

3656 4357 T30C, JMS NSTRUC /VERIFY WRITE AND FILE PROT
3657 7206 MTH5 /YIELD IF AND EF,
3660 4532 SET3
3661 4000 4000 /GD
3662 0000 0 /WC
3663 0400 ODD /CH
3664 4000 WRITE /FR
3665 4541 TMS /0:1 IF, FILE PROT AND EF SHOULD BE SET,
3666 4005 4005
3667 1162 TAD ERSTAT
3670 4525 COMPAR
3671 6653 ER30C
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
3672 5260 JMP T30C+2 /SUBTEST LOOP,

3673 4532 T30D, SET3 /VERIFY WEOF AND FILE PROT
3674 4000 4000 /GD /YIELD IF AND EF,
3675 0000 0 /WC
3676 0400 ODD /CH
3677 5000 WEOF /FR
3700 4541 TMS /0:1 IF, FILE PROT AND EF SHOULD BE SET,
3701 4005 4005
3702 1162 TAD ERSTAT
3703 4525 COMPAR

3704 6655 ER30D
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
3705 5273 JMP T30D /SUBTEST LOOP,

3706 4532 T30E, SET3 /VERIFY A FUNCTION OTHER THAN
3707 0000 0 /GD /WRITE OR WEOF DOES NOT
3710 7777 -1 /WC /CAUSE IF AND EF WITH
3711 0400 ODD /CH /FILE PROT,
3712 3000 RDCOMP /FR /READ/COMPARE USED,
3713 4541 TMS /0:0 IF AND EF SHOULD NOT BE SET,
3714 4001 4001
3715 1162 TAD ERSTAT
3716 4525 COMPAR
3717 6657 ER30E
/GD: GOOD ERSTAT; BD: REAL ERSTAT,
3720 5306 JMP T30E /SUBTEST LOOP,

3721 4527 T30F, LOADPT /VERIFY ERROR FLAG SETS WHEN DRIVE PUT OFF LINE,
3722 4357 JMS NSTRUC /
3723 7043 MSG61
3724 4357 JMS NSTRUC
3725 7342 MTH6
3726 4530 SET1
3727 4000 T30FOK, 4000 /GD
3730 0000 0 /WC
    
```

```

3731 7342      T25BUF /CA
3732 7244      =534 /MEM
3733 0480      ODD /CM
3734 4100      WRITE+GO/FR /WRITE FUNCTION USED,
3735 4472      LWCR /WAIT AT MOST 35-40 SECONDS DURING WRITE,
3736 4501      RMSR
3737 7510      SPA
3740 5345      JMP ,+5
3741 2452      ISZ I XBUFF+1
3742 5335      JMP ,+5
3743 2453      ISZ I XBUFF+2
3744 5335      JMP ,+7
3745 0327      AND T30FOK
3746 4525      COMPAR
3747 6661      ER30F
/GO! GOOD MAIN STATUS; BDI REAL MAIN STATUS,
3750 5322      JMP T30F+1 /SUBTEST LOOP,
3751 5752      JMP I ,+1
3752 4000      T30G

3753 4357      T30FIN, JMS NSTRUC /CLOSE OUT AND EXIT,
3754 7043      MSG61
3755 4927      LOADPT
3756 5600      JMP I TEST30

3757 0000      NSTRUC, 0 /ROUTINE TO INSTRUCT USER
3760 7300      CLA CLL /THEN GO TO MONITOR TO
3761 1757      TAD I NSTRUC /AWAIT COMPLETION OF ACTION,

3762 3365      DCA ,+3
3763 2357      ISZ NSTRUC
3764 4460      JMS I AMGBEP
3765 0000      0
3766 6082      IOF
3767 1375      TAD NSTRP
3770 3000      DCA 0
3771 5772      JMP I ,+1
3772 4603      MONIT+3
3773 4515      NSTR, CLEAR1
3774 5757      JMP I NSTRUC
3775 3773      NSTRP, NSTR

4000          PAGE
4000 4460      T30G, JMS I AMGBEP /VERIFY THAT THE "START" OR "CLEAR"
4001 7303      MTH7 /KEY CAUSES "INITIALIZE"
4002 7240      CLA CHA /TO FUNCTION IN THE TM8E
4003 4472      LWCR /CONTROL BY CHECKING WC CLEARED,
4004 3146      DCA GOOD
4005 7402      HLT /PDP8/E USERS MAY INSERT 10T 6007 HERE
4006 7240      CLA CHA
4007 3173      DCA TTOFLG
4010 4477      RWCR
4011 4925      COMPAR
4012 6663      ER30G
/GO! GOOD WC.

```

```

4013 5202      JMP ,+11 /SUBTEST LOOP,
4014 5615      JMP I ,+1
4015 3753      T30FIN

```

/TMBE IOT SUBROUTINES.

/ALL TMBE IOT'S ARE PLACED IN SUBROUTINES
 /TO ENABLE EASE IN CHANGING THE IOT DEVICE CODE
 /SHOULD THE DEVICE CODES NOT BE 70-72,
 /IF AN IOT SKIPS WHICH SHOULD NEVER SKIP, IT RESULTS
 /IN ER31B (EXCEPT RWCR),

4016	0000	SLWCR,	0		
4017	6701		6701		/LWCR
4020	5616		JMP I	SLWCR	
4021	4336		JMS	SKIPR	
4022	0000	SLCAR,	0		
4023	6703		6703		/LCAR
4024	5622		JMP I	SLCAR	
4025	4336		JMS	SKIPR	
4026	0000	SLCMR,	0		
4027	6705		6705		/LCMR
4030	5626		JMP I	SLCMR	
4031	4336		JMS	SKIPR	
4032	0000	SLFGR,	0		
4033	6706		6706		/LFGR
4034	5632		JMP I	SLFGR	
4035	4336		JMS	SKIPR	
4036	0000	SLDBR,	0		
4037	6707		6707		/LDBR
4040	5636		JMP I	SLDBR	
4041	4336		JMS	SKIPR	
4042	0000	SRWCR,	0		
4043	6711		6711		
4044	5642		JMP I	SRWCR	
4045	4336		JMS	SKIPR	
4046	0000	SRCAR,	0		
4047	6713		6713		/RCAR
4050	5646		JMP I	SRCAR	
4051	4336		JMS	SKIPR	
4052	0000	SRMSR,	0		
4053	6714		6714		/RMSR
4054	5652		JMP I	SRMSR	
4055	4336		JMS	SKIPR	
4056	0000	SRCHR,	0		
4057	6715		6715		/RCHR
4060	5656		JMP I	SRCHR	
4061	4336		JMS	SKIPR	
4062	0000	SRFSR,	0		
4063	6716		6716		/RFSR
4064	5662		JMP I	SRFSR	
4065	4336		JMS	SKIPR	
4066	0000	SRDBR,	0		
4067	6717		6717		/RDBR
4070	5666		JMP I	SRDBR	
4071	4336		JMS	SKIPR	
4072	0000	SSKEF,	0		
4073	6721		6721		/SKEF

4074	5672		JMP I	SSKEF	
4075	2272		ISZ	SSKEF	
4076	5672		JMP I	SSKEF	
4077	0000	SSKCB,	0		
4100	6722		6722		/SKCB
4101	5677		JMP I	SSKCB	
4102	2277		ISZ	SSKCB	
4103	5677		JMP I	SSKCB	
4104	0000	SSKTD,	0		
4105	6723		6723		/SKTD
4106	5704		JMP I	SSKTD	
4107	2304		ISZ	SSKTD	
4110	5704		JMP I	SSKTD	
4111	0000	SSKTR,	0		
4112	6724		6724		/SKTR
4113	5711		JMP I	SSKTR	
4114	2311		ISZ	SSKTR	
4115	5711		JMP I	SSKTR	
4116	0000	SCLF,	0		
4117	6725		6725		/CLF
4120	5716		JMP I	SCLF	
4121	4336		JMS	SKIPR	
4122	0000	SCLT,	0		
4123	6712		6712		/CLT
4124	5722		JMP I	SCLT	
4125	4336		JMS	SKIPR	
4126	0000	SSDLE,	0		
4127	6726		6726		/SDLE
4130	5726		JMP I	SSDLE	
4131	4336		JMS	SKIPR	
4132	0000	SSBRM,	0		
4133	6727		6727		/SBRM
4134	5732		JMP I	SSBRM	
4135	4336		JMS	SKIPR	
4136	0000	SKIPR,	0		/AN IOT SKIPPED WHICH SHOULD NEVER SKIP, /SAVE AC AT TIME OF FAILURE, /GET POINTER TO IOT SUBROUTINE, /ENTRY AND SAVE.
4137	3170		DCA	ACLOC	
4140	1037		TAD	M4	
4141	1336		TAD	SKIPR	
4142	3336		DCA	SKIPR	
4143	7001		JAC		/GET POINTER TO IOT CODE.
4144	1336		TAD	SKIPR	
4145	3147		DCA	BAD	
4146	1547		TAD I	BAD	/GET IOT CODE AND SAVE.
4147	3147		DCA	BAD	
4150	1736		TAD I	SKIPR	/GET RETURN TO MAIN PROGRAM.
4151	3336		DCA	SKIPR	/SAVE FOR EXIT.
4152	1147		TAD	BAD	/PUT IOT IN EXTERNAL SCOPE LOOP.
4153	3364		DCA	,+11	
4154	4526		ERROR		/GO TO ERRORS.
4155	6453		ER31B		
4156	5363	/BD: FAILING IOT CODE; ACI CONTENTS OF AC AT TIME OF FAILURE,	JMP	,+5	/EXECUTE IF SR5=1,
4157	4520		LOOP5		/IF NOT CHECK SR5,
4160	5363		JMP	,+3	/EXECUTE IF SR5=1,

```

4161 1170      TAD ACLOC      /NO SCOPE LOOP, EXIT TO CORRECT
4162 5736      JMP I SKIPER  /POINT IN MAIN PROGRAM,
4163 1170      TAD ACLOC      /SCOPE LOOP, SET UP AC, THIS
4164 3020      0           /MAY OR MAY NOT BE CORRECT CONDITION,
4165 5357      JMP ,=6      /NOT CODE, NO FAILURE, CHECK SR5,
4166 5354      JMP ,=12     /FAILURE, CALL EPROHS,
    
```

/LITTLE TEST ROUTINES.

/EACH TEST ROUTINE CHECKS FOR A SPECIFIC CONDITION, IN
 /GENERAL IF A SKIP IS BEING TESTED, A BIT WILL BE SET IN ERSTAT
 /IF THE INSTRUCTION SKIPS, IF DATA IS BEING TESTED, THE BIT IS
 /SET IF THE DATA MATCHES THE COMPARATOR OR IN SOME CASES
 /IF AT LEAST THOSE BITS ARE SET AS INDICATED BY THE COMPARATOR,

```

4200 4200      PAGE
4200 0000      TSKEFR, 0           /IF SKEF SKIPS, SET A BIT
4201 4505      SKEF           /IN ERSTAT,
4202 7010      SKP CLA
4203 7201      CLA IAC
4204 4270      JMS ERFX
4205 5000      JMP I TSKEFR
4206 0000      TSKTOR, 0           /IF SKTD SKIPS, SET A BIT
4207 4507      SKTD           /IN ERSTAT,
4210 7010      SKP CLA
4211 7201      CLA IAC
4212 4270      JMS ERFX
4213 5000      JMP I TSKTOR
4214 0000      TSKCBR, 0           /IF SKCB SKIPS, SET A BIT
4215 4506      SKCB           /IN ERSTAT,
4216 7010      SKP CLA
4217 7201      CLA IAC
4220 4270      JMS ERFX
4221 5014      JMP I TSKCBR
4222 0000      TSKTHR, 0           /IF SKTR SKIPS, SET A BIT
4223 4510      SKTR           /IN ERSTAT,
4224 7010      SKP CLA
4225 7201      CLA IAC
4226 4270      JMS ERFX
4227 5022      JMP I TSKTRR
4230 0000      TMSR, 0
4231 4501      RMSR
4232 0630      AND I TMSR           /IF MS HAS AT LEAST THOSE
4233 7041      CIA           /BITS SET AS IN THE CONSTANT,
4234 1030      TAD I TMSR           /SET A BIT IN ERSTAT,
4235 2230      ISE TMSR
4236 7040      SEA CLA
4237 7410      SKP
4240 7001      IAC
4241 4270      JMS ERFX
4242 5030      JMP I TMSR
4243 0000      TFSR, 0
4244 4503      RFSR
4245 0643      AND I TFSR           /IF FS HAS AT LEAST THOSE
4246 7041      CIA           /BITS SET AS IN THE CONSTANT,
4247 1043      TAD I TFSR           /SET A BIT IN ERSTAT,
4250 2243      ISE TFSR
4251 7040      SEA CLA
4252 7410      SKP
4253 7001      IAC
    
```

```

4254 4270 JMS ERFX
4255 5643 JMP I TFSR

4256 0000 TWCR, 0 /IF W0=CONSTANT, SET A
4257 4477 RWCR /BIT IN ERSTAT,
4260 7041 CIA
4261 1656 TAD I TWCR
4262 7640 SZA CLA
4263 7410 SKP
4264 7001 IAC
4265 4270 JMS ERFX
4266 2256 ISZ TWCR
4267 5656 JMP I TWCR

4270 0000 ERFX, 0 /UPON ENTHY, A 1 IN AC 11
4271 3327 DCA ERFX1 /SETS THE CORRECT BIT IN ERSTAT,
4272 1327 TAD ERFX1 /A 0 IN AC 11 CLEARS THE CORRECT
4273 4314 JMS ERSHFT /BIT, BIT POSITION IS DETERMINED
4274 3326 DCA ERFX0 /BY THE CURRENT VALUE OF ERTAL,
4275 7001 IAC
4276 7040 CMA
4277 4314 JMS ERSHFT
4300 0162 AND ERSTAT
4301 1326 TAD ERFX0
4302 3162 DCA ERSTAT
4303 1164 TAD ERTAL
4304 1157 TAD SLKNST
4305 7640 SZA CLA
4306 5311 JMP ,+3
4307 4515 CLEAR1
4310 5560 JMP I SLADDR
4311 2164 ISZ ERTAL
4312 7000 NOP
4313 5670 JMP I ERFX
4314 0000 ERSHFT, 0
4315 3330 DCA ERFX2
4316 1164 TAD ERTAL
4317 3331 DCA ERFX3
4320 1330 TAD ERFX2
4321 2331 ISZ ERFX3
4322 7410 SKP I ERSHFT
4323 5714 JMP I ERSHFT
4324 7104 CLL RAL
4325 5321 JMP ,=4
4326 0000 ERFX0, 0
4327 0000 ERFX1, 0
4330 0000 ERFX2, 0
4331 0000 ERFX3, 0

```

```

/ROUTINES TO CLEAR ALL FLAGS SAFELY,
4332 0000 CLR1, 0 /CLEAN1
4333 7302 CLA CLL /ENSURE TTD SOFTWARE FLAG
4334 1173 TAD TTOPFLG /IS SET PRIOR TO CAF,
4335 7640 SZA CLA /GENERATE "INITIALIZE",
4336 5341 JMP ,+3
4337 6041 TSF
4340 5334 JMP ,=4
4341 7340 CLA CMA CLL
4342 3173 DCA TTOPFLG
4343 6002 IOF
4344 4512 CLT
4345 4476 LDBR
4346 4366 JMS CLRX
4347 7240 CLA CMA
4350 3165 DCA ALTENA
4351 6001 ION
4352 5732 JMP I CLR1
4353 0000 CLR4, 0
4354 4357 JMS CLR5
4355 3165 DCA ALTENA
4356 5753 JMP I CLR4
4357 0000 CLR5, 0
4360 7300 CLA CLL
4361 4511 CLF
4362 4366 JMS CLRX
4363 7240 CLA CMA
4364 3165 DCA ALTENA
4365 5757 JMP I CLR5

4366 0000 CLRX, 0
4367 1163 TAD KCMD
4370 4474 LCMR
4371 3162 DCA ERSTAT
4372 1041 TAD M14
4373 3164 DCA ERTAL
4374 5766 JMP I CLRX
4375 0000 SCDF, 0 /HOMING CDF TO PF,
4376 0000 SCDF1, 0/CDF PF
4377 5775 JMP I SCDF

```

/EXECUTIVE

/USING A TEST STATUS WORD, "TSTAT", EXEC DETERMINES WHICH
 /TESTS ARE TO BE RUN. EACH BIT IN "TSTAT" REPRESENTS ONE
 /TEST. (BIT N REPRESENTS TEST N+15 IN OCTAL.)
 /"TSTAT" IS AUTOMATICALLY SET TO 7777 WHEN THE PROGRAM IS
 /STARTED AT 200, WITH SR0=0. IF SR0=1 AT TEST START TIME,
 /USER MUST SELECT TESTS WITH MONITOR. MONITOR THEN SETS THE
 /SELECTED TEST BITS AND ENTERS EXEC AT REX. ALL TESTS
 /WILL BE RUN THE NUMBER OF TIMES SPECIFIED BY THEIR
 /RESPECTIVE TALLY CONSTANTS UNLESS QUICK VERIFY IS SET,
 /WHEN ALL TESTS ARE COMPLETE, EXEC GOES TO MONITOR,

4400		PAGE	
4400 6224	EXEC.	RIF	/GET INSTRUCTION FIELD,
4401 3156		DCA	/MODIFY HOMING CDF ROUTINE,
4402 1156		TAD	PRGFLD
4403 1050		TAD	KCDF
4404 3741		DCA I	SCDFIP
4405 6201		CDF	00
4406 1340		TAD	KRMF
4407 3742		DCA I	P1
4410 1335		TAD	KJMPI3
4411 3743		DCA I	P2
4412 1336		TAD	INTSEP
4413 3744		DCA I	P3
4414 4457		JMS I	SCDFP
4415 3157		DCA	SLKNST
4416 3174		DCA	EXITFL
4417 7240		CLA CMA	/CLEAN ERROR EXIT FLAG,
4420 3173		DCA	TTOFLG
4421 7430		SZL	
4422 5331		JMP	NOEXLP+2
4423 4460		JMS I	AMGBEP
4424 6776		MSG1	/PRINT TITLE MESSAGE
4425 4760	INIT.	JMS I	EXECFP
4426 3177		DCA	TSTNUM
4427 5331		JMP	NOEXLP+2
4430 3176	REX.	DCA	TSTAT
4431 3174		DCA	EXITFL
4432 7240		CLA CMA	/CLEAN ERROR EXIT FLAG
4433 3175		DCA	ACTFLG
4434 3337		DCA	PASCNT
4435 1041	EXECL2.	TAD	M14
4436 3345		DCA	EXTAL
4437 1356		TAD	TSTP
4440 3346		DCA	TSTPP
4441 1357		TAD	TALP
4442 3347		DCA	TALPP
4443 1024		TAD	M14
4444 3177		DCA	TSTNUM
4445 1176		TAD	TSTAT
4446 3350		DCA	TSTAEX

4447 1350	EXECL1.	TAD	TSTAEX	/CHECK FOR A TEST BIT SET
4450 7024		RAL		
4451 3350		DCA	TSTAEX	/SAVE TEST STATUS ROTATED,
4452 2346		IS2	TSTPP	/UPDATE POINTERS
4453 2347		IS4	TALPP	
4454 2177		IS2	TSTNUM	/UPDATE TEST NUMBER
4455 7420		SNL		/RUN THIS TEST
4456 5306		JMP	NOTSRN	/NO
4457 7604		LAS		/YES TEST PROGRESS REPORT?
4460 0362		AND	K2000	
4461 7640		SZA CLA		
4462 5270		JMP	NOTSPR	/NO
4463 1177		TAD	TSTNUM	/YES, GET TEST NUMBER
4464 4462		JMS I	EDITP	/INSERT IN MSG AND PRINT
4465 3751		DCA I	M7P1P	
4466 4460		JMS I	AMGBEP	
4467 7056		MSG7		
4470 1747	NOTSPR.	TAD I	TALPP	/GET THIS TESTS TALLY AND SAVE
4471 3352		DCA	TSTAL	
4472 1746		TAD I	TSTPP	/GET TEST STARTING ADDRESS
4473 3353		DCA	EXTEMP	
4474 4753	EXOMT.	JMS I	EXTEMP	/RUN TEST
4475 7604		LAS		/LOOP ON CURRENT TEST
4476 0020		AND	K2	
4477 7640		SZA CLA		
4500 5274		JMP	EXOMT	/YES RUN TEST AGAIN
4501 7604		LAS		/SR0 SET? (QUICK VERIFY)
4502 7710		SPA CLA		/NO, GO CHECK TEST TALLY
4503 5306		JMP	NOTSRN	/YES SEE IF ALL TESTS RUN
4504 2352		IS2	TSTAL	/NO DECREASE TALLY, TEST DONE?
4505 5274		JMP	EXOMT	/NO RUN AGAIN
4506 2345	NOTSRN.	IS2	EXTAL	/YES 12 TESTS CHECKED?
4507 5247		JMP	EXECL1	/NO CHECK FOR NEXT TEST
4510 3177		DCA	TSTNUM	
4511 7604		LAS		/YES. LOOP ON ALL SELECTED TESTS
4512 0361		AND	K1	
4513 7650		SNA CLA		
4514 5327		JMP	NOEXLP	/NO
4515 2337		IS4	PASCNT	/YES UPDATE PASS COUNTER
4516 1337		TAD	PASCNT	/INSERT IN MSG AND
4517 4462		JMS I	EDITP	/PRINT
4520 3754		DCA I	M7P2P	
4521 1463		TAD I	EDTEMP	
4522 3753		DCA I	M7P1P	
4523 4460		JMS I	AMGBEP	
4524 7076		MSG9		
4525 4464		JMS I	BELLP	/BELL
4526 5235		JMP	EXECL2	/RUN ALL OVER AGAIN
4527 4460	NOEXLP.	JMS I	AMGBEP	/PRINT DONE MESSAGE,
4530 7063		MSG8		
4531 7240		CLA CMA		
4532 3173		DCA	TTOFLG	
4533 3175		DCA	ACTFLG	/CLEAN TEST IN PROGRESS,
4534 5461		JMP I	MONITP	/GO TO MONITOR

```

4535 5493 KJMP13, 5403
4536 5600 INTSEP, INTSEV
4537 2000 PASCNT, 0
4540 6244 KRMF, RMF
4541 4376 SCDFIP, SCDF1
4542 0001 P1, 1
4543 0092 P2, 2
4544 0003 P3, 3
4545 0000 EXTAL, 0
4546 0000 TSTPP, 0
4547 0000 TALPP, 0
4550 0000 TSTALX, 0
4551 7061 M7P1P, M7P1
4552 0000 TSTIAL, 0
4553 0000 EXTAMP, 0
4554 7102 M9P2P, M9P2
4555 7101 M9P1P, M9P1
4556 6420 TSTP, TST=1

4557 6434 TALP, TAL=1
4560 6000 EXECFP, EXECFX
4561 0001 K1, 1
4562 2000 K2000, 2000
    
```

/MONITOR

```

/1, ENTERED UNDER ONE OF THE FOLLOWING CONDITIONS:
/1.1, PROGRAM STARTED AT 200 WITH SH3=1,
/1.2, ERROR OCCURS WITH SR2=0,
/1.3, ANY FATAL ERROR OCCURS,
/1.4, "ALTMODE" IS STRUCK ON THE KEYBOARD AFTER TEST START,
/1.5, SELECTED TESTS ARE RUN TO COMPLETION.
    
```

```

4600 4600 PAGE
4600 6002 MONIT, IOF
4601 4767 JMS I SAVEPP /INTERRUPT SYSTEM OFF,
4602 3172 DCA TRACE /SAVE SUBROUTINE ONFO,
4603 3157 DCA SLKNST
4604 4460 JMS I AMG8EP
4605 7116 MSG13
4606 3366 DCA TSTATM /PRINT "M",
4607 4772 JMS I LISNP /CLEAN TEST STATUS TEMP,
4610 7475 =303 /GET KEYBOARD,
4611 4625 CS /C STRING,
4612 7454 =324 /I STRING,
4613 4642 TS /E STRING,
4614 7473 =305
4615 4744 ES
4616 7474 =304
4617 5000 DS /"D" LOOK AT MEMORY,
4620 7467 =311
4621 4774 IS /REPEAT DIALOGUE,
4622 7455 =323 /SCOPE LOOP STRING,
4623 5071 SL
4624 0000 0
4625 3364 CS, DCA TSTEM1 /"C" OR "EX", PUT AC IN TEMP,
4626 1175 TAD ACTFLG /IF NO TEST IN PROGRESS; SYNTAX
4627 7700 SMA CLA /ERROR,
4630 9771 JMP I QUESP
4631 4772 JMS I LISNP /GET KEYBOARD,
4632 7563 =215 /"M",
4633 4635 =2
4634 0000 0
4635 4405 JMS I CRLFP /"C" OR "EX",
4636 4770 JMS I RESTPP /RESTORE SUBROUTINES,
4637 1364 TAD TSTEM1 /GET TEMP AND PUT IN EXIT
4640 3174 DCA EXITFL /GLAG,
4641 5773 JMP I INTOKP /EXIT TO INTERRUPT SERVICE,
4642 4772 TS, JMS I LISNP /I STRING, GET KEYBOARD,
4643 7456 =322
4644 4700 TR /"TR"
4645 7477 =301
4646 4727 TA /"TA"
4647 0001 1
4650 4652 =+2 /"TN"
4651 0000 0
4652 4352 JMS GET2N
4653 1041 TAD M14
    
```

4654	7041	CIA		
4655	3364	DCA	TSTEM1	/SUM TO ISZ SENSITIVE COUNTER,
4656	7120	CLL CML		/DETERMINE TEST BIT,
4657	7010	RAR		
4660	2364	ISZ	TSTEM1	
4661	5257	JMP	,+2	
4662	3364	DCA	TSTEM1	/SAVE TEST BIT,
4663	1364	TAD	TSTEM1	/EXCLUSIVE OR NEW TEST BIT
4664	0366	AND	TSTATM	/WITH OLD TEST STATUS,
4665	7104	CLL RAL		
4666	7041	CIA		
4667	1364	TAD	TSTEM1	
4670	1366	TAD	TSTATM	/SAVE NEW TEST STATUS,
4671	3366	DCA	TSTATM	
4672	4772	JMS I	LISNP	/GET INPUT;
4673	7563	-215		/"TNN="
4674	4712	TCR		
4675	7454	-324		/"TNNT="
4676	4642	TS		/GO BACK TO T STRING;
4677	0000	0		
TR,	4700	CLA CMA		/"TR" TRACE REQUESTED,
4701	3172	DCA	TRACE	/SET TRACE FLAG,
4702	4772	JMS I	LISNP	/GET KEYBOARD,
4703	7475	-303		
4704	4625	CS		/"TRC"
4705	7454	-324		
4706	4642	TS		/"TRT"
4707	7473	-305		
4710	4744	ES		/"TRE"
4711	0000	0		
TCR,	4712	TAD	KCMD	
4713	4474	LCHR		
4714	4501	RMSR		
4715	0034	AND	K1000	/MUST BE AT ROT TO EXIT,
4716	7640	SEA CLA		
4717	5323	JMP	,+4	
4720	4460	JMS I	AMGSEP	
4721	7130	MSG17		
4722	5203	JMP	MONIT+3	
4723	4915	CLEAR1		/NULL MAGTAPE INTERRUPTS, ION,
4724	4465	JMS I	CRLF	
4725	1366	TAD	TSTATM	/PUT TEST STATUS IN AC;
4726	5765	JMP I	REXP	/GO TO "EXEC,"
TA,	4727	CLA CMA		/SET ALL TESTS;
4730	3366	DCA	TSTATM	
4731	4772	JMS I	LISNP	/GET KEYBOARD,
4732	7454	-324		
4733	4642	TS		/"TAT"
4734	7563	-215		
4735	4712	TCR		/"TA="
4736	7450	-330		
4737	4741	,+2		
4740	0000	0		
4741	1036	TAD	M2	/TAX
4742	3366	DCA	TSTATM	

4743	5331	JMP	TA+2	
4744	4772	ES,	JMS I	LISNP
4745	7450	-330		/E STRING,
4746	4750	,+2		/"EX"
4747	0000	0		
4750	7240	CLA CMA		/SET ERROR EXIT FLAG;
4751	5225	JMP	CS	/EXIT VIA C STRING;
4752	0000	GET2N,	0	/ENTERED WITH THE FIRST NUMBER
4753	7104	CLL RAL		/IN AC RIGHT JUSTIFIED;
4754	7006	RTL		/SCALE 3 LEFT,
4755	3364	DCA	TSTEM1	
4756	4772	JMS I	LISNP	/LOOK FOR ANOTHER OCATL NUMBER,
4757	0001	1		
4760	4762	,+2		
4761	0000	0		
4762	1364	TAD	TSTEM1	
4763	5752	JMP I	GET2N	/ADD 2 NUMBERS AND EXIT,
4764	0000	TSTEM1,	0	
4765	4430	REXP,	0	
4766	0000	TSTATM,	0	
4767	5662	SAVEPP,	SAVEP	
4770	5703	RESTP,	RESTP	
4771	5106	QUESP,	QUES	
4772	5112	LISNP,	LISN	
4773	5653	INTOKP,	INTOK	
4774	4772	IS,	JMS I	LISNP
4775	7563	-215		
4776	4425	INIT		/REINITIALIZE,
4777	0000	0		
5000	1037	DS,	TAD	M4
5001	3270	DCA	DS2	/SET FOR 4 NUMBERS;
5002	4312	JMS	LISN	/LOOK FOR FIELD NUMBER;
5003	0001	1		
5004	5006	,+2		
5005	0000	0		
5006	7104	CLL RAL		/SCALE AND SAVE,
5007	7006	RTL		
5010	3267	DCA	DS1	
5011	4312	JMS	LISN	/LOOK FOR 1
5012	7506	-272		
5013	5015	,+2		
5014	0000	0		
5015	1267	TAD	DS1	/COMPUTE CDF INSTRUCTION,
5016	1050	TAD	KCDF	
5017	3236	DCA	DLF	
5020	3267	DCA	DS1	/CLEAR TEMP FOR NEXT 4 INPUTS,
DLF2,	5021	JMS	LISN	/GET KEYBOARD,
5022	0001	1		
5023	5025	,+2		/OCTAL NUMBER,
5024	0000	0		
5025	1267	TAD	DS1	/ADD TEMP,
5026	2270	ISZ	DS2	/4TH NUMBER?
5027	7410	SKP		

```

5030 5235      JMP      ,+5
5031 7104      CLL RAL
5032 7006      RTL
5033 3267      DCA      DS1
5034 5221      JMP      DLF2
5035 3267      DCA      DS1
5036 6201      DLF,   CDF      /N
5037 1667      TAD I   DS1
5040 4457      DLF1,  JMS I   SCDFP
5041 4462      JMS I   EDITP
5042 3705      DCA I   M14P2P
5043 1463      TAD I   EDTEMP
5044 3704      DCA I   M14P1P
5045 4460      JMS I   AMGBEP
5046 7120      MSG14
5047 4312      JMS      LISN
5050 7563      -215
5051 5065      DCR
5052 7566      -212
5053 5055      ,+2
5054 0000      0
5055 1032      TAD      K215
5056 4467      JMS I   TYPEP
5057 2267      IS#    DS1
5060 5236      JMP      DLF
5061 1236      TAD      DLF
5062 1023      TAD      K10
5063 3236      DCA      DLF
5064 5236      JMP      DLF
5065 4465      DCR,   JMS I   CRLFP
5066 5711      JMP I   MONP
5067 0000      DS1,   0
5070 0000      DS2,   0
5071 4312      SL,    JMS      LISN
5072 0001      1
5073 5075      ,+2
5074 0000      0
5075 4703      JMS I   GET2NP
5076 7041      CIA
5077 1024      TAD      K14
5100 3157      DCA      SLKNST
5101 5702      JMP I   ,*1
5102 4625      CS
5103 4752      GET2NP, GET2N
5104 7121      M14P1P, M14P1
5105 7122      M14P2P, M14P2
5106 4460      QUES,   JMS I   AMGBEP
5107 7104      MSG11
5110 5711      JMP I   MONP
5111 4603      MONP,  MONIT+3
5112 0000      LISN,   0
5113 6031      KSF
5114 5313      JMP      ,*1
5115 6036      KRB
5116 3373      DCA      LISNT1

```

```

/NO, MOVE 3 PLACES LEFT,
/STORE,
/GO TO NEXT NUMBER,
/YES, 4TH NUMBER, SAVE ADDRESS,
/CHANGE TO REQUESTED DATA FIELD,
/GET CONTENTS,
/CHANGE BACK TO THIS PROGRAM FIELD,
/EDIT AND INSERT IN
/MESSAGE,
/PRINT MESSAGE,
/"D="
/"DLF"
/"DLF", DUMP NEXT LOCATION,
/"D=", LOOK FOR NEW COMMAND,
/SCOPE LOOP STRING,
/GET FIRST OCTAL DIGIT,
/GET 2ND OCTAL DIGIT AND
/NEGATE WHOLE NUMBER,
/BIAS BY 12 FOR BIT NUMER
/COMPARISON AND SAVE,
/THEN EXIT TO C STRING FOR
/CONTINUE CHECKS,
/TYPE "I=", RING BELL,
/RESTART MONITOR,
/GET KEYBOARD AND CHECK SYNTAX,
/SAVE INPUT

```

```

5117 1373      TAD      LISNT1
5120 1044      TAD      M203
5121 7640      SZA CLA
5122 7410      SKP
5123 5470      JMP I   CONTCP
5124 1373      TAD      LISNT1
5125 1045      TAD      M222
5126 7640      SZA CLA
5127 7410      SKP
5130 5471      JMP I   CONTRP
5131 1373      TAD      LISNT1
5132 4467      JMS I   TYPEP
5133 1373      TAD      LISNT1
5134 1374      TAD      M240
5135 7650      SNA CLA
5136 5013      JMP      LISN+1
5137 1712      LISN1, TAD I   LISN
5140 7450      SNA
5141 5306      JMP      QUES
5142 7500      SMA
5143 5354      JMP      LISNUM
5144 1373      TAD      LISNT1
5145 7640      SZA CLA
5146 5351      JMP      LISN2
5147 3373      DCA      LISNT1
5150 5365      JMP      LISN3
5151 2312      LISN2, IS#    LISN
5152 2312      IS#    LISN
5153 5337      JMP      LISN1
5154 7200      LISNUM, CLA
5155 1373      TAD      LISNT1
5156 1046      TAD      M260
5157 7710      SPA CLA
5160 5351      JMP      LISN2
5161 1373      TAD      LISNT1
5162 1047      TAD      M270
5163 7700      SMA CLA
5164 5351      JMP      LISN2
5165 2312      LISN3, IS#    LISN
5166 1712      TAD I   LISN
5167 3312      DCA      LISN
5170 1373      TAD      LISNT1
5171 0022      AND     K7
5172 5712      JMP I   LISN
5173 0000      LISNT1, 0
5174 7540      M240,  -240

```

```

/NO,
/YES, FAKE TEMP CLEAR AND
/EXIT,
/UPDATE CALL POINTERS,
/GET NEXT COMPARATOR,
/OCTAL NUMBER REQUIRED,
/GET COMPARATOR,
/IF 0, SYNTAX ERROR,
/IF 0, LOOK FOR OCTAL NUMBER,
/MATCH
/NO,
/YES, UPDATE CALL,
/GET RETURN ADDRESS,
/GET 0 OR OCTAL NUMBER CODE
/MASK TO LOW ORDER 3 BITS,
/EXIT,

```

```

/ERROR SERVICE ROUTINE,
/DECODES ERROR STATUS WORDS AND CARRIES OUT THE INDICATED
/ACTION,
/SAMPLE ERROR TABLE ENTRY!
/ERROR, STAT1 ERROR STATUS WORD,
/ STAT2 PRINT STATUS WORD,

/STATUS WORD BIT ASSIGNMENTS!
/STAT1 BIT STATE INDICATION
/0 0 NON-FATAL ERROR
/ 1 FATAL ERROR
/1 0 NO EXIT ON "EX=",
/ 1 EXIT ON "EX=",
/2-5: N TEST NUMBER =14 OCTAL(HEXADECIMAL)
/6-11: NN ANSCII FOR SUBTEST LETTER (6 BIT),

/STAT2 BIT PRINT OUT THE FOLLOWING WHEN SET,
/0 10T1 (11)
/1 10T2 (12)
/2 GOOD (GD)
/3 BAD (BD)
/4 OLD (OD)
/5 WORD COUNT (WC)
/6 CURRENT ADDRESS (CA)
/7 COMMAND REGISTER (CM)
/8 FUNCTION AND STATUS 1 (FS)
/9 MAIN STATUS (MS)
/10 DATA BUFFER (DB)
/11 CONTENTS OF AC FOR SKIP ERROR (AC)

/ERRORS MAY BE ENTERED IN ONE OF TWO WAYS;
/1. COMPAR GOES TO "ERRORS" IF "GOOD" NOT EQUAL TO "BAD",
/ ADDR ADDRESS OF ERROR STATUS WORDS,
/
/2. ERROR MAY BE USED BY FATAL ERRORS ONLY,
/ ADDR

5200 5200 PAGE
5201 0000 ERRORS, 0 /ENTER HERE WITH "ERROR"
5202 1371 CLA /ENTER HERE FROM "COMP" (PC PRESTORED)
5203 3466 TAD K4352
5204 1600 DCA I ERMSGP
5205 3347 TAD I ERRORS
5206 1747 DCA ERRPTR
5207 7710 TAD I ERRPTR
5210 5232 SPA CLA /GET ERROR TABLE ADDRESS AND
5211 4241 JMP FATERR /SAVE,
5212 4515 JMS OPRNT /GET STAT 1,
5213 4766 JMS I QHALTP /NON-FATAL ERROR?
5214 2200 ISZ ERRORS /NO, FATAL ERROR,
5215 4520 LOOP5 /NON-FATAL ERROR, CHECK PRINTOUT,
5216 5600 JMP I ERRORS
5217 1174 TAD EXITFL /CHECK RETURN TO MONITOR,
/ MOVE POINTER TO SUBTEST LOOP,
/ CHECK SUBTEST LOOP,
/ EXIT TO LOOP JUMP,
/ "EX=" PENDING?

```

```

5220 7700 SMA CLA
5221 5230 JMP ,*7
5222 1747 TAD I ERRPTR /NO,
5223 7004 RAL /YES, DOES THIS ERROR REQUIRE
/ "EX=" SERVICE?
5224 7700 SMA CLA
5225 5230 JMP ,*3
5226 2200 ISZ ERRORS /NO, LEAVE EXITFL SET,
5227 3174 DCA EXITFL /YES, MOVE POINTER TO RECYCLE JUMP,
5230 2200 ISZ ERRORS /CLEAN "EX=" FLAG,
5231 5600 JMP I ERRORS /MOVE POINTER ONE MORE POSITION,
/EXIT,

5232 4460 FATERR, JMS I AMG0EP /FATAL ERROR, PRINT MSG,
5233 7106 MSG12
5234 4464 JMS I BELLP
5235 1373 TAD K605
5236 4254 JMS PRNT /PUT "FE" IN AC,
/PRINT ERROR MESSAGE,
5237 3175 DCA ACTFLG /CLEAR TEST IN PROGRESS FLAG,
5240 5461 JMP I MONITP /GO TO MONITOR,
/PRINTOUT?
5242 7604 QPRNT, 0
5243 0033 LAS
5244 7640 AND K400
5245 5641 SEA CLA
5246 1372 JMP I OPRNT /NO, EXIT,
5247 4254 TAD K522 /YES, PUT "ER" IN AC AND
/GENERATE ERROR PRINTOUT,
5250 7240 JMS PRNT /BACK BIAS ERROR TABLE POINTER,
/TO POINT TO STAT1,
5251 1347 TAD ERRPTR
5252 3347 DCA ERRPTR
5253 5641 JMP I QPRNT /EXIT,

5254 2000 PRNT, 0 /GENERATE ERROR MESSAGE,
5255 3750 DCA I ERMP1P /SAVE "FE" OR "ER",
5256 1747 TAD I ERRPTR /GET TEST NUMBER FROM
/STAT1 AND PUT IN ERROR
5257 4404 BSM /MESSAGE,
5260 0025 AND K17
5261 1024 TAD K14
5262 4462 JMS I EDITP
5263 3751 DCA I ERMP2P
5264 1747 TAD I ERRPTR
5265 4404 BSM /GET SUBTEST LETTER FROM STAT1
/AND PUT IN ERROR MESSAGE
5266 0374 AND K7700 /AFTER APPENDING "SPACE" CODE,
5267 1026 TAD K40
5270 3752 DCA I ERMP3P
5271 1200 TAD ERRORS
5272 4462 JMS I EDITP /EDIT RETURN JUMP FOR "PC" PRINTOUT,
/
5273 3755 DCA I ERMP5P
5274 1463 TAD I EDTEMP
5275 3754 DCA I ERMP4P
5276 2347 ISZ ERRPTR
5277 1040 TAD M5
5280 3356 DCA PRCNT1 /SET COUNTER FOR 5 SYMBOLS
5281 1041 TAD M4 /BEFORE CALL,
5282 3357 DCA PRCNT2 /SET COUNTER FOR 12 SYMBOLS
5283 1364 TAD SYMPTP /MAXIMUM,
5284 3361 DCA SYMBOL /GET SYMBOL ROUTINE TABLE
/POINTER AND PUT IN TEMP,

```

5305	1360	TAD	ERMPTP	/GET ERROR MESSAGE INSERT
5306	3362	DCA	PRMPT	/POINTER,
5307	7130	STL RAR		
5310	3363	DCA	PRNTK	
5311	7604	LAS		/COMPLETE DUMP?
5312	0031	AND	K200	
5313	7650	SNA CLA		
5314	5317	JMP	,+3	
5315	7240	CLA CMA		/YES,FAKE STAT2 FULL;
5316	7410	SKP		
5317	1747	TAD I	ERRPTR	/GET STAT2 AND PUT IN TEMP,
5320	3365	DCA	WHAT	
5321	1365	PRNTLP, TAD	WHAT	/BIT BY BIT INSPECT STAT2 FOR
5322	7004	RAL		/SYMBOL PRINTOUTS, AS A
5323	3365	DCA	WHAT	/BIT IS SENSED SET, GO TO
5324	7420	SNL		/SYMBOL ROUTINE AND INSERT
5325	5337	JMP	PRNTCK	/SYMBOL AND DATA IN
5326	1761	TAD I	SYMBOL	
5327	3353	DCA	SYNADR	
5330	4753	JMS I	SYNADR	
5331	2356	IS4	PRCNT1	/D SYMBOLS PRINTED?
5332	5335	JMP	,+3	
5333	1367	TAD	K4300	/YES, INSERT 1 CR-LF IN SPACING,
5334	7410	SKP		
5335	7130	STL RAR		/NO, INSERT 1 SPACE
5336	3363	DCA	PRNTK	
5337	2361	PRNTCK, IS4	SYMBOL	/UPDATE SYMBOL TABLE POINTER,
5340	2357	IS4	PRCNT2	/11 SYMBOLS CHECKED?
5341	5321	JMP	PRNTLP	/NO, CHECK AGAIN,
5342	1367	TAD	K4300	/YES, PUT CR-LF AND END
5343	3762	DCA I	PRMPT	/MSG CODE IN ERROR MSG;
5344	4460	JMS I	AMG8EP	/PRINT ERROR MESSAGE,
5345	6665	CONTC7, ERMSG		
5346	5654	JMP I	PRNT	/EXIT
5347	0000	ERRPTH, 0		/TEMP FOR STAT2 WORD POINTER,
5350	6666	ERMP1P, ERMP1		/ERROR MESSAGE POINTERS; ERROR TYPE,
5351	6667	ERMP2P, ERMP2		/1 TEST NUMBER
5352	6670	ERMP3P, ERMP3		/1 SUBTEST LETTER=SPACE,
5353	0000	SYNADR, 0		
5354	6673	ERMP4P, ERMP4		/HIGH ORDER PC,
5355	6674	ERMP5P, ERMP5		/1 LOW ORDER PC,
5356	0000	PRCNT1, 0		/SYMBOL POINTER,
5357	0000	PRCNT2, 0		/5 SYMBOL COUNTER,
5360	6675	ERMPTP, ERMP5+1		/12 SYMBOL COUNTER,
5361	0000	SYMBOL, 0		/SYMBOL TABLE POINTER,
5362	0000	PRMPT, 0		/ERROR MESSAGE INSERT START POINTER,
5363	0000	PRNTK, 0		/SYMBOL TABLE POINTER TEMP,
5364	6405	SYMPTP, SYMPT		/ERROR MESSAGE INSERT POINTER,
5365	0000	WHAT, 0		/SPACING CONSTANT,
5366	5400	QHALT, QHALT		
5367	4300	K4300, 4300		/TEMP FOR STAT2 ROTATES,
5370	4343	K4343, 4343		/POINTER TO RETURN TO MONITOR CHECK.

5371	4352	K4352, 4352		
5372	5222	K522, 522		
5373	6605	K605, 605		
5374	7700	K7700, 7700		
5400	0000	PAGE		/RETURN TO MONITOR?
5401	7604	LAS		
5402	0034	AND	K1000	
5403	7640	S2A CLA		
5404	5600	JMP I	QHALT	/NO,
5405	6002	IOF		/YES, TURN INTERRUPT OFF,
5406	1213	TAD	HALTCP	/PUT CONTINUE EXIT IN 0
5407	3000	DCA	0	/SO "C" OR "EX" EXITS TO
5410	5461	JMP I	MONITP	/HALTC, GO TO MONITOR,
5411	7200	HALTC, CLA	QHALT	/CONTINUE,
5412	5600	JMP I	QHALT	/EXIT,
5413	5411	HALTCP, HALTC		/HALTC POINTER;
5414	0000	/SYMBOL ROUTINES,		
5415	1146	GD, 0		/GOOD,
5416	4326	TAD	GOOD	/GET GOOD,
5417	0407	JMS	SYMSEV	
5420	5614	0407		/"GD",
5421	0000	JMP I	GD	
5422	1147	0		
5423	4326	TAD	BAD	/GET BAD,
5424	0402	JMS	SYMSEV	
5425	5621	0402		/"BD",
5426	0000	JMP I	BD	
5427	1150	0		
5430	4326	TAD	OLD	/GET OLD
5431	0417	JMS	SYMSEV	
5432	5626	0417		/"OD",
5433	0000	JMP I	OD	
5434	4477	0		
5435	4326	RWCR		/GET WC,
5436	0327	JMS	SYMSEV	
5437	5633	0327		/"WC",
5440	0000	JMP I	WC	
5441	4500	0		
5442	4326	RCAR		/GET CA
5443	0103	JMS	SYMSEV	
5444	5640	0103		/"CA",
5445	0000	JMP I	CA	
5446	4502	0		
5447	4326	RCMR		/GET CH
5450	1503	JMS	SYMSEV	
5451	5645	1503		/"CH",
5452	0000	JMP I	CM	
5453	4503	0		
5454	4326	RFSR		/GET FS
5455	2306	JMS	SYMSEV	
5456	5652	2306		/"FS"
		JMP I	FS	

```

5457 0000 MS, 0
5460 4501 RMSR /GET MS
5461 4326 JMS SYMSEV
5462 2315 2315 /"MS"
5463 5657 JMP I MS
5464 0000 DB, 0
5465 4504 RDBR /GET DB
5466 4326 JMS SYMSEV
5467 0204 0204 /"DB"
5470 5664 JMP I DB

5471 0000 AC, 0
5472 1170 TAD ACLOC
5473 4326 JMS SYMSEV
5474 0301 0301
5475 5671 JMP I AC
5476 0000 11, 0
5477 1166 TAD 10T1
5500 4310 JMS 1112
5501 6111 6111
5502 5676 JMP I 11
5503 0000 12, 0
5504 1167 TAD 10T2
5505 4310 JMS 1112
5506 6211 6211
5507 5703 JMP I 12
5510 0000 1112, 0
5511 0365 AND K377 /ENTER FROM 11 OR 12,
5512 1034 TAD K1000 /CONVERT INST TO TAD POINTER
5513 3314 DCA ,+1
5514 0000 0
5515 7001 IAC /STORE TAD POINTER HERE,
5516 3314 DCA ,+2 /POINTER #1 WHICH IS ADDRESS
5517 1710 TAD I 1112 /OF 10T CODE,
5520 3323 DCA ,+3 /GET SYMBOL CODE,
5521 1714 TAD I ,+5 /PUT IN CALL TO SYMSEV,
5522 4326 JMS SYMSEV /GET 10T CODE,
5523 0000 0 /GO TO SYMBOL SERVICE,
5524 2310 1SE 1112 /SYMBOL CODE GOES HERE,
5525 5710 JMP I 1112 /UPDATE RETURN TO 11 OR 12,
5526 0000 SYMSEV, 0 /RETURN,
5527 3363 DCA SYMTEM /COMMON SYMBOL ROUTINE SERVICE,
5530 1761 TAD I ERHPP /SAVE DATA,
5531 3362 DCA ERMSYM /GET ERROR MESSAGA CURRENT POINTER,
5532 1726 TAD I SYMSEV /PUT ON THIS PAGE OF MEMORY,
5533 3377 AND (77
5534 1776 TAD I (PRNTR
5535 3762 DCA I ERMSYM
5536 2362 1SE ERMSYM
5537 1726 TAD I SYMSEV
5540 2375 AND (7700
5541 1364 TAD K72
5542 3762 DCA I ERMSYM
5543 2362 1SE ERMSYM
5544 1363 TAD SYMTEM /DATA,
                    /GET DATA,

```

```

5545 4462 JMS I EDITP /EDIT,
5546 3363 DCA SYMTEM /SAVE LOW ORDER,
5547 1463 TAD I EDTEMP /GET HIGH ORDER AND INSERT,
5550 3762 DCA I ERMSYM
5551 2362 1SE ERMSYM /UPDATE POINTER TO LOW ORDER
5552 1363 TAD SYMTEM /DATA, GET LOW ORDER AND INSERT,
5553 3762 DCA I ERMSYM
5554 2362 1SE ERMSYM /UPDA1 POINTER TO SPACING,
5555 1362 TAD ERMSYM /PUT CURRENT POINTER ON
5556 3761 DCA I ERHPP /"ENRRORS" MEMORY PAGE,
5557 2326 1SE SYMSEV /UPDA1C RETURN,
5560 5726 JMP I SYMSEV /EXIT,
5561 5362 ERHPP, PRMPT
5562 0000 ERMSYM, 0
5563 0000 SYMTEM, 0
5564 0072 K72, 72
5565 0377 K377, 377

```

/GENERAL INTERRUPT HANDLER,

/CHECKS FOR TIO AND TTI INTERRUPTS AND SERVICES SAME,

```

5575 7700
5576 5363
5577 0077
5600 PAGE
5600 3154 INTSEV, DCA SAVEAC /SAVE AC AND LINK,
5601 7010 RAR
5602 3155 DCA SAVEL
5603 6201 CDF 00 /GET FIELD 0 LOC 0
5604 1661 TAD I P0 /AND PUT IN THIS FIELD'S
5605 3000 DCA 0 /LOC 0 FOR INT RETURN,
5606 4457 JMS I SCDFF /CHANGE TO THIS DF,
5607 6041 INTAGN, TSF /TIO INTERRUPT?
5610 5215 JMP ,*5
5611 6042 TCF /YES, CLEAR HARDWARE FLAG
5612 7240 CLA CMA /AND SET SOFTWARE FLAG,
5613 3173 DCA TIOFLG
5614 5253 JMP INTOK /TRY TO EXIT,
5615 6031 KSF /NO, KEYBOARD INTERRUPT?
5616 5247 JMP INTCON
5617 6036 KRB
5620 1044 TAD M203
5621 7450 SNA
5622 5470 JMP I CONTCP /?C
5623 1243 TAD M17
5624 7450 SNA
5625 5471 JMP I CONTRP /?R
5626 1244 TAD M11
5627 7450 SNA
5630 5237 JMP ALTIN /ALT=233
5631 1245 TAD M142
5632 7450 SNA
5633 5237 JMP ALTIN /ALT=375
5634 1246 TAD M1
5635 7640 SEA CLA /ALT=376
5636 5253 JMP INTOK
5637 1165 ALTIN, TAD ALTENA /ALT ENABLED?
5640 7650 SNA CLA
5641 5253 JMP INTOK
5642 5461 JMP I MONITP /NO,
5643 7761 M17, =17 /YES,
5644 7767 M11, =11
5645 7636 M142, =142
5646 7777 M1, =1
5647 1000 INTCON, TAD 0
5650 3147 DCA BAD
5651 4526 ERROR /ILLEGAL INTERRUPT
5652 6431 FE31A /FATAL ERROR,
/BDI CONTENTS OF LOCATION 0; ALL OTHERS ARE CURRENT VALUES AND
/MAY NOT APPLY,

```

```

5653 7300 INTOK, CLA CLL /NO, RESTORE AC AND LINK
5654 1155 TAD SAVEL /AND EXIT,
5655 7024 RAL
5656 1154 TAD SAVEAC
5657 6201 IDN
5660 5400 JMP I 0
5661 2000 P0, 0

```

/SAVE AND RESTORE COMMON SUBROUTINE INFORMATION ROUTINES,
 /LOCATIONS TYPE, TYPTM, CRLF, AMGBE, AMGBE1, AMGBE2,
 /AMGBE3, EDIT, EDTEM, EDTEM1, EDTEM2, EDTEM3, SRMSR MUST BE SAVED AND RESTORED
 /SINCE MONITOR MAY INTERRUPT THESE ROUTINES
 /AND USE THEM ITSELF,

```

5662 0000 SAVEP, 0
5663 7200 CLA
5664 1324 TAD M20
5665 3325 DCA SAVEP1
5666 1352 TAD SRPTS
5667 3326 DCA SAVEP2
5670 1331 TAD SRHOLD
5671 3327 DCA SAVEP3
5672 2326 ISZ SAVEP2
5673 2327 ISZ SAVEP3
5674 1726 TAD I SAVEP2
5675 3330 DCA SAVEP4
5676 1730 TAD I SAVEP4
5677 3727 DCA I SAVEP3
5700 2325 ISZ SAVEP1
5701 5272 JMP ,=7
5702 5662 JMP I SAVEP
5703 0000 RESTP, 0
5704 7200 CLA
5705 1324 TAD M20
5706 3325 DCA SAVEP1
5707 1331 TAD SRHOLD
5710 3326 DCA SAVEP2
5711 1352 TAD SRPTS
5712 3327 DCA SAVEP3
5713 2326 ISZ SAVEP2
5714 2327 ISZ SAVEP3
5715 1727 TAD I SAVEP3
5716 3330 DCA SAVEP4
5717 1726 TAD I SAVEP2
5720 3730 DCA I SAVEP4
5721 2325 ISZ SAVEP1
5722 5313 JMP ,=7
5723 5703 JMP I RESTP
5724 7760 M20, -20
5725 0000 SAVEP1, 0
5726 0000 SAVEP2, 0
5727 0000 SAVEP3, 0
5730 0000 SAVEP4, 0
5731 5731 SRHOLD, 0
5732 0000 0
5733 0000 0
5734 0000 0
5735 0000 0
5736 0000 0
5737 0000 0
5740 0000 0
5741 0000 0
5742 0000 0
  
```

```

5743 0000 0
5744 0000 0
5745 0000 0
5746 0000 0
5747 0000 0
5750 0000 0
5751 0000 0
5752 5752 SRPTS, 1
5753 6167 CRLF
5754 2553 TYPE
5755 2570 TYPTM
5756 2737 AMGBE
5757 2776 AMGBE1
5760 2753 AMGBE2
5761 2777 AMGBE3
5762 6105 EDIT
5763 6132 EDTEM
5764 6133 EDTEM1
5765 6134 EDTEM2
5766 6135 EDTEM3
5767 4052 SRMSR
5770 3941 SRBSW
5771 3553 SRBSWT
5772 4026 SLCMR
  
```

/EXECUTIVE ADDENDUM:

6000	6000	PAGE		
6001	0030	EXECFX, 0		
6002	4460	JMS I	AMG8EP	/WHAT DRIVE TO USE?
6003	7026	MSG3		/PUT DRIVE # IN DRIVE,
6004	4257	JMS	KBOCT	
6005	5201	JMP	,+3	
6006	3304	DCA	DRIVE	
6007	4460	EFX1, JMS I	AMG8EP	/7 OR 9 TRACK?
6008	7033	MSG4		/SET TRK9 IF 9 TRACK,
6009	4246	JMS	KB	/CLEAN IT IF 7,
6010	1277	TAD	M267	
6011	1277	SZA CLA		
6012	7640	JMP	,+3	
6013	5216	DCA	TRK9	
6014	3161	JMS	EFX2	
6015	5224	JMP	TXXTM1	
6016	1151	TAD	M271	
6017	1300	TAD	M271	
6020	7640	SZA CLA		
6021	5206	JMP	EFX1	
6022	7040	CMA		
6023	3161	DCA	TRK9	
6024	1026	EFX2, TAD	K40	/BITS 0-2, SET BIT 12=1, 11=0
6025	1304	TAD	DRIVE	/AND PROGRAM
6026	7112	CLL RTR		/FIELD IN BITS 6-8,
6027	7012	RTR		
6030	1156	TAD	PRGFLD	
6031	3163	DCA	KCMD	
6032	1304	TAD	DRIVE	/PUT DRIVE AND TRACK INFO
6033	4462	JMS I	EDITP	/IN INSTRUCTION MESSAGE,
6034	1303	TAD	M2000	
6035	3674	DCA I	M6P1F	
6036	1161	TAD	TRK9	
6037	7640	SZA CLA		
6040	1031	TAD	K200	
6041	1276	TAD	K6740	
6042	3675	DCA I	M6P2P	
6043	4460	JMS I	AMG8EP	/PRINT INSTRUCTION MESSAGE,
6044	7043	MSG6		
6045	5600	JMP I	EXECFX	
6046	0000	KB,	0	/KEYBOARD IN,
6047	6031	KSF		
6050	5247	JMP	,=1	
6051	6036	KRB		
6052	3151	DCA	TXXTM1	
6053	1151	TAD	TXXTM1	
6054	4467	JMS I	TYPFP	
6055	1151	TAD	TXXTM1	
6056	5646	JMP I	KB	
6057	0000	KBOCT,	0	/LOOK FOR OCTAL NUMBER
6060	4246	JMS	KB	/INPUT,
6061	1046	TAD	M260	
6062	7710	SPA CLA		

6063	5657	JMP I	KBOCT	
6064	1151	TAD	TXXTM1	
6065	1047	TAD	M270	
6066	7700	SMA CLA		
6067	5657	JMP I	KBOCT	
6070	1151	TAD	TXXTM1	
6071	3022	AND	K7	
6072	2257	IS#	KROCT	
6073	5657	JMP I	KBOCT	
6074	7050	M6P1P,	M6P1	
6075	7052	M6P2P,	M6P2	
6076	6740	K6740,	6740	
6077	7511	M267,	-267	
6100	7507	M271,	-271	
6101	7462	M316,	-316	
6102	7447	M331,	-331	
6103	6000	M2000,	-2000	
6104	0000	DRIVE,	0	/SELECTED DRIVE NUMBER,

```

/EDIT ROUTINE.
/CONVERTS A 4 DIGIT OCTAL NUMBER IN AC TO 2 PACKED ANSCII
/WORDS, EXITS WITH LEAST SIGNIFICANT PORTION IN AC
/AND MOST SIGNIFICANT IN "EDTEM".
6105 0000 EDIT, 0
6106 3333 DCA EDTEM1
6107 1333 TAD EDTEM1
6110 4404 BSW
6111 4316 JMS EDIT1
6112 3332 DCA EDTEM
6113 1333 TAD EDTEM1
6114 4316 JMS EDIT1
6115 5705 JMP I EDIT
6116 0000 EDIT1, 0
6117 3334 DCA EDTEM2
6120 1334 TAD EDTEM2
6121 0022 AND K7
6122 3335 DCA EDTEM3
6123 1334 TAD EDTEM2
6124 7006 RTL
6125 7004 RAL
6126 0337 AND K700
6127 1335 TAD EDTEM3
6130 1336 TAD K6060
6131 5716 JMP I EDIT1
6132 0000 EDTEM, 0
6133 0000 EDTEM1, 0
6134 0020 EDTEM2, 0
6135 0000 EDTEM3, 0
6136 6060 K6060, 6060
6137 0700 K700, 700

6140 7200 CONTC, CLA
6141 3175 DCA ACTFLG
6142 1000 TAD 0
6143 3146 DCA GOOD
6144 1160 TAD SLADDR
6145 3147 DCA BAD
6146 1363 TAD CONTC1
6147 3764 DCA I CONTC2
6150 1365 TAD CONTC3
6151 3925 DCA I CONTC4
6152 5766 JMP I CONTC4
6153 6457 CONTC5, TR31D
6154 7000 NOP
6155 1361 TAD ,*4
6156 3764 DCA I CONTC2
6157 4515 CLEAR1
6160 4460 JMS I ANGBEP
6161 6665 ERMSG
6162 5461 JMP I MONITP
6163 7124 CONTC1, MSG15
6164 5345 CONTC2, CONTC7

/*C
/CLEAN ACTIVE FLAG
/GET INTERRUPT LOC AND
/SAVE IN GOOD,
/SAVE LAST SETN ADDRESS,
/PUT *C IN ERROR PRINTER,
/PUT RETURN ADDRESS IN COMPARE,
/GO TO COMPARE TO SAVE UP INFORMATION
/GOOD AND BAD SAVED AS ABOVE,
/PUT ERMSG BACK IN ERRORS,
/CLEAR ALL
/PRINT ALL ERROR MESSAGE,
/GO TO MONITOR,

```

```

6165 6153 CONTC3, CONTC5
6166 0914 CONTC4, CONTC8

/CARRIAGE RETURN = LINE FEED.

6167 0000 CRLF, 0
6170 7200 CLA
6171 1032 TAD K215
6172 4467 JMS I TYPEP
6173 1377 TAD K212
6174 4467 JMS I TYPEP
6175 4467 JMS I TYPEP
6176 5767 JMP I CRLF
6177 0212 K212, 212

```

/SET UP ROUTINES; CALLING SEQUENCES FOLLOW,
 /SET1
 /CONSTANT TO BE STORED IN GOOD,
 /CONSTANT TO BE LOADED INTO WC,
 /CONSTANT TO BE LOADED INTO CA,
 /DATA TO BE STORED IN CA+1 AND CA+2
 /CONSTANT TO BE INCLUSIVE DR'ED WITH DRIVE, PROGRAM FIELD, MASTER BIT AND N TRACK
 /DENSITY CONSTANT AND LOADED INTO CM,
 /CONSTANT TO BE LOADED IN FR.
 /SET 2 SAME AS SET 1 EXCEPT THAT THE CM CONSTANT IS NOT
 /INCLUSIVLY DR'ED WITH THE DENSITY CONSTANT,
 /SET 3 SAME AS SET 1 EXCEPT CA CONSTANT AND MEM
 /DATA ARE NOT INCLUDED IN CALL,
 /SET 4 SAME AS SET 3 EXCEPT CM CONSTANT IS NOT
 /INCLUSIVE DR'ED WITH THE DENSITY CONSTANT,

6200	0000	SET1R, 0		/SET 1,
6201	7240	CLA CMA		/BIAS RETURN FOR AUTO INDEX,
6202	1200	TAD	SET1R	
6203	4232	JMS	SETA	
6204	4254	JMS	SETB	
6205	4270	JMS	SETC	
6206	4277	JMS	SETD	
6207	0000	SET2R, 0		/SET 2,
6210	7240	CLA CMA		/BIAS RETURN FOR AUTO INDEX,
6211	1207	TAD	SET2R	
6212	4232	JMS	SETA	
6213	4254	JMS	SETB	
6214	4303	JMS	SETC	
6215	4277	JMS	SETD	
6216	0000	SET3R, 0		/SET 3,
6217	7240	CLA CMA		/BIAS RETURN FOR AUTO INDEX,
6220	1216	TAD	SET3R	
6221	4232	JMS	SETA	
6222	4270	JMS	SETC	
6223	4277	JMS	SETD	
6224	0000	SET4R, 0		/SET 4,
6225	7240	CLA CMA		/BIAS RETURN FOR AUTO INDEX,
6226	1224	TAD	SET4R	
6227	4232	JMS	SETA	
6230	4303	JMS	SETC	
6231	4277	JMS	SETD	
6232	0000	SETA, 0		/SETA GENERAL SERVICE,
6233	3160	DCA	SLADDR	/STORE SCOPE LOOP STARTER,
6234	1160	TAD	SLADDR	
6235	3017	DCA	AUTO17	/SAVE POINTER IN AUTO INDEX,
6236	1417	TAD I	AUTO17	/GET AND STORE GOOD CONSTANT,
6237	3330	DCA	SETT	

6240	4352	JMS	INTERR	/CHECK FOR INTERMEDIATE ERRORS,
6241	4506	SKCB		/WAIT FOR CONTROL NOT BUSY,
6242	5241	JMP	.*1	
6243	1163	TAD	KCMD	
6244	4474	LCMR		
6245	4313	JMS	SETTST	
6246	4510	SKTR		/WAIT FOR TRANSPORT READY,
6247	5246	JMP	.*1	
6250	4515	CLEAR1		/GENERATE CLT, ETC;
6251	1417	TAD I	AUTO17	/GET WC CONSTANT AND LOAD IT,
6252	4472	LWCR		
6253	5632	JMP I	SETA	/EXIT
6254	0000	0		/SET B GENERAL SERVICE;
6255	1417	TAD I	AUTO17	/GET CA CONSTANT AND LOAD IT,
6256	4473	LCAR		
6257	4500	RCAR		
6260	7001	IAC		/COMPUTE MEM DATA ADDRESS
6261	3232	DCA	SETA	(/1 MORE THAN CA),
6262	1417	TAD I	AUTO17	/GET MEM DATA AND STORE IT
6263	3632	DCA I	SETA	/IN CA+1,
6264	1632	TAD I	SETA	/AND CA+2
6265	2232	ISE	SETA	
6266	3632	DCA I	SETA	
6267	5654	JMP I	SETB	/EXIT,
6270	0000	0		/SET C GENERAL SERVICE;
6271	1417	TAD I	AUTO17	/GET CM CONSTANT AND INCLUSIVE
6272	4405	HQL		/OR WITH DRIVE, PROGRAM FIELD, MASTER BIT
6273	1163	TAD	KCMD	/AND 202 BPI FOR 7 TRK, 800 BPI
6274	4426	HQA		(/NOT CORE DUMP) FOR 9 TRK,
6275	4474	LCMR		/THEN LOAD IT,
6276	5670	JMP I	SETC	/EXIT,
6277	0000	0		/SETD GENERAL SERVICE,
6300	1417	TAD I	AUTO17	/GET PR CONSTANT,
6301	4475	LFGR		/THEN LOAD IT,
6302	5417	JMP I	AUTO17	/EXIT TO MAIN PROGRAM,
6303	0000	0		/SETE GENERAL SERVICE,
6304	1417	TAD I	AUTO17	/GET CM CONSTANT AND
6305	4405	HQL		/INCLUSIVE OR WITH DRIVE AND
6306	1163	TAD	KCMD	/PROGRAM FIELD, (DENSITY BITS
6307	0037	AND	M4	/ARE FEMOLD IN KCMD,)
6310	4406	HQA		
6311	4474	LCMR		/THEN LOAD IT,
6312	5703	JMP I	SETE	/EXIT,
6313	0000	SETTST, 0		/VERIFY SELECT REMOTE BIT IS CLEAR;
6314	7200	CLA	GOOD	
6315	3146	DCA	GOOD	
6316	4501	RMSR		
6317	0033	AND	K400	
6320	7450	SNA		
6321	5324	JMP	.*3	/INHIBIT TRACE PRINT IF OK
6322	4525	COMPAR		
6323	6455	EH31C		
6324	7000	NOP		/THERE IS NO SUBTEST LOOP,
6325	1330	TAD	SETT	

```

6326 3146          DCA    GOOD
6327 5713          JMP I  SETTST
6330 0000          SETT,  0          /TEMPORARY FOR GOOD,

/CONTINUE ROUTINE, USED FOR CONTINUE AND CHANGE DIRECTION MODES,
/CALLING SEQUENCE
/ SET1 (OR SET2) THIS CALLS SET1(2) FOR FIRST FUNCTION,
/ N /GD
/ N /WC
/ N /CA
/ N /MEM
/ COMMAND/CM
/ FUNCTION/FR THIS IS FIRST FUNCTION,
/ INST ANY LEGAL INSTRUCTION IS REQUIRED HERE
/ CONTNU THIS DOES NEXT FUNCTION IN CONTINUE OR CHANGE
/ FUNCTION/FR DIRECTION MODE USING PREVIOUS SET DATA,

6331 0000          CONTNR, 0          /ENTER WITH "CONTNU";
6332 1391          TAD    N10          /BACK BIAS POINTER TO GD CALL,
6333 1331          TAD    CONTNR
6334 3017          DCA    AUTO17          /PUT IN AUTO INDEX;
6335 4352          JMS    INTERR          /CHECK FOR INTERMEDIATE ERRORS,
6336 4506          SKCB          /WAIT FOR CONTROL READY,
6337 5336          JMP    ,=1
6340 4511          CLF          /CLEAN FLAGS;
6341 1417          TAD I  AUTO17          /GET WC AND LOAD IT,
6342 4472          LWDR
6343 4254          JMS    SETB          /LOAD CA AND MEM,
6344 4270          JMS    SETC          /LOAD CM,
6345 2017          ISZ    AUTO17          /UPDATE POINTER TO NEW FUNCTION,
6346 2017          ISZ    AUTO17
6347 2017          ISZ    AUTO17
6350 4277          JMS    SETD          /GET IT, LOAD IT, AND EXIT,
6351 7770          N10,   -10

6352 0000          INTERR, 0          /CHECK FOR INTERMEDIATE ERRORS,
6353 4505          SKEF          /WAIT FOR ERROR
6354 7410          SKP
6355 5360          JMP    ,=3
6356 4506          SKCB          /OR CONTROL READY,
6357 5353          JMP    ,=4
6360 1171          TAD    EXPEOF          /EXPECTED EOF IN LINK
6361 7004          RAL
6362 7240          CLA CMA
6363 3171          DCA    EXPEOF          /CLEAN EXPECTED EOF
6364 4505          SKEF          /ERROR?
6365 5752          JMP I  INTERR          /NO, OK
6366 7620          SNL CLA          /YES, EOF EXPECTED?
6367 5752          JMP I  INTERR          /YES OK,
6370 1160          TAD    SLADDR          /NO, PC TO BAD
6371 3147          DCA    BAD

```

```

6372 4526          ERROR          /MCPONT EMRR ER31E
6373 6461          ER31E
/BD: INDICATES THE ADDRESS OF THE 2ND SET COMMAND OR THE CONTNU COMMAND
/WHICH FOLLOWED THE ERROR,
/AN ERROR HERE INDICATES THAT A SUBTEST WHICH ISSUED 2 FUNCTIONS TO
/THE CONTROL INDICATED AN ERROR AFTER THE FIRST FUNCTION WAS INITIATED
/BU! PRIOR TO THE 2ND FUNCTION,
6374 7000          NOP
6375 4515          CLEAR1
6376 5752          JMP I  INTERR /RECOMMENDED RECOVERY IS TO PROCEED WITH SR 5
/SET TO A 1 SO THAT THE SUBTEST WILL BE REINITIATED,

```

6377	4460	CONTR,	JMS I	AMGBEP	/R
6400	7126		MSG16		
6401	3175		DCA	ACTFLG	
6402	4515		CLEAR1		
6403	4527		LOADPT		
6404	5461		JMP I	MONITP	
6405	5476	SYMPT,	I1		/SYMBOL ROUTINE TABLE FOR ERRORS,
6406	5503		I2		
6407	5414		GD		
6410	5421		BD		
6411	5426		OD		
6412	5433		WC		
6413	5440		CA		
6414	5445		CM		
6415	5452		FS		
6416	5457		MS		
6417	5464		DB		
6420	5471		AC		
6421	0203	TST,	TEST15		/EXEC TEST POINTERS,
6422	0600		TEST16		
6423	1200		TEST17		
6424	1427		TEST20		
6425	1600		TEST21		
6426	2000		TEST22		
6427	2400		TEST23		
6430	2600		TEST24		
6431	3000		TEST25		
6432	3200		TEST26		
6433	3400		TEST27		
6434	3600		TEST30		
6435	0000	TAL,	0		/TEST TALLIES FOR T15
6436	7766		-12		/T16
6437	7766		-12		/T17
6440	7766		-12		/T20
6441	7766		-12		/T21
6442	7766		-12		/T22
6443	7766		-12		/T23
6444	7766		-12		/T24
6445	7766		-12		/T25
6446	7766		-12		/T26
6447	7766		-12		/T27
6450	7777		-1		/T30

/ERROR TABLE FOR ERRORS OCCURRING OUTSIDE OF FORMAL TESTS, (CODE 31X)

6451	7501	FE31A,	7501
6452	7776		7776
6453	1502	ER31B,	1502
6454	7601		7601
6455	1503	ER31C,	1503
6456	7776		7776
6457	1504	TR31D,	1504
6460	1576		1576
6461	1505	ER31E,	1505
6462	7777		7777

/ERROR TABLE FOR ERRORS OCCURRING IN FORMAL TESTS, (CODE 15X-30X)

6463	0101	ER15A,	0101
6464	0034		0034
6465	0102	ER15B,	0102
6466	1434		1434
6467	0103	ER15C,	0103
6470	1434		1434
6471	0104	ER15D,	0104
6472	4034		4034
6473	0105	ER15E,	0105
6474	1434		1434
6475	0106	ER15F,	0106
6476	1434		1434
6477	0107	ER15G,	0107
6500	4034		4034
6501	0110	ER15H,	0110
6502	1074		1074
6503	0111	ER15I,	0111
6504	1434		1434
6505	0112	ER15J,	0112
6506	1434		1434
6507	0201	ER16A,	0201
6510	1576		1576
6511	0202	ER16B,	0202
6512	1576		1576
6513	0203	ER16C,	0203
6514	1576		1576
6515	0204	ER16D,	0204
6516	1576		1576
6517	0205	ER16E,	0205
6520	1576		1576
6521	0206	ER16F,	0206
6522	1576		1576
6523	0207	ER16G,	0207
6524	1576		1576
6525	0210	ER16H,	0210
6526	1576		1576
6527	0211	ER16I,	0211
6530	1576		1576
6531	0212	ER16J,	0212

6532	1576		1576
6533	2321	ER17A,	2301
6534	1776		1776
6535	2322	ER17B,	2302
6536	1776		1776
6537	2303	ER17C,	2303
6540	1776		1776
6541	2304	ER17D,	2304
6542	1776		1776
6543	J401	ER20A,	0401
6544	1576		1576
6545	0402	ER20B,	0402
6546	1576		1576
6547	0403	ER20C,	0403
6550	1576		1576
6551	0501	ER21A,	0501
6552	1576		1576
6553	0502	ER21B,	0502
6554	1576		1576
6555	0503	ER21C,	0503
6556	1576		1576
6557	0601	ER22A,	0601
6560	1434		1434
6561	0602	ER22B,	0602
6562	5434		5434
6563	0603	ER22C,	0603
6564	1434		1434
6565	0604	ER22D,	0604
6566	1434		1434
6567	0605	ER22E,	0605
6570	1434		1434
6571	0606	ER22F,	0606
6572	1434		1434
6573	0607	ER22G,	0607
6574	1534		1534
6575	0610	ER22H,	0610
6576	1534		1534
6577	0611	ER22I,	0611
6600	1434		1434
6601	0612	ER22J,	0612
6602	1434		1434
6603	0613	ER22K,	0613
6604	1434		1434
6605	0701	ER23A,	0701
6606	1434		1434
6607	0702	ER23B,	0702
6610	1434		1434
6611	0703	ER23C,	0703
6612	1434		1434
6613	0704	ER23D,	0704
6614	1534		1534
6615	0705	ER23E,	0705
6616	1534		1534
6617	0706	ER23F,	0706
6620	1434		1434

6621	1001	ER24A,	1001
6622	1434		1434
6623	1002	ER24B,	1002
6624	1434		1434
6625	1003	ER24C,	1003
6626	1434		1434
6627	1101	ER25A,	1101
6630	1776		1776
6631	1102	ER25B,	1102
6632	1776		1776
6633	3103	ER25C,	3103
6634	1601		1601
6635	3201	ER26A,	3201
6636	1600		1600
6637	1301	ER27A,	1301
6640	1576		1576
6641	1302	ER27B,	1302
6642	1576		1576
6643	1303	ER27C,	1303
6644	1576		1576
6645	1304	ER27D,	1304
6646	1576		1576
6647	1401	ER30A,	1401
6650	1576		1576
6651	1402	ER30B,	1402
6652	1576		1576
6653	1403	ER30C,	1403
6654	1576		1576
6655	1404	ER30D,	1404
6656	1576		1576
6657	1405	ER30E,	1405
6660	1576		1576
6661	1406	ER30F,	1406
6662	1576		1576
6663	1407	ER30G,	1407
6664	1576		1576

/ERROR MESSAGE!

```

/THE FULL ERROR MESSAGE CALLED BY STA12*7777 WOULD BE
/ERMSG, TEXT **ERNNX PCINNNN I11 NNNN I21 NNNN GDI NNNN BDI NNNN ODI NNNN
/ WC: NNNN CAI NNNN CMI NNNN FSI NNNN MSI NNNN DBI NNNN ACI NNNN**
6665 0000 ERMSG, 0 /SPACING AND SYMBOL FILL IN,
6666 0000 ERMP1, 0 /ERROR TYPE
6667 0000 ERMP2, 0 /TEST TYPE
6670 0000 ERMP3, 0 /SUBTEST LETTER, SPACE
6671 4020 /SP,P
6672 0372 /C,1
6673 0000 ERMP4, 0 /CONTENTS OF PC.
6674 0000 ERMP5, 0 /DITTO
        DECIMAL /ALL OTHER INFORMATION INSERTED HERE,
6675 0000 ZBLOCK 65
6676 0000 ZBLOCK 65
6677 0000 ZBLOCK 65
6678 0000 ZBLOCK 65
6679 0000 ZBLOCK 65
6680 0000 ZBLOCK 65
6681 0000 ZBLOCK 65
6682 0000 ZBLOCK 65
6683 0000 ZBLOCK 65
6684 0000 ZBLOCK 65
6685 0000 ZBLOCK 65
6686 0000 ZBLOCK 65
6687 0000 ZBLOCK 65
6688 0000 ZBLOCK 65
6689 0000 ZBLOCK 65
6690 0000 ZBLOCK 65
6691 0000 ZBLOCK 65
6692 0000 ZBLOCK 65
6693 0000 ZBLOCK 65
6694 0000 ZBLOCK 65
6695 0000 ZBLOCK 65
6696 0000 ZBLOCK 65
6697 0000 ZBLOCK 65
6698 0000 ZBLOCK 65
6699 0000 ZBLOCK 65
6700 0000 ZBLOCK 65
6701 0000 ZBLOCK 65
6702 0000 ZBLOCK 65
6703 0000 ZBLOCK 65
6704 0000 ZBLOCK 65
6705 0000 ZBLOCK 65
6706 0000 ZBLOCK 65
6707 0000 ZBLOCK 65
6708 0000 ZBLOCK 65
6709 0000 ZBLOCK 65
6710 0000 ZBLOCK 65
6711 0000 ZBLOCK 65
6712 0000 ZBLOCK 65
6713 0000 ZBLOCK 65
6714 0000 ZBLOCK 65
6715 0000 ZBLOCK 65
6716 0000 ZBLOCK 65
6717 0000 ZBLOCK 65
6720 0000 ZBLOCK 65
6721 0000 ZBLOCK 65
6722 0000 ZBLOCK 65
6723 0000 ZBLOCK 65
6724 0000 ZBLOCK 65
6725 0000 ZBLOCK 65
6726 0000 ZBLOCK 65
6727 0000 ZBLOCK 65
6730 0000 ZBLOCK 65
6731 0000 ZBLOCK 65
6732 0000 ZBLOCK 65
6733 0000 ZBLOCK 65
6734 0000 ZBLOCK 65
6735 0000 ZBLOCK 65
6736 0000 ZBLOCK 65
6737 0000 ZBLOCK 65
6740 0000 ZBLOCK 65
6741 0000 ZBLOCK 65
6742 0000 ZBLOCK 65
6743 0000 ZBLOCK 65
6744 0000 ZBLOCK 65
    
```

```

6745 0000 ZBLOCK 65
6746 0000 ZBLOCK 65
6747 0000 ZBLOCK 65
6750 0000 ZBLOCK 65
6751 0000 ZBLOCK 65
6752 0000 ZBLOCK 65
6753 0000 ZBLOCK 65
6754 0000 ZBLOCK 65
6755 0000 ZBLOCK 65
6756 0000 ZBLOCK 65
6757 0000 ZBLOCK 65
6760 0000 ZBLOCK 65
6761 0000 ZBLOCK 65
6762 0000 ZBLOCK 65
6763 0000 ZBLOCK 65
6764 0000 ZBLOCK 65
6765 0000 ZBLOCK 65
6766 0000 ZBLOCK 65
6767 0000 ZBLOCK 65
6770 0000 ZBLOCK 65
6771 0000 ZBLOCK 65
6772 0000 ZBLOCK 65
6773 0000 ZBLOCK 65
6774 0000 ZBLOCK 65
6775 0000 ZBLOCK 65
    
```

/(DONE BY ASSEMBLER,)

DCIAL

/MESSAGE TEXTS:

```

6776 4324 MSG1, TEXT "#TM8E CONTROL TEST PART 2##MAINDEC-08-DHTMB-A#"
6777 1570
7000 0540
7001 0317
7002 1624
7003 2217
7004 1440
7005 2405
7006 2324
7007 4020
7010 0122
7011 2440
7012 6243
7013 4315
7014 0111
7015 1604
7016 0503
7017 5560
7020 7055
7021 0410
7022 2415
7023 0255
7024 0143
7025 0000

7026 4304 MSG3, TEXT "#DRIVE?"
7027 2211
7030 2605
7031 7740
7032 0000

7033 4367 MSG4, TEXT "#7 OR 9 TRACK?"
7034 4017
7035 2240
7036 7140
7037 2422
7040 0103
7041 1377
7042 4000

7043 4323 MSG6, TEXT "#SETUP DRV N (N TRK)?"
7044 0524
7045 2520
7046 4004
7047 2226
7050 4016
7051 4050
7052 1640
7053 2422
    
```

```

7054 1351
7055 4300
      7043 MSG61=MSG6
      7050 M6P1=MSG6+5
      7052 M6P2=MSG6+7
      /MSG7, TEXT "#TEST N#?"
      MSG7, 4324
7056 4324 MSG7, 4324
7057 0523      0523
7060 2440      2440
7061 0000 M7P1, 0
7062 4300      4300
7063 4323 MSG8, TEXT "#SELECTED TESTS DONE#"
7064 0514
7065 0503
7066 2405
7067 0440
7070 2405
7071 2324
7072 2340
7073 0417
7074 1605
7075 4300

      /MSG9, TEXT "#PASS NNNN"
      MSG9, 4320
7077 0123      0123
7100 2340      2340
7101 0000 M9P1, 0
7102 0000 M9P2, 0
7103 0000      0000
7104 7743 MSG11, TEXT "?#"
7105 0000

7106 4343 MSG12, TEXT "##FATAL ERROR#"
7107 0601
7110 2401
7111 1440
7112 0522
7113 2217
7114 2252
7115 0000

7116 4333 MSG13, TEXT "#["
7117 0000

      /MSG14, TEXT " NNNN"
      MSG14, 4040
7120 0000 M14P1, 0
7122 0000 M14P2, 0
7123 0000      0000

7124 3603 MSG15, TEXT "#C"
7125 0000

7126 3622 MSG16, TEXT "#R"
    
```

7127	0000		
7130	4316	MSG17, TEXT	"NOT AT BOT:0"
7131	1724		
7132	4001		
7133	2440		
7134	0217		
7135	2441		
7136	4300		
7137	4324	MTM2, TEXT	"#T30A1"
7140	6360		
7141	0172		
7142	0000		
7143	2611	MTM3, TEXT	"VISUALLY CHECK DRIVE OFF LINE AT BOT,"
7144	2325		
7145	0114		
7146	1431		
7147	4003		
7150	1005		
7151	0313		
7152	4004		
7153	2211		
7154	2605		
7155	4017		
7156	0606		
7157	4014		
7160	1116		
7161	0540		
7162	0124		
7163	4002		
7164	1724		
7165	5600		
7166	4324	MTM4, TEXT	"#T30B: PUT ON LINE, NO VACUUM,"
7167	6360		
7170	0272		
7171	4020		
7172	2524		
7173	4017		
7174	1640		
7175	1411		
7176	1605		
7177	5440		
7200	1617		
7201	4026		
7202	0103		
7203	2525		
7204	1556		
7205	0000		
7206	4324	MTM5, TEXT	"#T30C-EI REMOVE WRITE PERMISS, PUT ON LINE WITH VACUUM,"
7207	6360		
7210	0355		
7211	0572		

7212	4022		
7213	0515		
7214	1726		
7215	0540		
7216	2722		
7217	1124		
7220	0540		
7221	2005		
7222	2215		
7223	1123		
7224	2354		
7225	4020		
7226	2524		
7227	4017		
7230	1640		
7231	1411		
7232	1605		
7233	4027		
7234	1124		
7235	1040		
7236	2601		
7237	0325		
7240	2515		
7241	5600		
7242	4324	MTM6, TEXT	"#T30F: AS SOON AS DRIVE STARTS TO MOVE, MANUALLY PUT IT OFF LINE,"
7243	6360		
7244	0672		
7245	4001		
7246	2340		
7247	2317		
7250	1716		
7251	4001		
7252	2340		
7253	0422		
7254	1126		
7255	0540		
7256	2324		
7257	0122		
7260	2423		
7261	4024		
7262	1740		
7263	1517		
7264	2605		
7265	5440		
7266	1501		
7267	1625		
7270	0114		
7271	1431		
7272	4020		
7273	2524		
7274	4011		
7275	2440		
7276	1706		
7277	0640		

```

7300 1411
7301 1605
7302 5600

7303 4324   MTM7,   TEXT   "#T3PG: WHEN PDP HALTS, DEPRESS START OR CLEAR/CONTINUE,"
7304 6360
7305 3772
7306 4027
7307 1005
7310 1640
7311 2004
7312 2040
7313 1001
7314 1424
7315 2354
7316 4004
7317 0520
7320 2205
7321 2323
7322 4023
7323 2401
7324 2224
7325 4017
7326 2240
7327 0314
7328 0501
7331 2257
7332 0317
7333 1624
7334 1116
7335 2505
7336 5600

7337 0000   0           /TEST 25 BUFFER AREA,
7340 0000   0
7341 0000   0
              DECIMAL
7342 0000   T25BUF, ZBLOCK 41
7343 0000   T25BUF, ZBLOCK 41
7344 0000   T25BUF, ZBLOCK 41
7345 0000   T25BUF, ZBLOCK 41
7346 0000   T25BUF, ZBLOCK 41
7347 0000   T25BUF, ZBLOCK 41
7350 0000   T25BUF, ZBLOCK 41
7351 0000   T25BUF, ZBLOCK 41
7352 0000   T25BUF, ZBLOCK 41
7353 0000   T25BUF, ZBLOCK 41
7354 0000   T25BUF, ZBLOCK 41
7355 0000   T25BUF, ZBLOCK 41
7356 0000   T25BUF, ZBLOCK 41
7357 0000   T25BUF, ZBLOCK 41
7360 0000   T25BUF, ZBLOCK 41
7361 0000   T25BUF, ZBLOCK 41
7362 0000   T25BUF, ZBLOCK 41
7363 0000   T25BUF, ZBLOCK 41
    
```

```

7364 0000   T25BUF, ZBLOCK 41
7365 0000   T25BUF, ZBLOCK 41
7366 0000   T25BUF, ZBLOCK 41
7367 0000   T25BUF, ZBLOCK 41
7370 0000   T25BUF, ZBLOCK 41
7371 0000   T25BUF, ZBLOCK 41
7372 0000   T25BUF, ZBLOCK 41
7373 0000   T25BUF, ZBLOCK 41
7374 0000   T25BUF, ZBLOCK 41
7375 0000   T25BUF, ZBLOCK 41
7376 0000   T25BUF, ZBLOCK 41
7377 0000   T25BUF, ZBLOCK 41
7400 0000   T25BUF, ZBLOCK 41
7401 0000   T25BUF, ZBLOCK 41
7402 0000   T25BUF, ZBLOCK 41
7403 0000   T25BUF, ZBLOCK 41
7404 0000   T25BUF, ZBLOCK 41
7405 0000   T25BUF, ZBLOCK 41
7406 0000   T25BUF, ZBLOCK 41
7407 0000   T25BUF, ZBLOCK 41
7410 0000   T25BUF, ZBLOCK 41
7411 0000   T25BUF, ZBLOCK 41
7412 0000   T25BUF, ZBLOCK 41

          DCIAL

/WAIT ROUTINES!
/ THE CALLING SEQUENCE IS:
/WAIT 1 (OR WAIT 2)
/CONDITION BIT (1 OR 0) DESIRED FOR CONDITION CHECKED,
/TEST CALL (MUST BE 2 LOCATIONS - USE A NOP IF NECESSARY)
/ THE TEST CALL MUST BE ONE OF THE LITTLE TEST ROUTINES USED
/ FOR SETTING ERSTAT.

/WAIT ROUTINES EXIT WHEN THE CONDITION IS MET OR WHEN
/ TIME RUNS OUT, WHICHEVER OCCURS FIRST, EXIT RETURNS TO LAST
/ ITEM IN CALL+1.

/ APPROXIMATE WAITING TIMES (DEPENDENT ON TEST CALL),

/          MINIMUM          MAXIMUM
/WAIT1  350 MILLISEC.    690 MILLISEC.
/WAIT2   4.5 SEC.         9 SEC.

7413 0000   WAIT1R, 0           /CHECK FOR CONDITION TO BE
7414 1213   TAD   WAIT1R         /SATISFIED 4296 TIMES,
7415 3225   DCA   WAITX
7416 7240   CLA  CMA
7417 5226   JMP   WAITX+1

7420 0000   WAIT2R, 0           /CHECK FOR CONDITION TO BE
7421 1220   TAD   WAIT2R         /SATISFIED 49K TIMES,
7422 3225   DCA   WAITX
7423 1300   TAD   N14
    
```

```

7424 5226          JMP      WAITX+1
7425 0000      WAITX, 0
7426 3220          DCA      WAIT2R      /WAIT SERVICE,
7427 3213          DCA      WAIT1R      /STORE AC IN MULTIPLIER,
7430 1157          TAD      SLKNST      /SET MULTIPLICAND TO 4996,
7431 3276          DCA      WAITTH      /GET SCOPE LOOP CONSTANT
7432 3157          DCA      SLKNST      /AND SAVE,
7433 1625          TAD I   WAITX      /ZERO SCOPE LOOP CONSTANT,
7434 3275          DCA      WAITBT      /GET CONDITION BIT AND SAVE,
7435 2225          ISZ     WAITX
7436 1625          TAD I   WAITX      /GET FIRST PART OF TEST CALL
7437 3244          DCA      ,+5      /AND INSERT IN LOOP,
7440 2225          ISZ     WAITX
7441 1625          TAD I   WAITX      /GET SECOND PART OF TEST CALL
7442 3245          DCA      ,+3      /AND INSERT IN LOOP,
7443 2225          ISZ     WAITX      /UPDATE TO EXIT POINTER,
7444 0000          0
7445 0000          0      /FIRST TEST CALL WORD,
7446 7240          CLA CMA      /SECOND TEST CALL WORD,
7447 1164          TAD      ERTAL      /BACK BIAS ERTAL,
7450 3164          DCA      ERTAL
7451 1677          TAD I   ERFXP1
7452 7041          CIA
7453 1275          TAD      WAITBT      /COMPARE CONDITION BIT
7454 7650          SNA CLA      /WITH REAL TEST BIT,
7455 5262          JMP      ,+5
7456 2213          ISZ     WAIT1R      /IF EQUAL EXIT,
7457 5244          JMP      ,+13      /COUNT 1 OFF OF MULTIPLICAND,
7460 2220          ISZ     WAIT2R      /IF NOT 0, TEST AGAIN,
7461 5244          JMP      ,+15      /IF 0, DECREASE MULTIPLIER,
7462 1276          TAD      WAITTH      /IF NOT 0, TEST AGAIN,
7463 3157          DCA      SLKNST      /IF 0, TIME IS UP, RESTORE
7464 1157          TAD      SLKNST      /SCOPE LOOP CONSTANT,
7465 1164          TAD      ERTAL      /SCOPE LOOP WANTED FOR
7466 7640          SEA CLA      /THIS BIT TEST,
7467 5272          JMP      ,+3
7470 4515          CLEAR1
7471 5960          JMP I   SLADDR      /YES, CLEAR ALL AND
7472 2164          ISZ     ERTAL      /EXIT TO LOOP START,
7473 7000          NOP
7474 5625          JMP I   WAITX      /UPDATE ERTAL,
                               /EXIT,

7475 0000      WAITBT, 0
7476 0000      WAITTH, 0
7477 4327      ERFXP1, ERFX1
7500 7764      N14, -14
    
```

```

/INSTRUCTION DEFINITIONS:
/SPECIAL EXT MEM BASIC INSTRUCTIONS:
6221 CDF=6201      /CHANGE DATA FIELD
6224 RIF=6224      /READ INSTRUCTION FIELD,
6244 RMF=6244      /RESTORE MEMORY FIELD,

/TMBE IOT DEFINITIONS:
/LOAD IOT'S:
4472 LWCR=JMS I SLWCRP      /AC TO WC, 0 TO AC;
4473 LCAH=JMS I SLCARP      /AC TO CA, 0 TO AC;
4474 LCMH=JMS I SLCMRP      /AC TO CM, 0 TO AC;
4475 LFGH=JMS I SLFGRP      /AC0=3, 5 TO FR, 0 TO AC,
4476 LDBH=JMS I SLDBRP      /AC TO DB, 0 TO AC

/READ IOT'S:
4477 RWCR=JMS I SRWCRP      /0 TO AC, WC TO AC;
4500 RCAH=JMS I SRCARP      /0 TO AC, CA TO AC;
4501 RMRH=JMS I SRMRP      /0 TO AC, MS TO AC;
4502 RCMR=JMS I SRCMRP      /0 TO AC, CM TO AC;
4503 RFRH=JMS I SRFMRP      /0 TO AC, FR TO AC; GO BIT TO AC5, S1 TO AC6-11;
4504 RDBH=JMS I SRDBRP      /0 TO AC, DB TO AC;

/CONTROL IOT'S:
4505 SKEF=JMS I SSKEFP      /SKIP IF ERROR FLAG SET,
4506 SKCB=JMS I SSKCBP      /SKIP IF CONTROL NOT BUSY,
4507 SKTD=JMS I SSKTDP      /SKIP IF TRANSPORT DONE,
4510 SKTR=JMS I SSKTRP      /SKIP IF TUR,
4511 CLF=JMS I SCLFP      /0 TO REGISTERS AND FLAGS IF TUR; IF NOT 0 TO MTF, EF, SR
4512 CLT=JMS I SCLTP      /POWER CLEAR TO TRANSPORT,
4513 SLE=JMS I SSLEP      /SET DATA LATE ERROR,
4514 SBRM=JMS I SSBMRP      /SET ONE BREAK REQUEST,

/INSTRUCTION PSEUDO MNEMONICS DEFINITIONS:
4515 CLEA1=JMS I CLM1P
4516 CLEA4=JMS I CLM4P
4517 CLEA5=JMS I CLM5P
4527 LOAD1=JMS I LD1TP
4530 SET1=JMS I SET1P
4531 SET2=JMS I SET2P
4532 SET3=JMS I SET3P
4533 SET4=JMS I SET4P
4534 CONTNU=JMS I CONTNP
4535 TSKEF=JMS I TSKEFP
4536 TSKTD=JMS I TSKTDP
4537 TSKCB=JMS I TSKCBP
4540 TSKTR=JMS I TSKTRP
    
```

53

4541 TMS=JMS I TMSP
4542 TFS=JMS I TFSP
4543 TWC=JMS I TWCP
4544 WAIT1=JMS I WAIT1P
4545 WAIT2=JMS I WAIT2P
4520 LOOP5=JMS I LAS5P
4521 LOOP6=JMS I LAS6P
4522 LOOP7=JMS I LAS7P
4523 LOOP8=JMS I LAS8P
4524 LOOP9=JMS I LAS9P
4525 COMPAN=JMS I COMPP
4526 ERROR=JMS I CORRORP
4404 BSH=JMS I SRBSWP
4405 NQL=JMS I SRNQLP
4406 NQA=JMS I SRNQAP

/PSEUDO CONSTANT DEFINITIONS:

0000 OFFLIN=0000
1000 REWIND=1000
2000 READ=2000
3000 RDCOMP=3000
4000 WRITE=4000
5000 WEOF=5000
6000 SPCFWD=6000
7000 SPCHEV=7000
0100 GO=100
0200 ERLPCC=200
0400 XGAP=400
0000 EVEN=0000
0400 ODD=400
0200 EFI=200
0100 MTF1=100
0000 D200=0000
0001 D556=1
0002 D8007=2
0003 D8009=3
5555

/FUNCTION REGISTER CONSTANTS:

/COMMAND REGISTER CONSTANTS:

0000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111000
0400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0500 11111111 11111111 11111111 11111110 00000000 00000000 00000000 00000000
0600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0700 11111111 11111111 11111111 11111111 11111111 11111111 00000000 00000000

1000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1300 11111111 11111111 11111111 11111111 11111111 11111111 00000000 00000000
1400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1500 11111111 11111111 11111111 11111111 11111111 11111110 00000000 00000000
1600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1700 11111111 11111111 11111111 11111111 11111111 11111110 00000000 00000000

2000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11110000
2200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111000
2400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111110
2600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

3000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3300 11111111 11111111 11111111 11111111 11111111 11000000 00000000 00000000
3400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3500 11111111 11111111 11111111 11111111 11111111 11111111 11111110 00000000
3600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111100

4000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111110	00000000
4200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4300	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4500	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11100000	00000000
4600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
4700	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11110000
5200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5300	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11110000
5400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5500	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111100	00000111
5600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
5700	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11100000
6000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6300	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6500	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
6700	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
7000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
7100	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
7200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
7300	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
7400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
7500	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
7600	11111111	11111111	11111111	11000000	00000000	00000000	00000000	00000000	00000000
7700	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

AC	5471	CONTR	6331	ER16I	6027	ERFX1	4329
ACLOC	0170	CONTNU	4534	EM16J	6071	ERFX1P	7477
ACTFLG	0175	CONTR	6377	ER17A	6033	ERFX2	4330
ALTEANA	0165	CONTRP	0071	ER17R	6075	ERFX3	4331
ALTIM	5637	CKC	3264	ER17C	6037	ERLPCC	0220
AMGBE	2737	CRCROD	3310	ER17D	6041	ERMP1	6666
AMGBE1	2776	CRCXOR	3324	ER20A	6043	ERMP1P	5350
AMGBE2	2753	CRLF	6167	ER20B	6045	ERMP2	6667
AMGBE3	2777	CRLFP	0065	ER20C	6047	ERMP2P	5351
AMGBEP	0060	CS	4625	ER21A	6051	ERMP3	6670
AUTO10	0010	DZ00	0000	ER21B	6053	ERMP3P	5352
AUTO11	0011	D556	0001	ER21C	6055	ERMP4	6673
AUTO12	0012	D0007	0002	ER22A	6057	ERMP4P	5354
AUTO13	0013	D0009	0003	ER22B	6041	ERMP5	6674
AUTO14	0014	DB	5464	ER22C	6043	ERMP5P	5355
AUTO15	0015	DCR	5065	ER22D	6045	ERMP6	5561
AUTO16	2016	DLF	5036	ER22E	6047	ERMP7	5360
AUTO17	0017	DLF1	5040	ER22F	6071	ERMSG	6665
BAD	0147	DLF2	5021	ER22G	6073	ERMSGP	0266
BO	5421	DRIVE	6104	ER22H	6075	ERMSYM	5562
BELL	2971	DS	5003	ER22I	6077	ERRIP	0532
BELLP	0064	DS1	5067	ER22J	6081	ERROR	4526
BSW	4404	DS2	5070	ER22K	6083	ERRORP	0126
CA	5440	EDIT	6105	ER23A	6075	ERRORS	5200
CAF	6201	EDIT1	6116	ER23B	6077	ERRPTR	5347
CLEAR1	4515	EDITP	0062	ER23C	6011	ERSHFT	4314
CLEAR4	4516	EDTEM	6132	ER23D	6013	ERSTAT	0162
CLEAR5	4517	EDTEM1	6133	ER23E	6015	ERTAL	0164
CLF	4511	EDTEM2	6134	ER23F	6017	ES	4744
CLR1	4332	EDTEM3	6135	ER24A	6021	EVEN	0000
CLR1P	0115	EDTEMP	0063	ER24B	6023	EXEC	4400
CLR4	4353	EFI	0200	ER24C	6025	EXECFP	4560
CLR4P	0116	EF1	6006	ER25A	6027	EXECFX	6030
CLRS	4357	EFX1	0224	ER25B	6031	EXECL1	4447
CLRS5P	0117	EFX2	0224	ER25C	6033	EXECL2	4435
CLRX	4366	ER15A	6465	ER25D	6035	EXECP	0007
CLT	4512	ER15B	6465	ER26A	6037	EXITFL	0174
CM	5445	ER15C	6467	ER27A	6041	EXDPT	4474
COMP	0473	ER15D	6471	ER27B	6043	EXDPTP	4474
COMPAR	4525	ER15E	6473	ER27C	6045	EXDPTP	4474
COMPF	0524	ER15F	6475	ER27D	6047	EXDPTP	4474
COMPP	0125	ER15G	6477	ER27E	6049	EXDPTP	4474
CONTC	6140	ER15H	6501	ER27F	6051	EXDPTP	4474
CONTC1	6163	ER15I	6503	ER27G	6053	FATERR	5232
CONTC2	6164	ER15J	6505	ER27H	6055	FE31A	6451
CONTC3	6165	ER16A	6507	ER27I	6057	FS	5452
CONTC4	6166	ER16B	6511	ER27J	6059	GD	5414
CONTC5	6153	ER16C	6513	ER27K	6061	GET2N	4752
CONTC7	5345	ER16D	6515	ER27L	6063	GET2NP	5103
CONTC8	0514	ER16E	6517	ER27M	6065	GO	0100
CONTCP	0070	ER16F	6521	ER27N	6067	GOOD	0146
CONTRP	0134	ER16G	6523	ER27O	6069	HALTC	5411
		ER16H	6525	ERFX	4270	HALTCP	5413
				ERFX2	4326	II	5476

JJ

I112	5510	LAS5	1141	M316	6151	NOTSRN	4526
I2	5573	LAS5P	0120	M331	6172	NSTR	3773
IN11	4425	LAS6	1147	M4	0037	NSTRP	3775
INTAGN	5607	LAS6P	0121	M40	0042	NSTRUC	3757
INTCON	5647	LAS7	1155	M43	0043	OD	5426
INTEMR	6352	LAS7P	0122	M5	0040	ODD	0400
INTOK	5653	LAS8	1163	M50	3177	OFFLIN	0000
INTOKP	4773	LAS8P	0123	M6P1	7050	OLD	0150
INTSEP	4536	LAS9	1171	M6P1P	6074	P0	5661
INTSEV	5600	LAS9P	0124	M6P2	7052	P1	4542
IOY1	0166	LCAR	4473	M6P2P	6075	P2	4543
IOY2	0167	LCHR	4474	M7	0470	P3	4544
IS	4774	LDBR	4476	M77	1356	PARCAR	3235
K1	4561	LDPT	1742	M7P1	7061	PASCNT	4537
K10	0023	LDPTP	0127	M7P1P	4951	PRCNT1	5396
K100	0030	LFGR	4475	M9P1	7101	PRCNT2	5357
K1000	0034	LISN	5112	M9P1P	4955	PRGFLD	0196
K14	0024	LISN1	5137	M9P2	7102	PRMPT	5362
K17	0025	LISN2	5151	M9P2P	4954	PRNT	5254
K2	0020	LISN3	5165	MONIT	4000	PRNTCK	5337
K20	1177	LISNP	4772	MONITP	0061	PRNTK	5363
K200	0031	LISNT1	5173	MONP	5111	PRNTLP	5321
K2000	4562	LISNUM	5154	MOA	4406	PRNTP	0533
K207	2576	LOADPT	4527	MDL	4405	QHALT	5400
K212	6177	LOOP5	4520	MS	5457	QHALTP	5366
K215	0032	LOOP6	4521	HSG1	6776	OPRNT	5241
K2252	2535	LOOP7	4522	HSG11	7104	QUES	5106
K3	0472	LOOP8	4523	HSG12	7106	QUESP	4771
K377	5565	LOOP9	4524	HSG13	7116	RCAR	4903
K4	0021	LHCR	4472	HSG14	7120	RCHR	4902
K40	0026	ME	5646	HSG15	7124	RDBR	4904
K400	0033	M1	0471	HSG16	7126	RDCOMP	3000
K4040	0035	M10	1357	HSG17	7130	READ	2000
K4300	5367	M100	5644	HSG3	7026	HESTP	5703
K4324	0536	M11	0041	HSG4	7033	KESTPP	4770
K4343	5370	M12	5645	HSG6	7043	REWIND	1000
K4352	5371	M14P1	7121	HSG61	7043	REX	4430
K522	5372	M14P1P	5104	HSG7	7056	REXP	4765
K605	5373	M14P2	7122	HSG8	7043	RFSR	4903
K6060	6136	M14P2P	5105	HSG9	7076	RIF	6224
K6740	6076	M17	5643	MTF1	0100	RMF	6244
K7	0022	M2	0036	MTM2	7137	RMSR	4501
K700	6137	M20	5724	MTM3	7143	RWCR	4477
K72	5564	M2000	6103	MTM4	7166	SAVEAC	0154
K77	0027	M203	0044	MTM5	7206	SAVEL	0155
K7700	5374	M222	0045	MTM6	7242	SAVEP	5662
K8	6046	M240	5174	MTM7	7303	SAVEP1	5725
K800T	6057	M260	0046	N10	6391	SAVEP2	5726
KCDF	0050	M267	6077	N14	7000	SAVEP3	5727
KCMD	0163	M270	0047	N400	3350	SAVEP4	5730
KJMP13	4535	M271	6100	NOEXLP	4927	SAVEPP	4767
KRMF	4540	M3	3351	NOTSPN	4470	SBRM	4514

SCDF	4375	SRCARP	0100	T15J	0342	T22EX4	2134
SCDF1	4376	SRCMR	4056	T15LS1	0445	T22EX5	2135
SCDFIP	4541	SRCMRP	0102	T15LS2	0457	T22EX6	2136
SCDFP	0057	SHOBR	4066	T15T41	0444	T22F	2137
SCLF	4116	SROBRP	0104	T15X1	0367	T22G	2200
SCLFP	0111	SFRSR	4062	T15X2	0370	T22GL	2212
SCLT	4122	SFRSRP	0103	T15X3	0371	T22H	2236
SCLTP	0112	SRHOLD	5731	T15X4	0372	T22I	2262
SOLE	4513	SRMQA	3957	T15X5	0373	T22J	2314
SET1R	4530	SRMQAP	0006	T15X6	0374	T22JK	2335
SET1RP	6200	SRMQL	3954	T16A	0001	T22JK1	2350
SET2	0130	SRMQLP	0005	T16B	0641	T22JKJ	2360
SET2R	4531	SRMQLT	3966	T16C	0645	T22K	2321
SET2RP	6207	SRMSR	4052	T16D	0652	T22L7P	2374
SET3	0131	SRMSRP	0101	T16E	0660	T22LP7	2165
SET3R	4532	SRTS	5752	T16F	0665	T22LS	2171
SET3RP	6216	SRWCR	4042	T16G	0673	T23A	2401
SET4	0132	SRWCRP	0077	T16H	0677	T23B	2407
SET4R	4533	SSBRM	4132	T16I	0705	T23C	2415
SET4RP	6224	SSBRMP	0114	T16J	0711	T23D	2425
SETA	0133	SSDLE	4126	T17A	1201	T23E	2433
SETAHP	6232	SSDLEP	0113	T17AL1	1276	T23EX1	2455
SETB	4254	SSKCR	4077	T17ATH	1212	T23EX2	2510
SETC	6270	SSKCRP	0106	T17B	1234	T23F	2443
SETD	6277	SSKEF	4072	T17C	1279	T23FN1	2473
SETE	6303	SSKEFP	0105	T17CL1	1301	T23FN2	2476
SETI	6330	SSKTD	4104	T17CLP	1424	T23FN3	2526
SETTST	6313	SSKTD0	0107	T17CP	1423	T23FN4	2531
SKCB	4506	SSKTR	4111	T17CTM	1395	T24A	2631
SKEF	4505	SSKTRP	0110	T17CTP	1425	T24B	2636
SKIPER	4136	START	0200	T17D	1327	T24C	2666
SKTD	4507	SYHADR	5353	T17DCK	1400	T24EX1	2720
SKTR	4510	SYMBOL	5361	T17DP	1426	T24T1	2736
SL	5071	SYMPT	6405	T17LP6	1355	T25A	3027
SLADDR	0160	SYMPTP	5364	T17LP7	1413	T25ASL	3045
SLCAR	4022	SYMSEV	5526	T20A	1430	T25B	3103
SLCARP	0073	SYMTEH	5563	T20B	1457	T25BUF	7342
SLCMR	4026	T19A	0204	T20C	1912	T25BUP	3176
SLCMRP	2074	T19B	0215	T21A	1001	T25C	3144
SLDBR	4036	T19C	0225	T21B	1033	T25CCK	3167
SLDBRP	0076	T19D	0240	T21C	1033	T25T42	3175
SLFGR	4032	T19E	0253	T21D	1067	T26A	3205
SLFGRP	2075	T19EX1	0400	T22A	2001	T26ASL	3243
SLKNST	0157	T19EX2	0413	T22B	2021	T26SC1	3253
SLWCR	4016	T19EX3	0417	T22C	2044	T27A	3404
SLWCRP	0072	T19EX4	0423	T22D	2064	T27B	3411
SPCFWD	6000	T19EX5	0434	T22DE	2101	T27C	3420
SPGREV	7000	T19EX6	0437	T22DE1	2111	T27D	3431
SPRSW	3541	T19F	0262	T22DE2	2131	T27EX1	3473
SPRSWP	0004	T19G	0277	T22DEJ	2117	T27EX2	3517
SPRSWT	3553	T19H	0313	T22DEJ	2122	T30A	3601
SRCAR	4046	T19I	0326	T22E	2072	T30B	3645

56

T3PC	3656	TSPREP	0756
T3PD	3673	TSPREV	1246
T3PE	3706	TST	0421
T3PF	3721	TST17P	1422
T3PFIN	3753	TST22P	2345
T3PFOK	3727	TSTAEX	4550
T3PG	4000	TSTAT	0176
TA	4727	TSTATH	4766
TAL	6435	TSTEM1	4764
TALP	4557	TSTNUM	0177
TALPP	4547	TSTP	4556
TCR	4712	TSTPP	4546
TEST15	0203	TSTAL	4592
TEST16	0600	TTOFLG	0173
TEST17	1200	TWC	4543
TEST20	1427	TWCP	0143
TEST21	1600	TWCR	4296
TEST22	2000	TXXTM1	0191
TEST23	2400	TXXTM2	0192
TEST24	2600	TXXTM3	0193
TEST25	3000	TYPE	2593
TEST26	3200	TYREP	0067
TEST27	3400	TYPTCH	2570
TEST30	3600	WAIT1	4544
TFS	4542	WAIT1P	0144
TFSP	0142	WAIT1R	7413
TFSR	4243	WAIT2	4545
TMOFWD	1000	WAIT2P	0145
TMOFWP	0755	WAIT2R	7420
TMS	4541	WAITBT	7475
TMSP	0141	WAITTM	7476
TMSR	4230	WAITX	7425
TR	4700	WC	5433
TR31D	6457	WEOF	5000
TRACE	0172	WHAT	5365
TRKY	0161	WHITE	4000
TRPP	0534	XBUFF	0051
TS	4642	XGAP	0400
TSKCB	4937	YBUFF	3336
TSKCBP	0137		
TSKCBR	4214		
TSKEF	4935		
TSKEFP	0135		
TSKEFR	4200		
TSKTD	4936		
TSKTD P	0136		
TSKTD R	4206		
TSKTR	4540		
TSKTRP	0140		
TSKTRR	4222		
TSPFWD	1102		
TSPFWP	0757		

ERRORS DETECTED: 0
LINKS GENERATED: 0
RUN-TIME: 38 SECONDS
4K CORE USED

