```
                IDENTIFICATION
                --------------

PRODUCT CODE:   MAINDEC-Ø8-DHTMA-A-D

PRODUCT NAME:   TM8-E CONTROL TEST PART 1

DATE CREATED:   DECEMBER 4, 1972

MAINTAINER:     DIAGNOSTIC GROUP

AUTHOR:         LEONARD E. BEYERSDORFER
```

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS, Ø1754

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-Ø8-DHTMA)

2. TM8-E CONTROL TEST PART 2 (MAINDEC-Ø8-DHTMB)

3. TM8-E DRIVE FUNCTION TIMER (MAINDEC-Ø8-DHTMC)

4. TM8-E DATA RELIABILITY 9 TRACK (MAINDEC-Ø8-DHTMD)

5. TM8-E DATA RELIABILITY 7 TRACK (MAINDEC-Ø8-DHTME)

6. TM8-E RANDOM EXERCISER (MAINDEC-Ø8-DHTMF)

TABLE OF CONTENTS

1. ABSTRACT

THE TM8-E CONTROL TEST PART 1 IS AN INTEGRATED SERIES OF SUBTESTS DESIGNED TO AID IN THE CHECKOUT AND MAINTENANCE OF THE TM8-E DECMAGTAPE CONTROL WITHOUT DEPENDENCE ON TRANSPORT SYSTEM USE. VERSATILITY OF USEAGE IS AFFORDED THROUGH A MODEST KEYBOARD MONITOR AND SWITCH REGISTER CONTROL OPTIONS.

THIS PROGRAM CONSISTS OF 12 MAJOR TESTS (TEST 01 THROUGH TEST 14) EACH OF WHICH CONSISTS OF A NUMBER OF SUBTESTS DESIGNATED BY THE LETTERS A THROUGH Z.

2. REQUIREMENTS

2.1 HARDWARE

PDP-8/E, 8/M, 8/F
TELETYPE OR COMPATIBLE DEVICE (TTY)
TM8-E DECMAGTAPE CONTROL

2.2 MEMORY

THIS PROGRAM REQUIRES 4K OF MEMORY AND MAY RESIDE IN ANY MEMORY FIELD. ALL OF EXISTING MEMORY IS USED TO TEST THE ABILITY OF THE TM8-E TO ADDRESS CORRECTLY DURING DMA TRANSFERS.

2.3 PRELIMINARY PROGRAMS

ALL PROCESSOR/MEMORY DIAGNOSTICS

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,

THE PROGRAM PRINTS A LEFT BRACKET ([) WHEN IT IS READY TO ACCEPT A KEYBOARD COMMAND,

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 Ø2ØØ,

B, LOAD THE EXTENDED ADDRESS WITH THE PROGRAM FIELD,

C, SET THE SR TO 4ØØØ,

D, DEPRESS CLEAR, THEN CONTINUE,

E, THE PROGRAM WILL PRINT ITS TITLE AND MAINDEC NUMBER, THEN ASK "EXTENDED MEMORY?(Ø-7)", TYPE THE NUMBER OF THE HIGHEST EXISTING MEMORY FIELD, THEN THE PROGRAM MONITOR WILL ASSUME CONTROL, GO TO THE STANDARD TEST PROCEDURE IN PARAGRAPH 6,

NOTE: THE PROGRAM MAY BE RESTARTED AT ANY TIME AT ADDRESS Ø2Ø1, 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 CHECKOUT. ANY ERROR OCCURRENCE RESULTS IN AN ERROR REPORT ON THE TTY AND IN A RETURN TO THE PROGRAM MONITOR. ALL OPTIONAL 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,

ACCOMPLISH THE FOLLOWING STEPS,

A. START THE PROGRAM AS DESCRIBED IN PARAGRAPH 5,

B. IF A TU1Ø TRANSPORT SYSTEM IS CONNECTED, ENSURE THAT ALL DRIVES ARE OFF LINE, ALL LEVELS INPUT FROM THE TU1Ø TO THE TM8-E SHOULD BE AT A "HIGH" LEVEL,

C. SET THE SR=ØØØ1.

D. TYPE "TA←" WHICH RESULTS IN THE EXECUTION OF TEST Ø1 THROUGH TEST 14. SEVERAL PASSES WILL BE MADE OF EACH TEST,

E. AFTER ALL TESTS HAVE BEEN EXECUTED, THE PROGRAM WILL PRINT "PASS ØØØ1", THEN ANOTHER PASS IS AUTOMATICALLY STARTED,

F. ALLOW THE PROGRAM TO RUN FOR AT LEAST 2 PASSES, EACH PASS TAKES APPROXIMATELY 15 MINUTES,

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 ([), THE TEST INTERRUPT COMMAND MAY BE TYPED AT ANY TIME,

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

A. AFTER PROGRAM STARTUP WITH SR Ø=1, OR A RESTART AT Ø2Ø1,

B, AN ERROR OCCURS AND SR2=Ø,

C. ALL SELECTED TESTS ARE DONE.

D, THE 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 Ø1 THROUGH TEST 14)
TØ1T1ØTNN←	RUN THE TEST(S) INDICATED, TØ1 THROUGH T14 IN OCTAL ARE THE VALID SELECTIONS, IF THE USER SELECTS OTHER THAN TØ1 THROUGH T14 THE PROGRAM WILL EXECUTE ONE OF THE VALID TESTS (UNDETERMINED),

7,1.2 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=Ø.

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,

7,1,3 TEST INTERRUPT COMMAND

THE FOLLOWING COMMAND MAY BE USED TO INTERRUPT TESTING AND RETURN TO THE PROGRAM MONITOR.

COMMAND	RESULT
ALTMODE KEY	INTERRUPT TEST EXECUTION, RESPONDS TO "ALT" AND "ESC" KEYS. TEST SELECTION IS UNAFFECTED,

7,1,4 MISCELLANEOUS COMMANDS

COMMAND	RESULT
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,
DM:NNNN←	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,
        1      QUICK VERIFY (ONE PASS IS MADE OF EACH SE-
               LECTED 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      DISABLE TEST IN PROGRESS REPORTS,

2       0      RETURN TO PROGRAM MONITOR ON ANY ERROR
               OCCURRENCE,
        1      RETURN TO PROGRAM MONITOR ONLY IF A FATAL
               ERROR HAS OCCURRED,

3       0      ENABLE ERROR AND TRACE REPORTS,
        1      DISABLE

4       0      ERROR AND TRACE REPORTS INCLUDE APPLICABLE
               DATA ONLY,
        1      FORCE ERROR AND TRACE REPORTS TO INCLUDE ALL
               POSSIBLE DATA.

5       0      DISABLE SUBTEST LOOP,
        1      LOOP ON CURRENT SUBTEST,  ALL VARIABLE DATA
               REMAINS CONSTANT,

6-9     0      INHIBIT LOOPS 6-9 RESPECTIVELY,
        1      ENABLE LOOPS 6-9 RESPECTIVELY,  LOOPS 6-9 AL-
               LOW THE USER TO LOOP ON A SMALL GROUP OF SUB-
               TESTS 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 01 - TEST 14),
        1      ENABLE LOOP ON CURRENT TEST,

11      0      TERMINATE TESTING UPON COMPLETION OF ALL
               SELECTED TESTS AND DELETE TEST SELECTION,
        1      RUN ALL SELECTED TESTS CONTINUOUSLY,  AT
               THE COMPLETION OF EACH PASS THROUGH THE EN-
               TIRE TEST SELECTION, THE NUMBER OF PASSES EX-
               ECUTED (IN OCTAL) IS PRINTED (PASS NNNN),
```

//

8, ERRORS

8.1 ERROR HALTS

THERE ARE NO ERROR HALTS IN THIS PROGRAM, IF AN ERROR SHOULD OCCUR AND SR2=Ø, 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=Ø, 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.

```
*ERØ1D  PC:Ø25Ø I1:67Ø6 I2:6716 GD:ØØØØ BD:7777 OD:7777
WC:1234 CA:2345 CM:ØØØØ FS:ØØØØ MS:ØØØØ DB:ØØØØ AC:ØØØØ
```

THE SYMBOLS USED IN THE ERROR REPORTS ARE DEFINED BELOW.

SYMBOL DEFINITION

*ERNNX ERROR OCCURRED IN TEST NN, SUBTEST X. (IF NN=ØØ, THE ERROR OCCURRED OUTSIDE OF A FORMAL TEST.)

*FENNX FATAL ERROR

TR*NNX TRACE REPORT INDICATOR. NO ERROR OCCURRED BUT TRACE IS ENABLED.

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 (ERNNX OR FENNX),

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 ERROR IS FATAL (FENNX), THERE IS NO STANDARD RECOVERY. THE CAUSE OF THE FAILURE MUST BE DETERMINED THROUGH STATIC MEANS,

F. IF THE ERROR IS NON-FATAL (ERNNX), THE USER MAY ELECT TO CONTINUE IN THE TEST SEQUENCE (C←), OR TO ENTER A SUBTEST LOOP AS DESCRIBED BELOW,

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.

9. RESTRICTIONS

NONE.

1Ø. EXECUTION TIME

ONE NORMAL PASS OF ALL TESTS TAKES APPROXIMATELY 15 MINUTES, A QUICK VERIFY PASS (SRØ=1) TAKES ONLY 2 MINUTES,

11. TEST ABSTRACTS

TEST Ø1 VERIFIES THE EXISTENCE OF THE VARIOUS 67ØX AND 671X IOT'S EXCEPT RMSR (6714), AND TESTS THE BASIC OPERATION OF ALL TM8-E REGISTERS EXCEPT THE MAIN STATUS AND STATUS PORTION OF THE FUNCTION/STATUS REGISTERS.

TEST Ø2 IS A WORST CASE DATA TEST, DATA IS LOADED INTO AND READ BACK FROM ALL NON-STATUS REGISTER BITS AND VERIFIED,

TEST Ø3 VERIFIES THE EXISTENCE OF THE VARIOUS FUNCTIONS (AS FAR AS POSSIBLE WITHOUT A TRANSPORT) BY CHECKING EACH FUNCTION'S AFFECT ON CURRENT ADDRESS AND WORD COUNT INCREMENTING, AS WELL AS VERIFYING THAT DATA IS TRANSFERRED BETWEEN THE CONTROL AND THE PROCESSOR IN THE PROPER DIRECTION DURING DMA TRANSFERS,

TEST Ø4 TESTS THE ABILITY OF WORD COUNT, CURRENT ADDRESS AND COMMAND REGISTER BITS 6-8 TO INCREMENT PROPERLY,

TEST Ø5 VERIFIES THAT PROPER DATA TRANSFERS ARE MADE BETWEEN THE DATA BUFFER AND MEMORY IN BOTH DIRECTIONS, 6 C4)

TEST Ø6 VERIFIES THAT THE CURRENT ADDRESS REFERENCES THE CORRECT MEMORY LOCATION AND THAT COMMAND REGISTER BITS 6-8 REFERENCE THE CORRECT MEMORY FIELD. ALL EXISTING FIELDS ARE UTILIZED IN THIS TEST,

TEST Ø7 TESTS THE VARIOUS STATUS BITS TO CONTAIN THE PROPER STATUS AS GOVERNED BY THE "NO TRANSPORT" CONDITION,

TEST 1Ø CHECKS VARIOUS POSITIVE AND NEGATIVE CONDITIONS OF MAG TAPE FLAG, ERROR FLAG, GO BIT, ILLEGAL FUNCTION, READ-COMPARE ERROR AND EMA OVERFLOW ERROR AS A NO TRANSPORT CONDITION WILL ALLOW,

TEST 11 EXERCISES THE IOT'S SKEF AND SKTD UNDER INTERESTING CONDITIONS AS WELL AS OTHER SKIP AND NO SKIP IOT'S IN THE 672X IOT DECODER,

TEST 12 CHECKS THE UNIQUENESS OF INTERRUPTS AS CAUSED BY MAG TAPE FLAG AND THE ERROR FLAG,

TEST 13 VERIFIES THE UNIQUENESS OF ALL 672X IOT'S IN RELATION TO CLF AND SBRM,

TEST 14 COMPLETELY TESTS THE PROPER OPERATION OF THE READ-COMPARE LOGIC,

12. LISTING (ATTACHED)

```
                    /TM8E CONTROL TEST PART 1  MAINDEC-Ø8-DHTMA-A-L

                    /COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

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

                    /THIS DIAGNOSTIC PROGRAM TESTS AS MUCH OF THE TM8E CONTROL AS
                    /IS POSSIBLE WITHOUT UTILIZING A TRANSPORT SYSTEM AND MAY
                    /RESIDE IN ANY EXISTING MEMORY FIELD.

                    /BASIC STARTING PROCEDURE:
                            /POWER DOWN THE TRANSPORT SYSTEM AS NECESSARY
                            /TO ENSURE ALL SIGNALS RETURNING FROM THE TRANSPORT SYSTEM ARE HIGH.
                            /LOAD THE PROGRAM INTO THE DESIRED MEMORY FIELD WITH THE BINARY LOADER.
                            /LOAD THE INSTRUCTION AND DATA FIELDS WITH THE PROGRAM FIELD.
                            /LOAD ADDRESS 2ØØ.
                            /CLEAR ALL SWITCHES.
                            /SET SR BIT Ø IF IT IS DESIRED TO SELECT TESTS WITH THE MONITOR.
                            /IF SRØ IS CLEAR, ALL TESTS WILL BE RUN.
                            /SET ANY OTHER SR OPTIONS DESIRED.
                            /DEPRESS "CLEAR", THEN "CONTINUE".

                    /SWITCH REGISTER CONTROL OPTIONS.

                            /SR BIT STATE   CONTROL
                            /Ø!     Ø       RUN ALL TESTS NORMALLY.
                            /       1       IF SET TO 1 ANYTIME PRIOR TO THE QUICK
                            /               VERIFY MESSAGE, SELECT TESTS WITH MONITOR.
                            /               IF SET TO 1 AFTER QUICK VERIFY MESSAGE, RUN QUICK VERIFY.
                            /               (1 PASS OF EACH SELECTED TEST).
                            /1!     Ø       TEST PROGRESS REPORTS.
                            /       1       NO TEST PROGRESS REPORTS.
                            /2!     Ø       GO TO MONITOR ON ANY ERROR OCCURRENCE.
                            /       1       GO TO MONITOR ONLY UPON FATAL ERROR.
                            /3!     Ø       PRINT ERROR MESSAGES AND IF "TR" IS PENDING ENABLE TRACE MESSAGES.
                            /       1       PRINT ONLY FATAL ERROR MESSAGES.
                            /4!     Ø       ERROR MESSAGES DISPLAY NECESSARY DATA ONLY.
                            /       1       ERROR MESSAGES DISPLAY ALL DATA WHETHER APPLICABLE
                            /               OR NOT. THIS SWITCH SHOULD BE USED ONLY
```

```
        /                   IF TEST Ø1 PASSES,
/NOTE:  IF SR5=1 WHEN EXITING MONITOR TO RUN A NEW TEST SELECTION, THE
/PROGRAM MAY HANG UP IN THE ERØØB TEST, RELEASING THIS SWITCH MOMENTARILY
/WILL ALLOW THE PROGRAM TO CONTINUE WITH THE NEW SELECTED TESTS,
        /5!     Ø           NO SUBTEST LOOP,
        /       1           LOOP ON SUBTEST,  (SCOPE LOOP)
/LOOPS 6-9 ARE SELECTED BY SR 6-9, THESE LOOPS ALLOW THE USER TO LOOP
/ON A SMALL GROUP OF SUBTESTS WITHIN THE CURRENT TEST, THEREBY INCREASING
/THE TESTING FREQUENCY OF A SELECTED FUNCTION, THE LISTING MUST BE REFERENCED
/TO DETERMINE WHICH SUBTESTS ARE INCLUDED IN EACH LOOP,
/IF LOOP N IS SELECTED, SETTING LOOP N-1 SWITCH WILL TRANSFER TO THAT
/LOOP UPON COMPLETION OF THE PRESENT LOOP, BY THIS MEANS IT IS POSSIBLE
```

```
/TO BACKTRACK IN A GIVEN TEST,
        /6-9:   Ø       NO LOOP 6-9 RESPECTIVELY,
        /       1       LOOP 6-9 RESPECTIVELY,  (LOOP ON SMALL GROUP
        /               OF SUBTESTS),
        /1Ø:    Ø       NO LOOP ON CURRENT TEST (TØ1-T14),
        /       1       LOOP ON CURRENT TEST (TØ1-T14),
        /11:    Ø       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.
                        /NOTES! 1.  THE FOLLOWING COMMANDS MAY BE ENTERED WHEN "[" IS
                /TYPED.
                /  (UPON ENTRY MONITOR TYPES THE CONTENTS OF LOCATION Ø, AND THE TEST
                /THAT IS CURRENTLY BEING RUN, THEN "[")
                        /2.  "←" INDICATES "RETURN" KEY; "RUBOUT" ABORTS A COMMAND STRING
                /PRIOR TO "←"; "SPACE" MAY BE INPUT AT ANY POINT.
                        /3.  "ALTMODE" INTERRUPTS THE MAIN PROGRAM AND RETURNS TO MONITOR
                /ANYTIME THAT THE INTERRUPT INDICATOR IS ON.

                /COMMAND          RESULT
                /"TØ1TØ2TØ5TNN←"RUN THE TESTS INDICATED.  IF A TEST NUMBER IS INPUT 2
                /                 TIMES, THAT TEST IS NOT RUN.
                /"TA←"            RUN ALL 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 OCCURING PREVENTING CONTINUING
                /                 ON QUICKLY WITH THE NEXT SUBTEST, DEPRESS "ALTMODE"
                /                 (OR ALLOW RETURN TO MONITOR ON ERROR IF THE INTERRUPT SYSTEM
                /                 IS OFF DURING THIS TEST.)
                /                 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 IF ERROR DOES NOT OCCUR.
                /                 "TR" MAY PRECEED "TA","TNN","C", OR "EX" COMMANDS.
                /                 TRACE REMAINS IN EFFECT UNTIL "[" 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.
                /"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
/------         ----------

/*ERNNX         ERROR OCCURRED IN TEST NN SUBTEST X(TNNX),
/*FENNX         SAME AS ABOVE EXCEPT FATAL ERROR,
/TR*NNX         TRACE PRINTOUT (FROM "TR" MONITOR COMMAND),
/PC:NNNN        ADDRESS IN PROGRAM AT WHICH ERROR OCCURRED,
/I1:NNNN        OCTAL CODE FOR IOT1,
/I2:NNNN        OCTAL CODE FOR IOT2,
/GD:NNNN        GOOD TEST VALUE, DESCRIBED IN SUBTEST LISTING,
/BD:NNNN        BAD OR ACTUAL TEST VALUE, DESCRIBED IN SUBTEST LISTING,
/OD:NNNN        USUALLY PREVIOUS GOOD, DESCRIBED IN SUBTEST LISTING,
/THE FOLLOWING DISPLAY THE CONTENTS OF THE INDICATED REGISTER:
/WC:NNNN        WORD COUNT,
/CA:NNNN        CURRENT ADDRESS,
/CM:NNNN        COMMAND,
/FS:NNNN        FUNCTION - STATUS,
/MS:NNNN        MAIN STATUS,
/DB:NNNN        DATA BUFFER,
/AC:NNNN        ACCUMULATOR (USED FOR SKIP ERRORS ONLY),
                    /PROGRAM DIRECTORY:
                    NOPUNCH
          76ØØ      *76ØØ
                              /REFER TO ASSEMBLY COLUMN 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.

                              /TEST SECTION:
    76ØØ  Ø2Ø3      TESTØ1    /67ØX,671X IOT EXISTENCE AND BASIC WC,CA,CM,DB,FR REGISTER TEST,
    76Ø1  Ø6ØØ      TESTØ2    /WC,CA,CM,DB,FR REGISTER DATA TEST,
    76Ø2  1ØØØ      TESTØ3    /BASIC FUNCTIONS EXISTENCE TEST,
    76Ø3  1235      TESTØ4    /WC,CA REGISTERS INCREMENT TEST (INCLUDES CM6-8),
    76Ø4  14ØØ      TESTØ5    /MEMORY-DB DATA TEST,
    76Ø5  16ØØ      TESTØ6    /CA REGISTER MEMORY REFERENCE TEST (USES ALL EXISTING MEMORY FIELDS),
    76Ø6  2ØØØ      TESTØ7    /STATUS REGISTERS STAND ALONE TEST,
    76Ø7  22ØØ      TEST1Ø    /BASIC TESTS FOR MTTF, ILLEGAL FUNCTION, READ COMPARE, AND EF,
    761Ø  3ØØØ      TEST11    /MTTF AND EF SKIP-NO SKIP TEST,
    7611  32ØØ      TEST12    /MTTF-EF INTERRUPT TEST,
    7612  34ØØ      TEST13    /672X IOT UNIQUENESS TEST,
    7613  36ØØ      TEST14    /COMPLETE READ COMPARE TEST,

                              /PROGRAM CONTROL SECTION:
    7614  42ØØ      EXEC      /EXECUTIVE.
    7615  44ØØ      MONIT     /MONITOR,
    7616  54ØØ      INTSEV    /INTERRUPT SERVICE,

                              /ERROR HANDLERS:
    7617  5ØØØ      ERRORS    /ERROR HANDLING ROUTINE,
    762Ø  56ØØ      COMP      /ERROR DETECTOR,
    7621  6677      FEØØA     /ERROR TABLE,

                              /UTILITIES:
    7622  4ØØØ      S67ØØ     /TM8E IOT SUBROUTINES,
    7623  1624      MARK3     /(IF THE DEVICE CODES ARE
                              /NOT 7Ø-72 IN YOUR SYSTEM
                              /CHANGE ONLY THE IOT CODES IN
                              /EACH OF THE 23 SUBROUTINES AND IN THE LOCATION MARK3,)
    7624  61Ø5      LAS5      /LOOP SENSORS,
    7625  62ØØ      AMG8E     /PACKED TEXT MESSAGE GENERATOR,
    7626  6244      TYPE      /TTY ROUTINES,
    7627  63Ø1      EDIT      /OCTAL NUMBER OUTPUT EDITOR,
    763Ø  64ØØ      CLR1      /"INITIALIZE" GENERATORS,
    7631  6432      OFFLNE    /FUNCTION ROUTINES,

                              /DATA SECTION:
    7632  6613      T1LS1     /TESTØ1 IOT LISTS,
    7633  6647      T3LS1     /TESTØ3 FUNCTION LIST,
    7634  3146      T11LS1    /TEST11 IOT LIST
    7635  3337      T12LS1    /TEST12 INTERRUPT ENABLE LIST,
    7636  3146      T11LS1    /TEST13 IOT LIST,
    7637  ØØØØ      ØØØØ      /CONSTANTS, POINTERS, ETC,
    764Ø  7163      MSG1      /MESSAGE TEXTS,
                    ENPUNCH
          0000    *0000
    0000  0000            0                    /LOCATIONS 1-3 ARE ASSEMBLED SET
    0001  5001            JMP      1           /UP FOR HIGH SPEED DUMP,  THEY
    0002  0002            2                    /ARE CHANGED FOR INTERRUPT HANDLING
    0003  0003            3                    /BY THE PROGRAM,
    0004  1200    T3LODP, T3LOAD
    0005  2003    K3,     3


          0020    *0020

                  /WIDE USEAGE NUMERIC CONSTANTS,  (POSITIVE)
    0020  0001    K1,     1
    0021  0002    K2,     2
    0022  0007    K7,     7
    0023  0010    K10,    10
    0024  0020    K20,    20
    0025  0040    K40,    40
    0026  0070    K70,    70
    0027  0100    K100,   100
    0030  0200    K200,   200
    0031  0215    K215,   215
    0032  0400    K400,   400
    0033  2000    K2000,  2000
    0034  4000    K4000,  4000
    0035  4040    K4040,  4040
    0036  7400    K7400,  7400

                  /WIDE USEAGE NUMERIC CONSTANTS,  (NEGATIVE)
    0037  7776    M2,     -2
    0040  7774    M4,     -4
    0041  7773    M5,     -5
    0042  7764    M14,    -14
    0043  7760    M20,    -20
    0044  7520    M260,   -260
    0045  7510    M270,   -270

                  /WIDE USEAGE INSTRUCTIONAL CONSTANTS!
    0046  6201    KCDF,   CDF

                  /TRIAL BUFFER AREA.
    0047  0050    XBUFFP, XBUFF
    0050  0000    XBUFF,  0
    0051  0000            0
    0052  0000            0
    0053  0000            0
    0054  0000            0

                  /WIDE USEAGE POINTERS

    0055  4200    EXECP,  EXEC
    0056  6511    FUNCP,  FUNC
    0057  4144    SCDFP,  SCDF
    0060  5463    INSRSP, INSRS
```

```
Ø061  62ØØ     AMG8EP, AMG8E
Ø062  44ØØ     MONITP, MONIT
Ø063  63Ø1     EDITP,  EDIT
Ø064  6326     EDTEMP, EDTEM
Ø065  6262     BELLP,  BELL
Ø066  627Ø     CRLFP,  CRLF
Ø067  7313     ERMSGP, ERMSG
Ø07Ø  6244     TYPEP,  TYPE
Ø071  55Ø7     EXECFP, EXECFX

               /TM8E IOT SUBROUTINE POINTERS,
Ø072  4ØØØ     S67ØØP, S67ØØ                    /LOAD IOT,
Ø073  4ØØ4     SLWCRP, SLWCR                    /LWCR
Ø074  4Ø1Ø     SCWCRP, SCWCR                    /CWCR
Ø075  4Ø14     SLCARP, SLCAR                    /LCAR
Ø076  4Ø2Ø     SCCARP, SCCAR                    /CCAR
Ø077  4Ø24     SLCMRP, SLCMR                    /LCMR
Ø1ØØ  4Ø3Ø     SLFGRP, SLFGR                    /LFGR
Ø1Ø1  4Ø34     SLDBRP, SLDBR                    /LDBR
Ø1Ø2  4Ø4Ø     S671ØP, S671Ø                    /READ IOT,
Ø1Ø3  4Ø44     SRWCRP, SRWCR                    /RWCR
Ø1Ø4  4Ø5Ø     SRCARP, SRCAR                    /RCAR
Ø1Ø5  4Ø54     SRMSRP, SRMSR                    /RMSR
Ø1Ø6  4Ø6Ø     SRCMRP, SRCMR                    /RCMR
Ø1Ø7  4Ø64     SRFSRP, SRFSR                    /RFSR
Ø11Ø  4Ø7Ø     SRDBRP, SRDBR                    /RDBR
Ø111  4Ø74     S672ØP, S672Ø                    /CONTROL IOT,
Ø112  41ØØ     SSKEFP, SSKEF                    /SKEF
Ø113  41Ø5     SSKCBP, SSKCB                    /SKCB
Ø114  4112     SSKTDP, SSKTD                    /SKTD
Ø115  4117     SSKTRP, SSKTR                    /SKTR
Ø116  4124     SCLFP,  SCLF                     /CLF
Ø117  413Ø     SCLTP,  SCLT                     /CLT
Ø12Ø  4134     SSDLEP, SSDLE                    /SDLE
Ø121  414Ø     SSBRMP, SSBRM                    /SBRM

               /PSEUDO MNEMONIC SUBROUTINE POINTERS,
Ø122  6547     SRBSWP, SRBSW                    /BSW
Ø123  64ØØ     CLR1P,  CLR1                     /CLEAR1
Ø124  64Ø6     CLR2P,  CLR2                     /CLEAR2
Ø125  6414     CLR3P,  CLR3                     /CLEAR3
Ø126  61Ø5     LAS5P,  LAS5                     /LOOP5
Ø127  6113     LAS6P,  LAS6                     /LOOP6
Ø13Ø  6121     LAS7P,  LAS7                     /LOOP7
Ø131  6127     LAS8P,  LAS8                     /LOOP8
Ø132  6135     LAS9P,  LAS9                     /LOOP9
Ø133  56ØØ     COMPP,  COMP                     /COMPAR
Ø134  5ØØØ     ERRORP, ERRORS                   /ERROR
Ø135  6534     DATBRP, DATBRK                   /BREAK
Ø136  6525     CMDP,   CMD                      /COMAND
Ø137  6432     OFFLNP, OFFLNE                   /OFFLIN
Ø14Ø  6436     REWNP,  REWND                    /REWIND
Ø141  6442     READDP, READD                    /READ
Ø142  6446     RDCMPP, RDCMP                    /RDCOMP
   3  6452     WRITP,  WRIT                     /WRITE
```

```
      Ø144  6456    EOFP,   EOFW                   /WEOF
      Ø145  6462    SPFORP, SPFORD                 /SPCFWD
      Ø146  6466    SPREVP, SPREV                  /SPCREV
      Ø147  6472    XGAPP,  XGAP                   /EXTGAP
      Ø15Ø  65ØØ    GOEP,   GOE                    /GO
      Ø151  Ø565    SIOT1P, SIOT1                  /LXXR,CXXR
      Ø152  Ø57Ø    SIOT2P, SIOT2                  /RXXR
      Ø153  6541    WCOVEP, WCOVER                 /WCOV

                    /WIDE USEAGE TEMPORARIES.
      Ø154  ØØØØ    GOOD,   Ø                      /GOOD DATA.
      Ø155  ØØØØ    BAD,    Ø                      /BAD DATA
      Ø156  ØØØØ    OLD,    Ø                      /OLD DATA.
      Ø157  ØØØØ    TXXTM1, Ø                      /THREE TEMPORARIES FOR
      Ø16Ø  ØØØØ    TXXTM2, Ø                      /FORMAL TEST USEAGE ONLY.
      Ø161  ØØØØ    TXXTM3, Ø
      Ø162  ØØØØ    SAVEAC, Ø                      /AC SAVED.
      Ø163  ØØØØ    SAVEL,  Ø                      /LINK SAVED.
      Ø164  ØØØØ    INTEF,  Ø                      /EF INTERRUPT SERVICE ADDRESS.
      Ø165  ØØØØ    PRGFLD, Ø                      /MEMORY FIELD WHICH PROGRAM OCCUPIES.
      Ø166  ØØØØ    INTMTF, Ø                      /MTTF INTERRUPT SERVICE ADDRESS.
      Ø167  ØØØØ    MEMFLD, Ø                      /NUMBER OF EXTENDED MEM FIELDS PRESENT.
      Ø17Ø  ØØØØ    ACLOC,  Ø                      /CONTENTS OF AC FOR ILLEGAL SKIP.

                    /WIDE USEAGE SOFTWARE FLAGS AND INDICATORS.
      Ø171  ØØØØ    TRACE,  Ø                      /SET IF "TR" PENDING.
      Ø172  ØØØØ    INTFLG, Ø                      /SET IF INTERRUPT SHOULD BE TURNED ON
                                                   /AFTER INTERRUPT SERVICE.
      Ø173  ØØØØ    TTOFLG, Ø                      /SET IF TTO FLAG GETS SET.
      Ø174  ØØØØ    EXITFL, Ø                      /SET IF AUTO-EXIT "EX-".
      Ø175  ØØØØ    ACTFLG, Ø                      /SET IF TEST IN PROGRESS.
      Ø176  ØØØØ    TSTAT,  Ø                      /BIT N SET MEANS RUN TEST
                                                   /N+1 (IN OCTAL).
      Ø177  ØØØØ    TSTNUM, Ø                      /CURRENT TEST NUMBER.


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

            Ø2ØØ    *2ØØ

      Ø2ØØ  5455    START,  JMP I   EXECP          /START HERE, GO TO EXEC.
      Ø2Ø1  712Ø            STL
      Ø2Ø2  5455            JMP I   EXECP
```

```
/TEST SECTION,

/THE TEST SECTION IS COMPOSED OF 12 TESTS (T01-T14) EACH
/OF WHICH IS MADE UP OF SUBTESTS DESIGNATED BY THE LETTERS A-Z, THEN 1-9,

/THE SUBTEST FORMAT IS DESCRIBED BELOW,

/SUBTEST FORMAT FOR NON-DATA TESTS:

/T01Z,  INST    /HOUSEKEEP TO SET UP LOOPS AND SPECIAL LOCATIONS
/       INST    /SUCH AS "GOOD" AND "OLD",
/       RUN TEST
/       AT THE END OF THE TEST, AC CONTAINS SOME PREDETERMINED VALUE,
/       COMPAR  /COMPARE THE VALUE IN AC WITH THE VALUE IN "GOOD",
                /IF AC="GOOD" CHECK FOR SUBTEST LOOP,
                /IF AC NOT EQUAL TO "GOOD" GO TO "ERRORS", TAKE
                /THE REQUIRED ACTION, THEN CHECK FOR SUBTEST LOOP,
/       ER01Z   /ADDRESS OF ERROR STATUS WORDS,
/       JMP     T01Z    /SUBTEST LOOP JUMP-REENTER HERE IF
                        /SUBTEST LOOP REQUESTED,
/       ETC,            /POINT OF CONTINUATION,


/THE SUBTEST FORMAT FOR DATA TESTS HAS ONE ADDITIONAL
/INSTRUCTION FOLLOWING THE SUBTEST LOOP JUMP:

/       JMP     T01Z    /SUBTEST LOOP JUMP,
/       JMP     TAG     /THIS INSTRUCTION RETURNS TO USE NEW DATA,
/IF AN "EX←" COMMAND IS PENDING AND AN ERROR OCCURS, THE ERROR
/ROUTINE ("ERRORS") WILL SKIP THIS INSTRUCTION THEREBY INHIBITING ANY
/MORE DATA FROM BEING USED IN THIS SUBTEST,   THIS ENABLES THE USER
/TO ESCAPE FROM A FAILING DATA TEST AND PROCEED TO THE NEXT SUBTEST,
                        /TEST Ø1, 67ØX, 671X IOT EXISTENCE AND BASIC NON-STATUS REGISTERS
                                /TEST.

                                /1.  VERIFIES THAT 67ØX AND 671X IOT'S EXIST EXCEPT FOR
                                /RMSR AND THAT WC, CA, CM, DB AND FRØ-4 CAN BE
                                /LOADED AND READ WITH ALL 1'S AND ALL Ø'S.  THE FACT
                                /THAT EACH OF THESE IOT'S FUNCTION ONLY WITH THEIR
                                /RELATED REGISTERS AND NOT WITH ANOTHER REGISTER IS
                                /ALSO VERIFIED.

                                /2.  TESTS TØ1D-TØ1P ARE GENERALIZED AND ARE CONTROLLED BY
                                /A LOCAL EXECUTIVE WHICH SELECTS THE IOT'S TO USE.

                                /3.  LOOPS 6-9:
                                /3.1  LOOP 6 CYCLES ON TØ1A-TØ1C (AC CLEARING)
                                /3.2  LOOP 7 CYCLES ON TØ1D-TØ1N (IOT EXISTENCE & BASIC REGISTER)
                                /3.3  LOOP 8 CYCLES ON TØ1O-TØ1P (IOT UNIQUENESS)


        Ø2Ø3  ØØØØ      TESTØ1, Ø
        Ø2Ø4  7ØØ1              IAC
        Ø2Ø5  3177              DCA     TSTNUM
        Ø2Ø6  6ØØ2              IOF                     /RUN WITH INTERRUPT SYSTEM OFF.
        Ø2Ø7  3172              DCA     INTFLG
        Ø21Ø  3154              DCA     GOOD

                        /VERIFY 67ØØ CLEARS AC.
        Ø211  724Ø      TØ1A,   CLA CMA
        Ø212  4472              I67ØØ
        Ø213  4533              COMPAR
        Ø214  67Ø5              ERØ1A
                        /IOT 67ØØ DID NOT CLEAR AC,  GD=GOOD AC; BD=REAL AC,
        Ø215  5211              JMP     TØ1A            /SUBTEST LOOP,

                        /VERIFY 671Ø CLEARS AC.
        Ø216  724Ø      TØ1B,   CLA CMA
        Ø217  45Ø2              I671Ø
        Ø22Ø  4533              COMPAR
        Ø221  67Ø7              ERØ1B
                        /IOT 671Ø DID NOT CLEAR AC,  GD=GOOD AC; BD=REAL AC
        Ø222  5216              JMP     TØ1B            /SUBTEST LOOP,

                        /VERIFY 672Ø DOES NOT CLEAR AC
        Ø223  724Ø      TØ1C,   CLA CMA
        Ø224  3154              DCA     GOOD
        Ø225  724Ø              CLA CMA
        Ø226  4511              I672Ø
        Ø227  4533              COMPAR
        Ø23Ø  6711              ERØ1C
                        /IOT 672Ø CLEARED AC, GD=GOOD AC, BD=REAL AC,
        Ø231  5225              JMP     TØ1C+2          /SUBTEST LOOP

        Ø232  4527              LOOP6
        Ø233  521Ø              JMP     TØ1A-1          /*****LOOP 6*****
        Ø234  4775              JMS I   T1EX1P          /TO EXECUTIVE.

                        /VERIFY LOAD IOT LXXR CLEARS AC.
        Ø235  3154      TØ1D,   DCA     GOOD
        Ø236  1157              TAD     TXXTM1
        Ø237  4551              LXXR
        Ø24Ø  4533              COMPAR
        Ø241  6713              ERØ1D
                        /LOAD IOT LXXR DID NOT CLEAR AC.  GD=GOOD AC; BD=REAL AC;
                        /I1=LXXR.
        Ø242  5236              JMP     TØ1D+1          /SUBTEST LOOP.


                        /VERIFY THAT LXXR WITH AC=7777, THEN RXXR RESULTS IN A
                        /NON-ZERO AC.
        Ø243  1157      TØ1E,   TAD     TXXTM1
        Ø244  4551              LXXR
        Ø245  72ØØ              CLA                     /CLEAR AC TO ENSURE GOOD TEST.
        Ø246  4552              RXXR
        Ø247  Ø157              AND     TXXTM1          /MASK BITS Ø-4 FOR FS ONLY.
        Ø25Ø  765Ø              SNA CLA
        Ø251  7Ø4Ø              CMA
        Ø252  4533              COMPAR
        Ø253  6715              ERØ1E
                        /LXXR WITH AC=7777, THEN RXXR DID NOT RESULT IN A NON-ZERO
                        /AC.  I1=LXXR; I2=RXXR
        Ø254  5243              JMP     TØ1E            /SUBTEST LOOP

                        /VERIFY XX REGISTER TO BE LOADED AND READ ALL 1'S BY LXXR
                        /THEN RXXR.
        Ø255  724Ø      TØ1F,   CLA CMA
        Ø256  3154              DCA     GOOD
        Ø257  1157              TAD     TXXTM1
        Ø26Ø  4551              LXXR
        Ø261  72ØØ              CLA                     /AC CLEARED TO ENSURE GOOD TEST.
        Ø262  4552              RXXR
        Ø263  Ø157              AND     TXXTM1          /MASK TO BITS Ø-4 FOR FS ONLY.
        Ø264  116Ø              TAD     TXXTM2          /ADD IN COMPLEMENT MASK.
        Ø265  4533              COMPAR
        Ø266  6717              ERØ1F
                        /XX REGISTER NOT LOADED AND/OR READ WITH ALL 1'S USING
                        /LXXR AND RXXR.  GD=GOOD AC; BD=REAL AC; I1=LXXR;
                        /I2=RXXR.
        Ø267  5257              JMP     TØ1F+2          /SUBTEST LOOP

                        /VERIFY THAT LXXR WITH AC=ØØØØ, THEN RXXR RESULTS IN
                        /AT LEAST ONE BIT CLEAR IN AC.
        Ø27Ø  3154      TØ1G,   DCA     GOOD
        Ø271  4551              LXXR
        Ø272  72ØØ              CLA
        Ø273  4552              RXXR
        Ø274  Ø157              AND     TXXTM1          /MASK BITS Ø-4 FOR FS ONLY.
        Ø275  116Ø              TAD     TXXTM2          /ADD IN COMPLEMENT MASK.
        Ø276  7ØØ1              IAC                     /ADD 1 TO CHECK FLIP TO Ø.
          77  765Ø              SNA CLA
        0300  7040              CMA
        0301  4533              COMPAR
        0302  6721              ER01G
                        /LXXR WITH AC=0000, THEN RXXR DID NOT RESULT IN AT
                        /LEAST ONE BIT CLEAR IN AC,  I1=LXXR; I2=RXXR,
        0303  5271              JMP     T01G+1          /SUBTEST LOOP


                        /VERIFY XX REGISTER CAN BE LOADED AND READ ALL 0'S,
        0304  4551      T01H,   LXXR
        0305  7200              CLA
        0306  4552              RXXR
        0307  0157              AND     TXXTM1          /MASK BITS 0-4 FOR FS ONLY
        0310  4533              COMPAR
        0311  6723              ER01H
                        /LXXR WITH AC=0000, THEN RXXR DID NOT RESULT IN A ZERO AC,
                        /GD=GOOD AC; BD=REAL AC; I1=LXXR; I2=RXXR,
        0312  5304              JMP     T01H            /SUBTEST LOOP,

                        /VERIFY CLT CLEARS THE XX REGISTER WHICH IS PRELOADED TO 1'S
                        /BY LXXR,  LDBR WITH AC=0000 USED FOR DB,
        0313  1157      T01I,   TAD     TXXTM1
        0314  4551              LXXR
        0315  4525              CLEAR3
        0316  4552              RXXR
        0317  0157              AND     TXXTM1          /MASK BITS 0-4 FOR FS ONLY
        0320  4533              COMPAR
        0321  6725              ER01I
                        /XX REGISTER NOT CLEARED BY CLT (OR LDBR IF DB),  GD=GOOD AC; BD=REAL AC;
                        /I1=LXXR; I2=RXXR,
        0322  5313              JMP     T01I            /SUBTEST LOOP,

                        /VERIFY RXXR CLEARS AC,  REGISTER PRELOADED TO 0000,
        0323  4525      T01J,   CLEAR3
        0324  7240              CLA CMA
        0325  4552              RXXR
        0326  0157              AND     TXXTM1          /MASK BITS 0-4 FOR FS ONLY
        0327  4533              COMPAR
        0330  6727              ER01J
                        /RXXR DOES NOT CLEAR AC,  GD=GOOD AC; BD=REAL AC; I2=RXXR,
        0331  5323              JMP     T01J            /SUBTEST LOOP

        0332  1157              TAD     TXXTM1          /PRELOAD REGISTER TO ALL 1'S
        0333  4551              LXXR                    /FOR CXXR CHECKS
        0334  7200              CLA                     /ONLY TEST REGISTERS FOR WHICH
        0335  4776              JMS I   T1EX2P          /CXXR APPLIES,


                        /VERIFY CXXR CLEARS AT LEAST ONE BIT IN THE REGISTER WHICH IS
                        /PRELOADED TO ALL 1'S,
        0336  7200      T01K,   CLA
        0337  4551              CXXR
        0340  4552              RXXR
        0341  7001              IAC
        0342  7650              SNA CLA
```

```
      Ø343  7Ø4Ø              CMA
      Ø344  4533              COMPAR
      Ø345  6731              ERØ1K
                        /CXXR DID NOT CLEAR AT LEAST ONE BIT IN XX REGISTER
                        /I1=CXXR; I2=RXXR
      Ø346  5337              JMP     TØ1K+1          /SUBTEST LOOP

                        /VERIFY THAT CXXR CLEARS THE XX REGISTER
      Ø347  72ØØ        TØ1L,   CLA
      Ø35Ø  4551              CXXR
      Ø351  4552              RXXR
      Ø352  4533              COMPAR
      Ø353  6733              ERØ1L
                        /CXXR DID NOT CLEAR XX REGISTER,  GD=GOOD AC; BD=REAL AC;
                        /I1=CXXR; I2=RXXR,
      Ø354  535Ø              JMP     TØ1L+1          /SUBTEST LOOP

                        /VERIFY CXXR CLEARS AC.
      Ø355  724Ø        TØ1M,   CLA CMA
      Ø356  4551              CXXR
      Ø357  4533              COMPAR
      Ø36Ø  6735              ERØ1M
                        /CXXR DOES NOT CLEAR AC.  GD=GOOD AC; BD=REAL AC;
                        /I1=CXXR,
      Ø361  5355              JMP     TØ1M            /SUBTEST LOOP

                        /VERIFY CXXR WITH AC=7777 DOES NOT SET ANY BIT IN XX
                        /REGISTER,
      Ø362  724Ø        TØ1N,   CLA CMA
      Ø363  4551              CXXR
      Ø364  4552              RXXR
      Ø365  4533              COMPAR
      Ø366  6737              ERØ1N
                        /CXXR SET AT LEAST ONE BIT IN XX REGISTER,  GD=GOOD AC;
                        /BD=REAL AC; I1=CXXR; I2=RXXR,
      Ø367  5362              JMP     TØ1N            /SUBTEST LOOP

      Ø37Ø  4777              JMS I   T1EX3P          /BACK TO EXECUTIVE.


      Ø371  453Ø        T1LP7,  LOOP7
      Ø372  5232              JMP     TØ1D-3          /*****LOOP 7*****

      Ø373  5774              JMP I   .+1             /GO TO TØ1Ø WHEN D-N DONE
      Ø374  Ø4ØØ              TØ1ØL
      Ø375  Ø45Ø        T1EX1P, T1EX1
      Ø376  Ø473        T1EX2P, T1EX2
      Ø377  Ø5Ø1        T1EX3P, T1EX3

            Ø4ØØ              PAGE


      Ø4ØØ  43Ø5        TØ1ØL,  JMS     T1EX4           /TO EXECUTIVE TO INSERT
      Ø4Ø1  4525              CLEAR3                  /IOT'S,
      Ø4Ø2  3154              DCA     GOOD
```

```
                    /VERIFY SPECIFIC LOAD, CLEAR, AND READ IOT'S AFFECT ONLY THEIR
                    /RELATED REGISTERS.  AC SET TO 7777 PRIOR TO LOAD OR CLEAR
                    /WITH ALL OTHER REGISTERS CLEAR.  IOT'S INSERTED BY EXECUTIVE.
Ø4Ø3  724Ø          TØ1Ø,   CLA CMA
Ø4Ø4  4551                  LXXR
Ø4Ø5  4552                  RXXR
Ø4Ø6  Ø157                  AND     TXXTM1
Ø4Ø7  4533                  COMPAR
Ø41Ø  6741                  ERØ1Ø
                    /IOT LXXR AFFECTS REGISTER READ BY RXXR, OR RXXR READS
                    /REGISTER LOADED BY LXXR.  GD=GOOD AC; BD=REAL AC;
                    /I1=LXXR; I2=RXXR.
Ø411  52Ø3                  JMP     TØ1Ø            /SUBTEST LOOP

                    /VERIFY THAT SPECIFIC LOAD, CLEAR AND READ IOT'S AFFECT
                    /ONLY THEIR RELATED REGISTERS.  AC SET TO ØØØØ PRIOR TO
                    /LOAD OR CLEAR WITH ALL OTHER REGISTERS SET TO 1'S.
                    /SAME METHOD AS TØ1Ø.
Ø412  4525                  CLEAR3
Ø413  724Ø                  CLA CMA                 /LOAD ALL REGISTERS WITH 1'S.
Ø414  3154                  DCA     GOOD
Ø415  724Ø                  CLA CMA
Ø416  4473                  LWCR
Ø417  724Ø                  CLA CMA
Ø42Ø  4475                  LCAR
Ø421  724Ø                  CLA CMA
Ø422  4477                  LCMR
Ø423  724Ø                  CLA CMA
Ø424  45Ø1                  LDBR
Ø425  1374                  TAD     K76ØØ
Ø426  45ØØ                  LFGR
Ø427  72ØØ          TØ1P,   CLA
Ø43Ø  4551                  LXXR                    /LOAD ONE REGISTER
Ø431  4552                  RXXR                    /READ ANOTHER
Ø432  Ø157                  AND     TXXTM1          /MASK BITS Ø-4 FOR FS ONLY.
Ø433  116Ø                  TAD     TXXTM2          /ADD COMPLEMENT MASK.
Ø434  4533                  COMPAR
Ø435  6743                  ERØ1P
                    /IOT LXXR AFFECTS REGISTER READ BY RXXR, OR RXXR READS
                    /REGISTER LOADED BY LXXR.  GD=GOOD AC; BD=REAL AC;
                    /I1=LXXR; I2=RXXR.
Ø436  523Ø                  JMP     TØ1P+1          /SUBTEST LOOP

Ø437  434Ø                  JMS     T1EX5           /TO EXECUTIVE

Ø44Ø  4531                  LOOP8
Ø441  5773                  JMP I   T1LP7P          /*****LOOP 8*****

Ø442  4523                  CLEAR1                  /CLEAR REGISTERS - ION.
Ø443  1647                  TAD I   TST1P           /EXIT.
Ø444  3246                  DCA     .+2
Ø445  5646                  JMP I   .+1
Ø446  ØØØØ                  Ø
Ø447  Ø2Ø3          TST1P,  TESTØ1
```

```
                    /TEST Ø1 LOCAL EXECUTIVE TO SERVICE TESTS TØ1D-TØ1N,

        Ø45Ø  ØØØØ  T1EX1,  Ø                   /ENTER PRIOR TO TØ1D,
        Ø451  1356          TAD     T1LS1P      /SET UP LIST POINTERS
        Ø452  3Ø1Ø          DCA     1Ø
        Ø453  1357          TAD     T1LS2P
        Ø454  3Ø11          DCA     11
        Ø455  136Ø          TAD     T1LS3P
        Ø456  3Ø12          DCA     12
        Ø457  1Ø41          TAD     M5
        Ø46Ø  3361          DCA     T1EXT1
        Ø461  141Ø  T1EX1A, TAD I   1Ø          /SET IOT1 FOR LXXR,
        Ø462  3366          DCA     IOT1
        Ø463  1412          TAD I   12          /SET IOT2 FOR RXXR,
        Ø464  3371          DCA     IOT2
        Ø465  1412          TAD I   12          /SET MASK AND COMPLEMENT
        Ø466  3157          DCA     TXXTM1      /MASK,
        Ø467  1157          TAD     TXXTM1
        Ø47Ø  7Ø4Ø          CMA
        Ø471  316Ø          DCA     TXXTM2
        Ø472  565Ø          JMP I   T1EX1       /GO BACK AND RUN TØ1D-TØ1J,
        Ø473  ØØØØ  T1EX2,  Ø                   /ENTER AFTER TØ1J,
        Ø474  1411          TAD I   11          /SEE IF CXXR APPLIES,
        Ø475  745Ø          SNA
        Ø476  53Ø2          JMP     .+4
        Ø477  3366          DCA     IOT1        /CXXR APPLIES,  PUT IN IOT1,
        Ø5ØØ  5673          JMP I   T1EX2       /GO BACK AND RUN TØ1K-TØ1N,
        Ø5Ø1  ØØØØ  T1EX3,  Ø                   /ENTER AFTER TØ1N,
        Ø5Ø2  2361          ISZ     T1EXT1      /ALL IOT'S CHECKED?
        Ø5Ø3  5261          JMP     T1EX1A      /NO,  RESET IOT'S
        Ø5Ø4  57Ø1          JMP I   T1EX3       /YES,  EXIT TO END OF TØ1N,

                    /TEST Ø1 LOCAL EXECUTIVE TO SERVICE TESTS TØ1O-TØ1P,

        Ø5Ø5  ØØØØ  T1EX4,  Ø
        Ø5Ø6  1356          TAD     T1LS1P      /SET LIST POINTERS,
        Ø5Ø7  3Ø1Ø          DCA     1Ø
        Ø51Ø  1357          TAD     T1LS2P
        Ø511  3Ø11          DCA     11
        Ø512  136Ø          TAD     T1LS3P
        Ø513  3361          DCA     T1EXT1
        Ø514  1Ø41          TAD     M5          /SET FOR 5 LOAD, CLEAR COMBINATIONS,
        Ø515  3362          DCA     T1EXT2

        Ø516  141Ø  T1EX7,  TAD I   1Ø          /GET LOAD,
        Ø517  3366          DCA     IOT1        /PUT IN IOT1,
        Ø52Ø  1Ø37          TAD     M2          /SET FOR PAIR COUNTER,
        Ø521  3364          DCA     T1EXT4
        Ø522  2361          ISZ     T1EXT1      /UPDATE READ POINTER,
        Ø523  2361          ISZ     T1EXT1
        Ø524  1361  T1EX6,  TAD     T1EXT1      /GET READ POINTER AND
        Ø525  3Ø12          DCA     12          /STORE IN AUTO-INDEX,
          26  1Ø4Ø          TAD     M4          /SET FOR 4 READS,
Ø527  3363            DCA     T1EXT3
Ø53Ø  1412    T1EX8,  TAD I   12              /GET READ.
Ø531  3371            DCA     IOT2            /PUT IN IOT2.
Ø532  1412            TAD I   12              /GET MASK.  SAVE MASK
Ø533  3157            DCA     TXXTM1          /AND COMPLEMENT MASK.
Ø534  1157            TAD     TXXTM1
Ø535  7Ø4Ø            CMA
Ø536  316Ø            DCA     TXXTM2
Ø537  57Ø5            JMP I   T1EX4           /RUN TEST.
Ø54Ø  ØØØØ    T1EX5,  Ø
Ø541  2363            ISZ     T1EXT3          /ENTER AFTER TEST.  4 READS?
Ø542  533Ø            JMP     T1EX8           /NO.  DO NEXT READ.
Ø543  2364            ISZ     T1EXT4          /YES.  DONE WITH THIS LOAD, CLEAR PAIR?
Ø544  741Ø            SKP
Ø545  5353            JMP     .+6             /YES.
Ø546  1411            TAD I   11              /NO.  GET CLEAR.
Ø547  745Ø            SNA
Ø55Ø  5353            JMP     .+3
Ø551  3366            DCA     IOT1            /CLEAR APPLIES.  RUN TEST AFTER
Ø552  5324            JMP     T1EX6           /SETTING UP READS.
Ø553  2362            ISZ     T1EXT2          /ALL DONE?
Ø554  5316            JMP     T1EX7           /NO.  RUN NEXT PAIR.
Ø555  574Ø            JMP I   T1EX5           /YES.  EXIT.
Ø556  6612    T1LS1P, T1LS1-1
Ø557  6617    T1LS2P, T1LS2-1
Ø56Ø  6624    T1LS3P, T1LS3-1
Ø561  ØØØØ    T1EXT1, Ø
Ø562  ØØØØ    T1EXT2, Ø
Ø563  ØØØØ    T1EXT3, Ø
Ø564  ØØØØ    T1EXT4, Ø
Ø565  ØØØØ    SIOT1,  Ø                       /IOT1 SUBROUTINE.
Ø566  ØØØØ    IOT1,   Ø
Ø567  5765            JMP I   SIOT1
Ø57Ø  ØØØØ    SIOT2,  Ø                       /IOT2 SUBROUTINE.
Ø571  ØØØØ    IOT2,   Ø
Ø572  577Ø            JMP I   SIOT2
Ø573  Ø371    T1LP7P, T1LP7
Ø574  76ØØ    K76ØØ,  76ØØ
```

```
                    /TEST Ø2.  NON-STATUS REGISTER DATA TEST.

                            /1.  VERIFIES THAT WC, CA, CM, DB AND FRØ-4 CAN BE LOADED
                            /AND READ USING INCREMENTED COMPLEMENT DATA.

                            /2.  ALL SUBTEST LOOPS CYCLE ON CONSTANT DATA.

                            /3.  LOOPS 6-9:
                            /3.1  LOOP 6:  TØ2A(WC)
                            /3.2  LOOP 7:  TØ2B(CA)
                            /3.3  LOOP 8:  TØ2C(CM)
                            /3.4  LOOP 9:  TØ2D-TØ2E(DB,FRØ-4)


        Ø6ØØ                PAGE
  Ø6ØØ  ØØØØ        TESTØ2, Ø
  Ø6Ø1  4523                CLEAR1                  /CLEAR ALL-ION.

                    /VERIFY WC CAN BE LOADED AND READ USING INCREMENTAL
                    /COMPLEMENTED PATTERN.
  Ø6Ø2  43Ø7        TØ2A,   JMS     TØ2LDS          /GENERATE DATA.
  Ø6Ø3  7777                7777                    /MASK TO CHECK ALL BITS.
  Ø6Ø4  5215                JMP     TØ2B-2          /DONE.
  Ø6Ø5  1157                TAD     TXXTM1          /GET DATA
  Ø6Ø6  4473                LWCR                    /LOAD WC.
  Ø6Ø7  72ØØ                CLA                     /CLEAR AC TO ENSURE GOOD TEST.
  Ø61Ø  45Ø3                RWCR                    /ROAD WC.
  Ø611  4533                COMPAR
  Ø612  6745                ERØ2A
                    /WC NOT LOADED OR READ CORRECTLY. GD=GOOD WC; OD=PREVIOUS
                    /GOOD WC. SUBTEST LOOP CYCLES ON PRESENT DATA.
  Ø613  52Ø5                JMP     .-6             /SUBTEST LOOP
  Ø614  5331                JMP     TØ2LDR          /GET MORE DATA.

  Ø615  4527                LOOP6
  Ø616  52Ø2                JMP     TØ2A            /*****LOOP 6*****

                    /VERIFY CA CAN BE LOADED AND READ USING INCREMENTAL
                    /COMPLEMENTED DATA.
  Ø617  43Ø7        TØ2B,   JMS     TØ2LDS
  Ø62Ø  7777                7777
  Ø621  5232                JMP     TØ2C-2
  Ø622  1157                TAD     TXXTM1
  Ø623  4475                LCAR
  Ø624  72ØØ                CLA                     /CLEAR AC TO ENSURE GOOD TEST.
  Ø625  45Ø4                RCAR
  Ø626  4533                COMPAR
  Ø627  6747                ERØ2B
                    /CA NOT LOADED OR READ CORRECTLY.  GD=GOOD CA; OD=PREVIOUS
                    /GOOD CA.  SUBTEST LOOP CYCLES ON PRESENT DATA.
  Ø63Ø  5222                JMP     .-6             /SUBTEST LOOP.
  Ø631  5331                JMP     TØ2LDR
```

```
        Ø632  453Ø              LOOP7
        Ø633  5215              JMP     TØ2B-2          /*****LOOP 7*****

                        /VERIFY CM CAN BE LOADED AND READ USING INCREMENTAL
                        /COMPLEMENTED DATA.
        Ø634  3172      TØ2C,   DCA     INTFLG          /ALLOW NO INTERRUPTS.
        Ø635  6ØØ2              IOF
        Ø636  43Ø7              JMS     TØ2LDS
        Ø637  7777              7777
        Ø64Ø  5251              JMP     TØ2D-3
        Ø641  1157              TAD     TXXTM1
        Ø642  4477              LCMR
        Ø643  72ØØ              CLA                     /CLEAR AC TO INSURE GOOD TEST.
        Ø644  45Ø6              RCMR
        Ø645  4533              COMPAR
        Ø646  6751              ERØ2C
                        /CM NOT LOADED OR READ CORRECTLY, GD=GOOD CM; OD=PREVIOUS
                        /GOOD CM, SUBTEST LOOP CYCLES ON PRESENT DATA.
        Ø647  5241              JMP     .-6             /SUBTEST LOOP
        Ø65Ø  5331              JMP     TØ2LDR

        Ø651  4523              CLEAR1                  /CLEAR ALL - ION.
        Ø652  4531              LOOP8
        Ø653  5232              JMP     TØ2C-2          /*****LOOP 8*****

                        /VERIFY DB CAN BE LOADED AND READ USING INCREMENTAL
                        /COMPLEMENTED DATA.
        Ø654  4524      TØ2D,   CLEAR2
        Ø655  43Ø7              JMS     TØ2LDS
        Ø656  7777              7777
        Ø657  527Ø              JMP     TØ2E
        Ø66Ø  1157              TAD     TXXTM1
        Ø661  45Ø1              LDBR
        Ø662  72ØØ              CLA                     /CLEAR AC TO ENSURE GOOD TEST.
        Ø663  451Ø              RDBR
        Ø664  4533              COMPAR
        Ø665  6753              ERØ2D
                        /DB NOT LOADED OR READ CORRECTLY, GD=GOOD DB; OD=PREVIOUS
                        /GOOD DB, SUBTEST LOOP CYCLES ON PRESENT DATA.
        Ø666  526Ø              JMP     .-6             /SUBTEST LOOP
        Ø667  5331              JMP     TØ2LDR

                        /VERIFY FS Ø-4 CAN BE LOADED AND READ USING INCREMENTAL
                        /COMPLEMENTED DATA.
        Ø67Ø  43Ø7      TØ2E,   JMS     TØ2LDS
        Ø671  76ØØ              76ØØ                    /MASK FOR BITS Ø-4
        Ø672  53Ø4              JMP     TØ2LP9
        Ø673  1157              TAD     TXXTM1
        Ø674  45ØØ              LFGR
        Ø675  72ØØ              CLA
        Ø676  45Ø7              RFSR
        Ø677  Ø271              AND     TØ2E+1          /MASK
        Ø7ØØ  4533              COMPAR
        Ø7Ø1  6755              ERØ2E
                        /FS Ø-4 NOT LOADED OR READ CORRECTLY, GD=GOOD FS Ø-4;
                        /BD=REAL FS Ø-4 (MASKED); OD=PREVIOUS GOOD FS Ø-4,
     Ø7Ø2  5273                 JMP     .-7                 /SUBTEST LOOP
     Ø7Ø3  5331                 JMP     TØ2LDR

     Ø7Ø4  4532         TØ2LP9, LOOP9
     Ø7Ø5  5252                 JMP     TØ2D-2              /*****LOOP 9*****

     Ø7Ø6  56ØØ                 JMP I   TESTØ2              /EXIT,


                        /DATA SUPPLIER FOR WC, CA, CM, DB, FR.
                                /STARTING FROM Ø OUTPUTS THE FOLLOWING INCREMENTAL COMPLEMENTED
                                /DATA PATTERN,
                                        /Ø:  7777,7777,Ø,Ø,7777,7777,Ø,Ø
                                        /1:  7776,7776,1,1,7776,7776,1,1, ETC,
                                /THE CALLING SEQUENCE IS:
                                /JMS    TØ2LDS
                                /DATA MASK CONSTANT
                                /EXIT
                                /TEST INSTRUCTIONS
                                /ERROR CHECK
                                /JMP TØ2LDR TO GET NEW DATA,

     Ø7Ø7  ØØØØ         TØ2LDS, Ø                           /ENTER,
     Ø71Ø  3157                 DCA     TXXTM1              /Ø FOR A STARTER,
     Ø711  1Ø21                 TAD     K2                  /COMPUTE RETURN EXIT
     Ø712  13Ø7                 TAD     TØ2LDS              /AND SAVE,
     Ø713  316Ø                 DCA     TXXTM2
     Ø714  1Ø4Ø         TØ2LDU, TAD     M4                  /SET FOR FOUR COMPLEMENTS,
     Ø715  3161                 DCA     TXXTM3
     Ø716  1Ø37         TØ2LDT, TAD     M2                  /SET FOR TWO SENDS OF ONE PATTERN,
     Ø717  3345                 DCA     TØ2LDM
     Ø72Ø  1154                 TAD     GOOD                /PUT PREVIOUS DATA
     Ø721  3156                 DCA     OLD                 /IN OLD,
     Ø722  1157                 TAD     TXXTM1              /SET LAST DATA,
     Ø723  7Ø4Ø                 CMA                         /COMPLEMENT,
     Ø724  3157                 DCA     TXXTM1              /SAVE,
     Ø725  1157                 TAD     TXXTM1              /GET LAST DATA,
     Ø726  Ø7Ø7                 AND I   TØ2LDS              /MASK,
     Ø727  3154                 DCA     GOOD                /THIS IS WHAT WE SHOULD READ,
     Ø73Ø  556Ø                 JMP I   TXXTM2              /RUN THE TEST,
     Ø731  2345         TØ2LDR, ISZ     TØ2LDM              /THIS PATTERN SENT 2 TIMES,
     Ø732  741Ø                 SKP
     Ø733  5337                 JMP     .+4
     Ø734  1154                 TAD     GOOD                /YES,  PUT OLD DATA IN OLD,
     Ø735  3156                 DCA     OLD
     Ø736  556Ø                 JMP I   TXXTM2              /RUN PATTERN AGAIN,
     Ø737  2161                 ISZ     TXXTM3              /COMPLEMENTING DONE 4 TIMES?
     Ø74Ø  5316                 JMP     TØ2LDT              /NO,  SET UP NEXT DATA,
     Ø741  2157                 ISZ     TXXTM1              /YES,  INCREMENT DATA,
     Ø742  5314                 JMP     TØ2LDU              /SEND OUT NEXT DATA TRAIN,
     Ø743  23Ø7                 ISZ     TØ2LDS              /ALL DONE,  UPDATE RETURN,
     Ø744  57Ø7                 JMP I   TØ2LDS              /EXIT,
     Ø745  ØØØØ         TØ2LDM, Ø
```

```
                /TEST Ø3,  BASIC FUNCTIONS EXISTENCE TEST,

                        /1,  REFER TO SUBTESTS FOR SPECIFIC TEST INFORMATION,
                        /2,  TESTS TØ3D-TØ3I ARE GENERALIZED AND ARE CONTROLLED BY A
                /LOCAL EXECUTIVE,
                        /3,  LOOPS 6-9:
                        /3,1  LOOP 6 CYCLES ON TØ3A-TØ3C,
                        /3,2  LOOP 7 CYCLES ON TØ3D-TØ3E,
                        /3,3  LOOP 8 CYCLES ON TØ3F-TØ3G,
                        /3,4  LOOP 9 CYCLES ON TØ3H-TØ3I,
                        /4,  EACH SUBTEST LOOP CYCLES ON THE FUNCTION IN CURRENT USE,


      1ØØØ              PAGE
1ØØØ  ØØØØ      TESTØ3, Ø
1ØØ1  4523              CLEAR1                  /CLEAR ALL REGISTERS-ION
1ØØ2  1Ø47              TAD     XBUFFP
1ØØ3  3156              DCA     OLD

                /VERIFY WRITE-GO DOES NOT CAUSE A DATA BREAK WITH NO
                /TRANSOPRT READY, WC STARTS AT ØØØØ AND SHOULD
                /REMAIN ØØØØ,
1ØØ4  3154      TØ3A,   DCA     GOOD
1ØØ5  4524              CLEAR2                  /CLEAR REGISTERS
1ØØ6  4473              LWCR                    /Ø TO WC
1ØØ7  4543              WRITE
1Ø1Ø  455Ø              GO
1Ø11  45Ø3              RWCR
1Ø12  4533              COMPAR
1Ø13  6757              ERØ3A
                /WC CHANGED WHEN WRITE-GO ISSUED WITH NO TRANSPORT READY,
                /GD=GOOD WC,
1Ø14  52Ø5              JMP     TØ3A+1          /SUBTEST LOOP,

                /VERIFY WRITE CAUSES TRANSFER OUT OF MEMORY, CA REFERENCES
                /ADDRESSES XBUFF OR XBUFF+1, CM6-8 REFERENCE THE PROGRAM
                /FIELD, AND THAT DB CAN RECEIVE ALL 1'S, (XBUFF AND
                /XBUFF+1 CONTAIN 7777,)
1Ø15  724Ø      TØ3B,   CLA CMA
1Ø16  3154              DCA     GOOD
1Ø17  44Ø4              JMS I   T3LODP          /LOAD UP REGISTERS,
1Ø2Ø  45Ø1              LDBR                    /Ø TO DB
1Ø21  4543              WRITE
1Ø22  4535              BREAK                   /1 DATA BREAK
1Ø23  451Ø              RDBR
1Ø24  4533              COMPAR
1Ø25  6761              ERØ3B
                /GD=GOOD DB; OD=ADDRESS XBUFF (CA SHOULD EQUAL THIS
                /QUANTITY OR ONE MORE-CA INCREMENT IS NOT BEING
                /TESTED,)
1Ø26  5217              JMP     TØ3B+2          /SUBTEST LOOP,

                /VERIFY READ CAUSES TRANSFER INTO MEMORY, CA REFERENCES
                /ADDRESSES XBUFF OR XBUFF+1, AND THAT DB CAN SEND ALL
                /Ø'S. (XBUFF AND XBUFF+1 CONTAIN 7777 PRIOR TO
                /BREAK,)
1027  4404      TØ3C,   JMS I   T3LODP          /LOAD UP REGISTERS
1Ø3Ø  45Ø1              LDBR                    /Ø TO DB
1Ø31  4541              READ
1Ø32  4535              BREAK                   /1 DATA BREAK
1Ø33  1Ø5Ø              TAD     XBUFF           /SUM
1Ø34  1Ø51              TAD     XBUFF+1
1Ø35  4533              COMPAR
1Ø36  6763              ERØ3C
                /GD=GOOD SUM OF THE CONTENTS OF XBUFF AND XBUFF+1 (ONE OF
                /THE VALUES SHOULD BE 7777, THE OTHER ØØØØ YIELDING A
                /SUM OF 7777); BD=ACTUAL SUM; OD=ADDRESS XBUFF
                /(CA SHOULD EQUAL THIS QUANTITY OR 1 MORE = CA INCREMENT
                /NOT BEING TESTED.)
1Ø37  5227              JMP     TØ3C            /BSUBTEST LOOP

1Ø4Ø  4527              LOOP6
1Ø41  52Ø4              JMP     TØ3A            /*****LOOP6*****

                /VERIFY ALL FUNCTIONS EXCEPT READ DO NOT CAUSE TRANSFER
                /INTO PROCESSOR MEMORY. XBUFF AND XBUFF+1 CONTAIN
                /7777 PRIOR TO BREAK. DB CONTAINS ØØØØ.
1Ø42  3154      TØ3D,   DCA     GOOD
1Ø43  4776              JMS I   T3EX1P          /TO EXECUTIVE TO INSERT FUNCTION.
1Ø44  6647              T3LS1                   /FUNCTION LIST POINTER
1Ø45  7771              -7                      /NUMBER OF FUNCTIONS TO USE.
1Ø46  1Ø51              .+3                     /WHERE TO PUT FUNCTION.
1Ø47  44Ø4              JMS I   T3LODP          /LOAD ALL REGISTERS.
1Ø5Ø  45Ø1              LDBR                    /Ø TO DB.
1Ø51  ØØØØ              Ø                       /VARIABLE FUNCTION HERE.
1Ø52  4535              BREAK                   /1 DATA BREAK.
1Ø53  1Ø5Ø              TAD     XBUFF           /XBUFF AND XBUFF+1 SHOULD
1Ø54  7Ø41              CIA                     /BE EQUAL AND UNCHANGED.
1Ø55  1Ø51              TAD     XBUFF+1
1Ø56  4533              COMPAR
1Ø57  6765              ERØ3D
                /GD=GOOD DIFFERENCE OF THE CONTENTS OF XBUFF+1=XBUFF;
                /BD=ACTUAL DIFFERENCE; OD=ADDRESS XBUFF; FS=FUNCTION
                /WHICH FAILED. SUBTEST LOOP CYCLES ON CURRENT FUNCTION.
1Ø6Ø  5777              JMP I   T3EX32          /SUBTEST LOOP.
1Ø61  4775              JMS I   T3EX2P          /BACK FOR ANOTHER FUNCTION.

                /VERIFY THAT OFFLINE, REWIND, WEOF, SPCFWD, SPCREV AND READ
                /DO NOT CAUSE TRANSFER FROM PROCESSOR MEMORY TO DB.
                /XBUFF AND XBUFF+1 CONTAIN 7777 PRIOR TO BREAK, DB CONTAINS ØØØØ.
1Ø62  4776      TØ3E,   JMS I   T3EX1P          /TO EXECUTIVE TO GET FUNCTION.
1Ø63  6651              T3LS1+2                 /FUNCTION LIST POINTER.
1Ø64  7772              -6                      /# OF FUNCTIONS
1Ø65  1Ø7Ø              .+3                     /WHERE TO INSERT FUNCTION
1Ø66  44Ø4              JMS I   T3LODP          /LOAD REGISTERS


1Ø67  45Ø1              LDBR                    /Ø TO DB
1Ø7Ø  ØØØØ              Ø                       /VARIABLE FUNCTION
[illegible] 4535        BREAK                   /1 DATA BREAK
```

```
1Ø72  451Ø            RDBR                      /DB SHOULD STILL BE ØØØØ,
1Ø73  4533            COMPAR
1Ø74  6767            ERØ3E
                /GD=GOOD DB; OD=ADDRESS XBUFF; FS=FUNCTION WHICH FAILED
1Ø75  5777            JMP I   T3EX32            /SUBTEST LOOP
1Ø76  4775            JMS I   T3EX2P            /GET ANOTHER FUNCTION

1Ø77  453Ø            LOOP7
11ØØ  524Ø            JMP     TØ3D-2            /*****LOOP7*****

                /VERIFY SPCFWD, SPCREV, READ, RDCOMP AND WRITE CAUSE
                /WC INCREMENT FROM ØØØØ TO ØØØ1,
11Ø1  72Ø1      TØ3F, CLA IAC
11Ø2  3154            DCA     GOOD
11Ø3  4776            JMS I   T3EX1P            /SET VARIABLE FUNCTION,
11Ø4  6654            T3LS1+5
11Ø5  7773            -5
11Ø6  1111            .+3
11Ø7  44Ø4            JMS I   T3LODP            /LOAD REGISTERS
111Ø  4473            LWCR                      /Ø TO WC
1111  ØØØØ            Ø                         /FUNCTION INSERTED
1112  4535            BREAK                     / 1 DATA BREAK
1113  45Ø3            RWCR
1114  4533            COMPAR
1115  6771            ERØ3F
                /GD=GOOD WC; FS=FUNCTION WHICH FAILED,
1116  5777            JMP I   T3EX32            /SUBTEST LOOP
1117  4775            JMS I   T3EX2P            /GET ANOTHER FUNCTION

                /VERIFY READ, RDCOMP, WRITE CAUSE CA INCREMENT FROM
                /XBUFF TO XBUFF+1
112Ø  72Ø1      TØ3G, CLA IAC
1121  1Ø47            TAD     XBUFFP
1122  3154            DCA     GOOD
1123  4776            JMS I   T3EX1P            /GET A FUNCTION,
1124  6646            T3LS1-1
1125  7775            -3
1126  113Ø            .+2
1127  44Ø4            JMS I   T3LODP            /LOAD REGISTERS
113Ø  ØØØØ            Ø                         /FUNCTION INSERTED HERE,
1131  4535            BREAK                     /1 DATA BREAK,
1132  45Ø4            RCAR
1133  4533            COMPAR
1134  6773            ERØ3G
                /GD=GOOD CA; FS=FAILING FUNCTION,
1135  5777            JMP I   T3EX32            /SUBTEST LOOP
1136  4775            JMS I   T3EX2P            /GET ANOTHER FUNCTION

1137  4531            LOOP8
114Ø  5277            JMP     TØ3F-2            /*****LOOP 8*****

                /VERIFY OFFLINE, REWIND, WEOF DO NOT CAUSE WC INCREMENT,
                /WC PRESET TO ØØØØ.
1141  3154      TØ3H, DCA     GOOD
1142  4776            JMS I   T3EX1P            /GET FUNCTION
```

```
        1143  6651              T3LS1+2
        1144  7775              -3
        1145  1147              .+2
        1146  4473              LWCR                        /Ø TO WC
        1147  ØØØØ              Ø                           /FUNCTION INSERTED
        115Ø  4535              BREAK                       /1 DATA BREAK
        1151  45Ø3              RWCR
        1152  4533              COMPAR
        1153  6775              ERØ3H
                      /GD=GOOD WC; FS=FAILING FUNCTION
        1154  5777              JMP I   T3EX32              /SUBTEST LOOP,
        1155  4775              JMS I   T3EX2P              /GET ANOTHER FUNCTION,

                      /VERIFY OFFLINE, REWIND, WEOF, SPCFWD, SPCREV DO NOT
                      /CAUSE CA INCREMENT. CA PRESET TO ØØØØ,
        1156  4776    TØ3I,     JMS I   T3EX1P              /GET A FUNCTION,
        1157  6651              T3LS1+2
        116Ø  7773              -5
        1161  1163              .+2
        1162  4475              LCAR                        /Ø TO CA
        1163  ØØØØ              Ø                           /FUNCTION
        1164  4535              BREAK                       /1 DATA BREAK
        1165  45Ø4              RCAR
        1166  4533              COMPAR
        1167  6777              ERØ3I
                      /GD=GOOD CA; FS=FAILING FUNCTION
        117Ø  5777              JMP I   T3EX32              /SUBTEST LOOP
        1171  4775              JMS I   T3EX2P              /GET ANOTHER FUNCTION

        1172  4532              LOOP9
        1173  5337              JMP     TØ3H-2              /*****LOOP 9*****

        1174  56ØØ              JMP I   TESTØ3              /EXIT,

        1175  1231    T3EX2P, T3EX2
        1176  1213    T3EX1P, T3EX1
        1177  1227    T3EX32, T3EX3+2
              12ØØ              PAGE

        12ØØ  ØØØØ    T3LOAD, Ø                             /LOAD XBUFF AND XBUFF+1
        12Ø1  4524              CLEAR2
        12Ø2  724Ø              CLA CMA                     /WITH 7777, AND LCAR WITH
        12Ø3  3Ø5Ø              DCA     XBUFF               /XBUFF ADDRESS,
        12Ø4  724Ø              CLA CMA
        12Ø5  3Ø51              DCA     XBUFF+1
        12Ø6  1Ø47              TAD     XBUFFP
        12Ø7  4475              LCAR
        121Ø  1165              TAD     PRGFLD
        1211  4477              LCMR
        1212  56ØØ              JMP I   T3LOAD

        1213  ØØØØ    T3EX1,    Ø                           /TEST Ø3 FUNCTION SELECTOR EXECUTIVE,
        1214  1613              TAD I   T3EX1               /GET CALL +1 = THIS IS STARTING
        1215  3Ø1Ø              DCA     1Ø                  /FUNCTION IN T3LS1,
          16  2213              ISZ     T3EX1               /GET CALL +2 = THIS IS THE NUMBER
```

```
      1217  1613            TAD I   T3EX1         /OF FUNCTIONS TO BE USED,
      1220  3157            DCA     TXXTM1
      1221  2213            ISZ     T3EX1         /UPDATE CALL POINTER TO LOCATION IN
      1222  1613            TAD I   T3EX1
      1223  3160            DCA     TXXTM2
      1224  2213            ISZ     T3EX1
      1225  1410    T3EX3,  TAD I   1Ø            /WHICH FUNCTION TO BE INSERTED AND
      1226  356Ø            DCA I   TXXTM2        /POINT OF EXIT,
      1227  4524            CLEAR2
      123Ø  5613            JMP I   T3EX1         /RUN TEST WITH THIS FUNCTION,
      1231  ØØØØ    T3EX2,  Ø                     /ENTER AT END OF SUBTEST,
      1232  2157            ISZ     TXXTM1        /ALL FUNCTIONS USED?
      1233  5225            JMP     T3EX3         /NO,  GET NEXT FUNCTION,
      1234  5631            JMP I   T3EX2         /YES,  EXIT TO END OF THIS SUBTEST
                                                  /PLUS 1 INSTRUCTION,
                        /TEST Ø4,  FULL WORD COUNT AND CURRENT ADDRESS INCREMENT TEST
                                /INCLUDING CM6-8,

                                /1.  REFER TO SUBTEST FOR SPECIFIC TEST INFORMATION,
                                /2.  LOOPS 6-7:
                                /2.1  LOOP 6 CYCLES ON TØ4A,  (WC INCREMENT)
                                /2.2  LOOP 7 CYCLES ON TØ4B, CD, AND E (CA AND EMA TESTS),
                                /3.  THE SUBTEST LOOPS CYCLE ON THE PRESENT OLD VALUE
                                /BEING INCREMENTED TO THE GOOD VALUE,

        1235  ØØØØ      TESTØ4, Ø
        1236  4523              CLEAR1


                        /VERIFY COMPLETE WC INCREMENT USING SPCFWD FUNCTION,
                        /START WITH WC=Ø AND INCREMENT A FULL COUNT FOUR
                        /TIMES DISREGARDING WC OVERFLOW,
        1237  4545      TØ4A,   SPCFWD                  /SPACE FORWARD,
        124Ø  1Ø4Ø              TAD     M4
        1241  3157              DCA     TXXTM1
        1242  3154              DCA     GOOD
        1243  1154      TØ4AL,  TAD     GOOD            /GOOD TO OLD,
        1244  3156              DCA     OLD
        1245  2154              ISZ     GOOD            /+1 TO GOOD,
        1246  5252              JMP     TØ4ASL
        1247  2157              ISZ     TXXTM1
        125Ø  741Ø              SKP
        1251  5262              JMP     T4LP6           /YES,
        1252  1156      TØ4ASL, TAD     OLD             /LOAD WC WITH OLD,
        1253  4473              LWCR
        1254  4535              BREAK                   /1 BREAK,
        1255  45Ø3              RWCR                    /CHECK
        1256  4533              COMPAR
        1257  7ØØ1              ERØ4A
                        /WC INCREMENT FAILURE, GD=GOOD WC; OD=PREVIOUS GOOD WC;
                        /WC=REAL WC.  THE SUBTEST LOOP CYCLES ON THE CURRENT PREVIOUS
                        /GOOD WC (OD) BEING INCREMENTED TO THE GOOD WC VALUE,
        126Ø  5252              JMP     TØ4ASL          /SUBTEST LOOP,
        1261  5243              JMP     TØ4AL

        1262  4527      T4LP6,  LOOP6
        1263  5237              JMP     TØ4A            /*****LOOP 6*****

                        /VERIFY THAT"GO"BIT CAN BE SET WITH MITF CLEAR,
        1264  1Ø27      TØ4B,   TAD     K1ØØ
        1265  3154              DCA     GOOD
        1266  4524              CLEAR2                  /CLEAR ALL REGISTERS INCLUDING
        1267  4543              WRITE                   /MTTF,
        127Ø  455Ø              GO                      /GO,
        1271  45Ø7              RFSR                    /GO SHOULD BE SET,
        1272  ØØ27              AND     K1ØØ
        1273  4533              COMPAR
        1274  7ØØ3              ERØ4B
                        /GO BIT NOT SET, GD=GOOD GO BIT; BD=REAL    BIT FROM FS,
1275  5264              JMP     T04B            /SUBTEST LOOP,


                  /VERIFY COMPLETE CA AND EMA INCREMENT, T04C PERTAINS
                  /TO CA TEST, TO4D TO EMA TEST, START WITH
                  /CA=0, EMA=0, CA WILL OVERFLOW, EMA WILL COUNT TO
                  /7 THEN FREEZE, CA SHOULD CONTINUE TO WRAP AROUND,
1276  1043        T04CD,  TAD     M20             /20 PASSES,
1277  3157              DCA     TXXTM1
1300  3160              DCA     TXXTM2
1301  3376              DCA     T04T1
1302  1160        T04CL,  TAD     TXXTM2
1303  3161              DCA     TXXTM3
1304  1376              TAD     T04T1
1305  3377              DCA     T04T2
1306  2160              ISZ     TXXTM2
1307  5324              JMP     T04CSL
1310  2157              ISZ     TXXTM1
1311  7410              SKP
1312  5373              JMP     T4LP7
1313  1376              TAD     T04T1
1314  1023              TAD     K10
1315  0026              AND     K70
1316  3376              DCA     T04T1
1317  1376              TAD     T04T1
1320  7640              SZA CLA
1321  5324              JMP     .+3
1322  1026              TAD     K70
1323  3376              DCA     T04T1
1324  1160        T04CSL, TAD     TXXTM2
1325  3154              DCA     GOOD
1326  4517              CLT                     /CLEAR ALL,
1327  1161              TAD     TXXTM3          /SET UP OLD,
1330  3156              DCA     OLD
1331  1161              TAD     TXXTM3          /LOAD CA WITH OLD VALUE,
1332  4475              LCAR
1333  1377              TAD     T04T2           /LOAD EMA WITH OLD VALUE,
1334  4477              LCMR
1335  4456              JMS I   FUNCP           /WRITE-GO WITH EMA INC ENABLE,
1336  4140              4140
1337  4535              BREAK                   /1 BREAK
1340  4504              RCAR                    /CHECK CA INCREMENT
1341  4533              COMPAR
1342  7005              ER04C
                  /CA INCREMENT FAILURE, CA DID NOT INCREMENT FROM
                  /OLD TO GOOD VALUE. GD=GOOD CA; OD=OLD CA; CA=REAL CA,
                  /SUBTEST LOOP CYCLES ON OLD BEING INCREMENTED TO GOOD,
1343  7000              NOP
1344  7410              SKP
1345  5357              JMP     T04E
1346  1376              TAD     T04T1           /PUT GOOD EMA IN GOOD,
1347  3154              DCA     GOOD
1350  1377              TAD     T04T2           /OLD EMA IN OLD,
1351  3156              DCA     OLD
1352  4506              RCMR                    /CHECK EMA (CM6-8)
```

```
       1353  4533                COMPAR
       1354  7007                ERØ4D
                         /EMA INCREMENT OR FREEZE FAILURE, EMA (CM6-8) DID
                         /NOT INCREMENT FROM OLD TO GOOD VALUE OR DID
                         /INCREMENT OR CHANGE WHEN IT SHOULD NOT HAVE,
                         /GD=GOOD EMA; OD=OLD EMA; CM=REAL EMA IN
                         /BITS 6-8, THE SUBTEST LOOP CYCLES ON CURRENT DATA,
       1355  5324                JMP     TØ4CSL          /SUBTEST LOOP,
       1356  53Ø2                JMP     TØ4CL

                         /VERIFY THAT EMA DOES NOT INCREMENT WHEN EMA INC
                         /DISABLED, CA SET TO 7777, EMA TO Ø, THEN CAUSE CA
                         /INCREMENT AND EMA SHOULD REMAIN Ø,
       1357  4524        TØ4E,   CLEAR2                  /CLEAR ALL,
       136Ø  3154                DCA     GOOD
       1361  724Ø                CLA CMA                 /7777 TO CA,
       1362  4475                LCAR
       1363  4477                LCMR                    /Ø TO EMA
       1364  4456                JMS I   FUNCP           /WRITE-GO, EMA INC
       1365  41ØØ                41ØØ                    /DISABLED,
       1366  4535                BREAK                   /1 BREAK,
       1367  45Ø6                RCMR                    /CHECK EMA=Ø,
       137Ø  4533                COMPAR
       1371  7Ø11                ERØ4E
                         /EMA CHANGED WHEN EMA INC DISABLED, GD=GOOD EMA;
                         /CM=REAL EMA IN BITS 6-8,
       1372  5357                JMP     TØ4E            /SUBTEST LOOP,

       1373  453Ø        T4LP7,  LOOP7
       1374  5262                JMP     T4LP6           /*****LOOP 7*****
       1375  5635                JMP I   TESTØ4          /EXIT
       1376  ØØØØ        TØ4T1,  Ø
       1377  ØØØØ        TØ4T2,  Ø
```

```
                      /TEST Ø5,  MEMORY-DATA BUFFER DATA TEST,

                              /1,  VERIFIES THAT DB CAN RECEIVE DATA FROM MEMORY AND
                      /CAN TRANSFER DATA TO MEMORY,

                              /2,  ALL SUBTEST LOOPS CYCLE ON CONSTANT DATA,

                              /3,  LOOPS 6-7:
                              /3,1  LOOP 6 CYCLES ON TØ5A (MEM TO DB-WRITE)
                              /3,2  LOOP 7 CYCLES ON TØ5B (DB TO MEM-READ)

                              /4,  LOCATION XBUFF IS LOADED INTO CA
                      /WHICH CAUSES XBUFF+1 TO BE USED FOR TRANSFERS,


            14ØØ              PAGE
      14ØØ  ØØØØ      TESTØ5, Ø
      14Ø1  4523              CLEAR1

                      /VERIFY PROPER DATA TRANSFERS FROM XBUFF+1 IN MEMORY TO
                      /DB USING WRITE FUNCTION,  CA IS SET TO XBUFF AND
                      /SHOULD INCREMENT TO XBUFF+1 BEFORE TRANSFER,  COMPLEMENT
                      /INCREMENTING DATA IS USED,
      14Ø2  4543      TØ5A,   WRITE                   /SET WRITE FUNCTION,
      14Ø3  465Ø              JMS I   TØ5LDS          /GET DATA,
      14Ø4  7777              7777
      14Ø5  5222              JMP     T5LP6           /DONE EXIT,
      14Ø6  1Ø47      TØ5ASL, TAD     XBUFFP          /LOAD CA=XBUFF,
      14Ø7  4475              LCAR
      141Ø  1157              TAD     TXXTM1          /LOAD XBUFF+1 WITH DATA WORD,
      1411  3Ø51              DCA     XBUFF+1
      1412  1165              TAD     PRGFLD          /LOAD CM WITH PROPER MEMORY
      1413  4477              LCMR                    /FIELD,
      1414  4535              BREAK                   /1 DATA BREAK,
      1415  451Ø              RDBR                    /READ AND CHECK DB,
      1416  4533              COMPAR
      1417  7Ø13              ERØ5A
                      /GD GOOD DB; OD=PREVIOUS GOOD DB; DB=REAL DB,
      142Ø  52Ø6              JMP     TØ5ASL          /SUBTEST LOOP,
      1421  5651              JMP I   TØ5LDR

      1422  4527      T5LP6,  LOOP6
      1423  52Ø2              JMP     TØ5A            /*****LOOP 6*****

                      /VERIFY PROPER DATA TRANSFERS FROM DB TO XBUFF+1 (IN MEMORY)
                      /USING THE READ FUNCTION,  CA SET TO XBUFF AND SHOULD
                      /INCREMENT TO XBUFF+1 BEFORE TRANSFER,  COMPLEMENT INCREMENTING
                      /DATA USED,
      1424  465Ø      TØ5B,   JMS I   TØ5LDS          /GET DATA,
      1425  7777              7777
      1426  5245              JMP     T5LP7           /DONE EXIT,
      1427  4524      TØ5BSL, CLEAR2                  /CLEAR REGISTERS,
      143Ø  1Ø47              TAD     XBUFFP          /LOAD CA WITH XBUFF,
      1431  4475              LCAR
```

```
        1432  1157          TAD     TXXTM1          /DATA TO DB,
        1433  45Ø1          LDBR
        1434  1165          TAD     PRGFLD          /FIELD TO CM,
        1435  4477          LCMR
        1436  4541          READ                    /SET READ FUNCTION
        1437  4535          BREAK                   /1 DATA BREAK,
        144Ø  1Ø51          TAD     XBUFF+1         /COMPARE MEMORY


        1441  4533          COMPAR
        1442  7Ø15          ERØ5B
                      /GD= GOOD CONTENTS OF XBUFF+1: BD:REAL CONTENTS OF XBUFF+1:
                      /OD= PREVIOUS GOOD CONTENTS OF XBUFF+1,
        1443  5227          JMP     TØ5BSL          /SUBTEST LOOP,
        1444  5651          JMP I   TØ5LDR

        1445  453Ø    T5LP7, LOOP7
        1446  5222          JMP     T5LP6           /*****LOOP 7*****
        1447  56ØØ          JMP I   TESTØ5

        145Ø  Ø7Ø7    TØ5LDS, TØ2LDS
        1451  Ø731    TØ5LDR, TØ2LDR
```

```
                        /TEST Ø6,   CURRENT ADDRESS MEMORY REFERENCE TEST,

                                /1,   VERIFIES THAT CA AND CM6=8 CAN REFERENCE MEMORY
                        /CORRECTLY IN ALL EXISTING MEMORY FIELDS,
                                /2,   THE METHOD USED INSERTS THE CURRENT ADDRESS AND THEN THE COMPLEMENT
                        /OF THE ADDRESS IN THE CURRENT MEMORY LOCATION FOR DATA,   THE CA AND EMA ARE SET TO THE
                        /CURRENT LOCATION = 1, THEN A 1 WORD SIMULATED WRITE IS
                        /EXECUTED,   MEMORY IS RESTORED TO ITS ORIGINAL CONTENTS AND THEN
                        /DB AND DATA ARE COMPARED,   DATA IS NOT INSERTED IN LOCATIONS
                        /MARK1 THROUGH MARK2,   THE ACTUAL MEMORY CONTENTS APE USED FOR DATA,
                                /3,   THE SUBTEST LOOP CYCLES ON THE SAME MEMORY LOCATION
                        /USING THE SAME DATA,


          16ØØ                  PAGE
    16ØØ  ØØØØ          TESTØ6, Ø
    16Ø1  4523                  CLEAR1

                        /VERIFY THAT CA AND EMA (CM6=8) REFERENCE THE CORRECT MEMORY LOCATION
                        /AND THAT DB RECEIVES CORRECT DATA, WRITE FUNCTION USED, DATA IS ADDRESS
                        /AND COMPLEMENT ADDRESS,
    16Ø2  4242          TØ6A,   JMS     T6EX1
    16Ø3  4517                  CLT
    16Ø4  1156                  TAD     OLD             /SET UP CA AND CM,
    16Ø5  4475                  LCAR
    16Ø6  116Ø                  TAD     TXXTM2
    16Ø7  4477                  LCMR
    161Ø  4456                  JMS I   FUNCP           /WRITE=GO=EMA INCREMENT ENABLED
    1611  414Ø                  414Ø
    1612  6ØØ2                  IOF                     /INTERRUPT SYSTEM TURNED
    1613  1161                  TAD     TXXTM3          /OFF,
    1614  3154                  DCA     GOOD
    1615  62Ø1          TØ6ADF, CDF     /N              /CHANGE TO CURRENT DATA FIELD,
    1616  1623                  TAD I   TØ6AMP          /GET ORIGINAL MEM AND SAVE,
    1617  3361                  DCA     T6SAVE
    162Ø  1161                  TAD     TXXTM3          /GET DATA AND PUT IN MEM,
    1621  3623                  DCA I   TØ6AMP
    1622  741Ø          MARK1,  SKP
    1623  ØØØØ          TØ6AMP, Ø
    1624  6727          MARK3,  6727                    /DATA BREAK,
    1625  7ØØØ                  NOP
    1626  1361                  TAD     T6SAVE          /RESTORE MEMORY,
    1627  3623          MARK2,  DCA I   TØ6AMP
    163Ø  4457                  JMS I   SCDFP           /CHANGE TO THIS PROGRAMS DATA FIELD,
    1631  6ØØ1                  ION                     /INTERRUPT SYSTEM ON,
    1632  451Ø                  RDBR                    /DATA SHOULD BE IN DB,
    1633  4533                  COMPAR
    1634  7Ø17                  ERØ6A
                        /GD=GOOD DB; OD=CURRENT MEMORY LOCATION BEING USED MINUS ONE; DB=REAL DB
    1635  52Ø3                  JMP     TØ6A+1          /SUBTEST LOOP,
    1636  4264                  JMS     T6EX2

    1637  4527                  LOOP6
    164Ø  52Ø1                  JMP     TØ6A-1          /*****LOOP 6*****
```

```
        1641  5600              JMP I   TEST06


        1642  0000      T6Ex1,  0                       /TEST 06 LOCAL EXECUTIVE.
        1643  1167              TAD     MEMFLD          /GET EXTENDED MEM FIELDS
        1644  3157              DCA     TXXTM1          /AND PUT IN TEMP.
        1645  1046              TAD     KCDF            /INITIALIZE CDF INSTRUCTION.
        1646  3215              DCA     T06ADF
        1647  3223              DCA     T06AMP          /SET STARTING ADDRESS TO 0.
        1650  1362              TAD     MARK1P          /MAKE THE ADDRESS MARK1 INTO
        1651  7041              CIA                     /A COUNTER.
        1652  3364              DCA     T6CNT1
        1653  1364              TAD     T6CNT1          /MAKE (MARK2-MARK1)+2 INTO
        1654  1363              TAD     MARK2P          /A COUNTER.
        1655  7041              CIA
        1656  3365              DCA     T6CNT2
        1657  7040              CMA                     /FAKE THE COMPLEMENT COUNTER SO
        1660  3366              DCA     T6CNT3          /IT RESETS ON FIRST ENTRY.
        1661  1356              TAD     KNOP            /SET T6I1 TO NOP.
        1662  3313              DCA     T6I1
        1663  7410              SKP
        1664  0000      T6EX2,  0                       /ENTER HERE AT END OF EACH PASS.
        1665  2366              ISZ     T6CNT3          /TIME FOR COMPLEMENT DATA?
        1666  7610              SKP CLA
        1667  5273              JMP     .+4
        1670  7240              CLA CMA                 /YES, SET COMPLEMENT DATA
        1671  3367              DCA     T6FLAG          /FLAG, THEN GO TO T6I1
        1672  5313              JMP     T6I1            /FOR ROUTING.
        1673  3367              DCA     T6FLAG          /NO,  CLEAR COMPLEMENT FLAG.
        1674  1037              TAD     M2              /RESET COMPLEMENT COUNTER.
        1675  3366              DCA     T6CNT3
        1676  1223              TAD     T06AMP          /PUT PREVIOUS ADDRESS IN
        1677  3156              DCA     OLD             /OLD.
        1700  1215              TAD     T06ADF          /PUTPREVIOUS DATA FIELD
        1701  0026              AND     K70             /IN TEMP FOR LOADING
        1702  3160              DCA     TXXTM2          /CM6-8.
        1703  2223              ISZ     T06AMP          /UPDATE TO NEW CURRENT ADDRESS.
        1704  5313              JMP     T6I1            /NO OVERFLOW - GO TO T6I1 FOR ROUTING.
        1705  2157              ISZ     TXXTM1          /ADDRESS OVERFLOW, CHECK FOR
        1706  7410              SKP                     /ANOTHER EXISTING MEM FIELD.
        1707  5664              JMP I   T6EX2           /NO MORE LEFT, EXIT TO END OF TEST.
        1710  1215              TAD     T06ADF          /SOME FIELDS LEFT, UPDATE
        1711  1023              TAD     K10             /CDF INSTRUCTION FOR NEW
        1712  3215              DCA     T06ADF          /FIELD.
        1713  0000      T6I1,   0               /THIS INSTRUCTION CAN BE ONE OF THREE INSTRUCTIONS:
                        /       NOP             /PRIOR TO REACHING MARK1;
                        /       JMP     T6EX3   /AFTER REACHING MARK1 BUT NOT MARK2;
                        /       JMP     T6EX4   /AFTER PASSING MARK2.
        1714  1367              TAD     T6FLAG          /NOT UP TO MARK1.  TIME FORDATA COMPLEMENT?
        1715  7710              SPA CLA
        1716  5347              JMP     T6EX4           /YES, BYPASS MARK1 UPDATES.
        1717  1160              TAD     TXXTM2          /HAS THE CURRENT ADDRESS REACHED
        1720  7041              CIA                     /THE PROGRAM FIELD?
        1721  1165              TAD     PRGFLD
        1722  7650              SNA CLA                 /NO, BYPASS MARK1 UPDATES.
```

```
1723  2364            ISZ     T6CNT1        /NO, UPDATE MARK1 COUNTER,
1724  5347            JMP     T6EX4         /NOT THERE YET, USE DATA,
1725  1357            TAD     KJMP61        /AT MARK1, CHANGE ROUTING TO T6EX3,
1726  3313            DCA     T6I1
1727  1367    T6EX3,  TAD     T6FLAG        /PASSED OR AT MARK1, COMPLEMENT
1730  7710            SPA CLA               /DATA PASS?
1731  5337            JMP     T6EX5         /YES, BYPASS (MARK2-MARK1)+2 UPDATES,
1732  2365            ISZ     T6CNT2        /NO, ARE WE PASSED MARK2?
1733  5337            JMP     T6EX5         /NO, USE ACTUAL MEM CONTENTS,
1734  1360            TAD     KJMP62        /YES, CHANGE ROUTING TO EX4,
1735  3313            DCA     T6I1
1736  5347            JMP     T6EX4
1737  1215    T6EX5,  TAD     TØ6ADF        /USE MEM CONTENTS VICE DATA, GET CURRENT
1740  3342            DCA     .+2           /CDF INSTRUCTION AND STORE FOR USE,
1741  6002            IOF                   /INTERRUPT SYSTEM OFF,
1742  6201            CDF     /N            /CHANGE TO CURRENT DF,
1743  1623            TAD I   TØ6AMP        /GET CONTENTS OF CURRENT ADDRESS,
1744  4457            JMS I   SCDFP         /RESET TO DF OF PROGRAM,
1745  6001            ION                   /INTERRUPT SYSTEM ON,
1746  5353            JMP     T6EX6         /PUT MEM CONTENTS IN TEMP AND EXIT,
1747  1223    T6EX4,  TAD     TØ6AMP        /USE DATA, GET CURRENT ADDRESS
1750  2367            ISZ     T6FLAG        /FOR DATA, TIME FOR COMPLEMENT?
1751  7040            CMA                   /NO, COMPLEMENT DATA ONCE,
1752  7040            CMA                   /YES, DITTO OR COMPLEMENT DATA AGAIN,
1753  3161    T6EX6,  DCA     TXXTM3        /STORE DATA OR MEM CONTENTS IN TEMP
1754  4474            CWCR
1755  5642            JMP I   T6EX1         /USED FOR STORING DATA IN CURRENT
                                            /LOCATION,   THEN RUN TEST,

1756  7000    KNOP,   NOP
1757  5327    KJMP61, JMP     T6EX3
1760  5347    KJMP62, JMP     T6EX4
1761  0000    T6SAVE, Ø
1762  1622    MARK1P, MARK1
1763  1631    MARK2P, MARK2+2               /BIASED BY +2 TO YIELD CORRECT TALLY
                                            /NUMBER FOR (MARK2-MARK1)+2,
1764  ØØØØ    T6CNT1, Ø                     /MARK1 COUNTER,
1765  ØØØØ    T6CNT2, Ø                     /(MARK2-MARK1)+2 COUNTER,
1766  ØØØØ    T6CNT3, Ø                     /COMPLEMENT DATA COUNTER,
1767  ØØØØ    T6FLAG, Ø                     /SET TO 7777 IF TIME FOR COMPLEMENT DATA,
                        /TEST Ø7:  STATUS REGISTERS STAND ALONE TEST.

                                /1.  THE PURPOSE OF THIS TEST IS TO VERIFY THAT MS AND FS
                        /INDICATE ONLY WHAT IS EXPECTED WITH THE TRANSPORT SYSTEM
                        /EFFECTIVELY DISCONNECTED FROM THE TM8E CONTROL.  ALL INDICATORS
                        /WHICH MAY BE IN ONE STATE OR ANOTHER ARE DISREGARDED.

                                /2.  LOOPS 6-7:
                                /2.1  LOOP 6 CYCLES ON MS CHECK (TØ7A-TØ7B)
                                /2.2  LOOP 7 CYCLES ON S1 CHECK (TØ7C-TØ7D)

        2ØØØ                    PAGE
  2ØØØ  ØØØØ            TESTØ7, Ø
  2ØØ1  4523                    CLEAR1

                        /WITH ALL DRIVES OFF LINE OR POWERED DOWN THE MS SHOULD
                        /INDICATE ONLY THE SELECT REMOTE BIT SET DISCOUNTING
                        /EF, R/C AND IF BITS.
  2ØØ2  1Ø32            TØ7A,   TAD     K4ØØ            /SELECT REMOTE BIT TO GOOD.
  2ØØ3  3154                    DCA     GOOD
  2ØØ4  45Ø5                    RMSR                    /READ MS AND MASK OUT
  2ØØ5  Ø251                    AND     K3774           /EF, R/C AND IF BITS.
  2ØØ6  4533                    COMPAR
  2ØØ7  7Ø21                    ERØ7A
                        /GD=GOOD MS MASKED; BD=REAL MS MASKED.
  2Ø1Ø  52Ø4                    JMP     TØ7A+2          /SUBTEST LOOP

                        /VERIFY THAT RMSR CLEARS AC BEFORE READING.  SELECT REMOTE
                        /SHOULD BE ONLY BIT SET.
  2Ø11  724Ø            TØ7B,   CLA CMA                 /AC=7777.
  2Ø12  45Ø5                    RMSR                    /READ MS AND MASK OUT
  2Ø13  Ø251                    AND     K3774           /EF,R/C AND IF BITS.
  2Ø14  4533                    COMPAR
  2Ø15  7Ø23                    ERØ7B
                        /GD=GOOD MS MASKED; BD=REAL MS MASKED.
  2Ø16  5211                    JMP     TØ7B            /SUBTEST LOOP.

  2Ø17  4527                    LOOP6
  2Ø2Ø  52Ø2                    JMP     TØ7A            /*****LOOP6*****

                        /FS SHOULD INDICATE BIT 6 SET SINCE NO DRIVE ON LINE
                        /DEFAULTS TO 9 TRK.
  2Ø21  4537            TØ7C,   OFFLIN                  /FUNCTION TO ØØ.
  2Ø22  1Ø25                    TAD     K4Ø
  2Ø23  3154                    DCA     GOOD
  2Ø24  1ØØ5                    TAD     K3              /LOAD 9 TRK 8ØØBPI
  2Ø25  4477                    LCMR
  2Ø26  45Ø7                    RFSR
  2Ø27  4533                    COMPAR
  2Ø3Ø  7Ø25                    ERØ7C
                        /GD=GOOD FS; BD=REAL FS.
  2Ø31  5224                    JMP     TØ7C+3          /SUBTEST LOOP.

                        /VERIFY THAT RFSR CLEARS AC PRIOR TO READING.
```

```
2Ø32  724Ø    TØ7D,   CLA CMA                 /AC=7777,
2Ø33  45Ø7            RFSR                    /READ FS-9TRK BIT SHOULD
2Ø34  4533            COMPAR                  /STILL BE SET,
2Ø35  7Ø27            ERØ7D
              /GD=GOOD FS; BD=REAL FS,
2Ø36  5232            JMP     TØ7D            /SUBTEST LOOP

              /VERIFY THAT CM BIT 11 = Ø FORCES 7 CHANNEL,  (FS BIT 6 = Ø)
2Ø37  3154    TØ7E,   DCA     GOOD
2Ø4Ø  7126            STL RTL                 /DENSITY BITS=1Ø
2Ø41  4477            LCMR
2Ø42  45Ø7            RFSR
2Ø43  4533            COMPAR
2Ø44  7Ø31            ERØ7E
              /GD=GOOD FS;  BD=REAL FS,
2Ø45  524Ø            JMP     TØ7E+1          /SUBTEST LOOP,

2Ø46  453Ø            LOOP7
2Ø47  5217            JMP     TØ7C-2          /*****LOOP 7*****

2Ø5Ø  56ØØ            JMP I   TESTØ7
2Ø51  3774    K3774,  3774
```

```
                /TEST 1Ø,       BASIC TESTS FOR MTTF, GO BIT, IF, R/C ERROR, EMA OVERFLOW
                /AND EF,
                        /VERIFIES EXISTENCE AND PROPER FUNCTIONING OF THE ABOVE ON A
                /PARTIAL BASIS,

                        /LOOPS 6-9;
                        /LOOP 6 CYCLES ON MTTF AND GO BIT TESTS (T1ØA-T1ØJ)
                        /LOOP 7 CYCLES ON IF TESTS (T1ØK-T1ØR)
                        /LOOP 8 CYCLES ON R/C AND EMA CHECKS (T1ØS-T1ØW)
                        /LOOP 9 CYCLES ON EF CHECKS (T1ØX-T1Ø3)



      2200              PAGE
2200  0000      TEST1Ø, Ø
2201  4523              CLEAR1

2202  4204              JMS     T1ØA            /EXECUTE T1ØA
2203  5216              JMP     T1ØB            /THEN GO TO T1ØB

                /VERIFY CLT CLEARS MTTF,  CHECK VIA SKTD NOT SKIPPING,
2204  0000      T1ØA,   Ø                       /JMS'D HERE
2205  3154              DCA     GOOD
2206  4524              CLEAR2                  /CLT
2207  4514              SKTD                    /SHOULD NOT SKIP,
2210  7410              SKP
2211  7040              CMA
2212  4533              COMPAR
2213  7033              ER1ØA
2214  5206              JMP     T1ØA+2          /SUBTEST LOOP,
2215  5604              JMP I   T1ØA

                /VERIFY WCOV OCCURS WITH WC=7777, SPACE FORWARD THEN
                /1 DATA BREAK FOLLOWED BY LDBR,  LDBR,WCOV SHOULD SET MTTF AND
                /SKTD SHOULD SKIP,
2216  4545      T1ØB,   SPCFWD                  /SPACE FORWARD,
2217  4553              WCOV                    /FORCE WCOV
2220  4501              LDBR                    /SET MTTF
2221  4514              SKTD                    /MTTF SHOULD BE SET CAUSING
2222  7040              CMA                     /SKTD TO SKIP,
2223  4533              COMPAR
2224  7035              ER1ØB
2225  5216              JMP     T1ØB            /SUBTEST LOOP,

2226  4204              JMS     T1ØA            /T1ØA AGAIN
2227  7240              CLA CMA                 /SET UP FOR NEXT,
2230  3154              DCA     GOOD

                /VERIFY LDBR WITH NO WCOV DOES NOT SET MTTF,
                /CHECK VIA SKTD.
2231  4524      T1ØC,   CLEAR2                  /CLEAR ALL,
2232  4473              LWCR                    /ENSURE WC=ØØØØ
  33  4545              SPCFWD                  /SPACE FORWARD FUNCTION,
  34  4550              GO                      /GO,
 2235  4535           BREAK                   /1 DATA BREAK.
 2236  4501           LDBR
 2237  4514           SKTD                    /SKTD SHOULD NOT SKIP
 2240  7040           CMA
 2241  4533           COMPAR
 2242  7037           ER1ØC
 2243  5231           JMP     T1ØC            /SUBTEST LOOP.

                /VERIFY WCOV AND AN IOT OTHER THAN LDBR
                /DOES NOT SET MTTF.  VARIABLE IOT (I1!) INSERTED BY EXECUTIVE.
 2244  4524     T1ØD,     CLEAR2
 2245  4761           JMS I   T1ØX1P          /GET IOT.
 2246  7477           T1ØLS1-1
 2247  7753           -25
 2250  2253           .+3
 2251  4545           SPCFWD
 2252  4553           WCOV
 2253  ØØØØ           Ø                       /PUT IT HERE.

 2254  7ØØØ           NOP
 2255  72ØØ           CLA
 2256  4514           SKTD                    /SKTD SHOULD NOT SKIP.
 2257  7Ø4Ø           CMA
 226Ø  4533           COMPAR
 2261  7Ø41           ER1ØD
                /I1=VARIABLE IOT
 2262  5762           JMP I   T1ØX2P          /SUBTEST LOOP.
 2263  4763           JMS I   T1ØX3P          /GET MORE IOT'S.



                /VERIFY THAT REWIND FUNCTION WITHOUT RW STATUS (SINCE
                /NO DRIVE ON LINE) DOES NOT SET MTTF.
 2264  4524     T1ØE,     CLEAR2
 2265  454Ø           REWIND
 2266  4514           SKTD                    /SHOULD NOT SKIP.
 2267  7Ø4Ø           CMA
 227Ø  4533           COMPAR
 2271  7Ø43           ER1ØE
 2272  5264           JMP     T1ØE            /SUBTEST LOOP.

                /VERIFY SPACE REVERSE WITHOUT BOT DOES NOT SET MTTF.
 2273  4524     T1ØF,     CLEAR2
 2274  4546           SPCREV
 2275  4514           SKTD
 2276  7Ø4Ø           CMA
 2277  4533           COMPAR
 23ØØ  7Ø45           ER1ØF
 23Ø1  5273           JMP     T1ØF            /SUBTEST LOOP.

                /VERIFY READ, READ COMPARE OR WRITE WITHOUT LPCS DOES NOT SET
                /MTTF.
 23Ø2  4524     T1ØG,     CLEAR2
 23Ø3  4541           READ
 23Ø4  4542           RDCOMP
```

```
        2305  4543            WRITE
        2306  4514            SKTD                        /SHOULD NOT SKIP.
        2307  7Ø4Ø            CMA
        231Ø  4533            COMPAR
        2311  7Ø47            ER1ØG
        2312  53Ø2            JMP     T1ØG                /SUBTEST LOOP.

                      /VERIFY "GO" BIT SETS WHEN LOADED AND MTTF CLEAR.
        2313  1Ø27    T1ØH,   TAD     K1ØØ
        2314  3154            DCA     GOOD
        2315  4524            CLEAR2
        2316  455Ø            GO
        2317  45Ø7            RFSR
        232Ø  ØØ27            AND     K1ØØ
        2321  4533            COMPAR
        2322  7Ø51            ER1ØH
                      /GD= GOOD "GO" BIT; BD= REAL "GO" BIT.
        2323  5315            JMP     T1ØH+2              /SUBTEST LOOP.

                      /VERIFY "GO" BIT CLEARS AFTER LOADED WITH MTTF SET.
        2324  3154    T1ØI,   DCA     GOOD
        2325  4545            SPCFWD                      /SET MTTF.
        2326  4553            WCOV
        2327  45Ø1            LDBR                        /SET MTTF.
        233Ø  455Ø            GO                          /SET "GO"; SHOULD CLEAR
        2331  45Ø7            RFSR                        /IMMEDIATELY.
        2332  ØØ27            AND     K1ØØ
        2333  4533            COMPAR
        2334  7Ø53            ER1ØI
                      /GD= GOOD "GO" BIT; BD= REAL GO BIT.
        2335  5325            JMP     T1ØI+1              /SUBTEST LOOP.



                      /VERIFY CLT CLEARS "GO" BIT.
        2336  4524    T1ØJ,   CLEAR2
        2337  455Ø            GO                          /SET "GO"
        234Ø  4524            CLEAR2                      /CLT
        2341  45Ø7            RFSR                        /CHECK CLEAR.
        2342  ØØ27            AND     K1ØØ
        2343  4533            COMPAR
        2344  7Ø55            ER1ØJ
                      /GD= GOOD "GO" BIT; BD= REAL "GO" BIT.
        2345  5337            JMP     T1ØJ+1              /SUBTEST LOOP.

        2346  4527            LOOP6
        2347  52Ø1            JMP     TEST1Ø+1            /*****LOOP 6*****

                      /VERIFY CLT CLEARS ILLEGAL FUNCTION (IF). ACTUALLY IT SHOULD
                      /NEVER BE SET IN THIS TEST - SEE T1ØQ.
        235Ø  3154    T1ØK,   DCA     GOOD
        2351  4524            CLEAR2
        2352  45Ø5            RMSR
        2353  ØØ2Ø            AND     K1
              4533            COMPAR
```

```
2355  7057            ER1ØK
                /GD= GOOD "IF"; BD= REAL "IF"
2356  5351            JMP     T1ØK+1          /SUBTEST LOOP

2357  5760            JMP I   .+1
236Ø  24ØØ            T1ØØ
2361  1213      T1ØX1P, T3EX1
2362  1227      T1ØX2P, T3EX3+2
2363  1231      T1ØX3P, T3EX2

      24ØØ            PAGE

                /VERIFY WRITE OR WEOF WITHOUT FILE PROTECT DOES NOT SET IF.
24ØØ  4524      T1ØO,   CLEAR2
24Ø1  4543            WRITE
24Ø2  4544            WEOF
24Ø3  45Ø5            RMSR
24Ø4  ØØ2Ø            AND     K1
24Ø5  4533            COMPAR
24Ø6  7Ø61            ER1ØO
                /GD= GOOD IF; BD= REAL IF (BIT 11)
24Ø7  52ØØ            JMP     T1ØO            /SUBTEST LOOP

                /VERIFY LCMR, LFGR OR LDBR WITHOUT CONTROL BUSY DO NOT
                /SET IF.
241Ø  4524      T1ØP,   CLEAR2
2411  4477            LCMR
2412  45ØØ            LFGR
2413  45Ø1            LDBR
2414  45Ø5            RMSR
2415  ØØ2Ø            AND     K1
2416  4533            COMPAR
2417  7Ø63            ER1ØP
                /GD= GOOD IF; BD= REAL IF (BIT 11)
242Ø  521Ø            JMP     T1ØP            /SUBTEST LOOP

                /VERIFY 9 CHANNEL AND CM1Ø=Ø AND "GO" DO NOT SET IF.
                /("GO" SHOULD NOT GENERATE "PRESET" NOW.)



2421  4524      T1ØQ,   CLEAR2
2422  455Ø            GO                      /CM1Ø=Ø FROM CLEAR.
2423  45Ø5            RMSR
2424  ØØ2Ø            AND     K1
2425  4533            COMPAR
2426  7Ø65            ER1ØQ
                /GD= GOOD IF; BD= REAL IF (BIT 11)
2427  5221            JMP     T1ØQ            /SUBTEST LOOP.

                /VERIFY SPCREV WITHOUT BOT DOES NOT SET IF.
243Ø  4524      T1ØR,   CLEAR2
2431  4546            SPCREV
2432  45Ø5            RMSR
```

```
        2433  0020              AND     K1
        2434  4533              COMPAR
        2435  7067              ER10R
                     /GD= GOOD IF; BD= REAL IF (BIT 11)
        2436  5230              JMP     T10R            /SUBTEST LOOP.

        2437  4530              LOOP7
        2440  5752              JMP I   T10KP           /*****LOOP 7*****

        2441  4243              JMS     T10S            /EXECUTE T10S
        2442  5254              JMP     T10T            /THEN ONTO T10T.

                     /VERIFY "CLT"CLEARS READ/COMPARE ERROR.
        2443  0000   T10S,      0
        2444  3154              DCA     GOOD
        2445  4524              CLEAR2
        2446  4505              RMSR
        2447  0021              AND     K2
        2450  4533              COMPAR
        2451  7071              ER10S
                     /GD= GOOD R/C ERROR BIT; BD= REAL R/C ERROR (BIT 10)
        2452  5245              JMP     T10S+2          /SUBTEST LOOP
        2453  5643              JMP I   T10S

                     /VERIFY THAT DB=7777 AND MEM=0000 YIELDS AN R/C ERROR.
        2454  1021   T10T,      TAD     K2
        2455  3154              DCA     GOOD
        2456  3051              DCA     XBUFF+1         /MEM TO 0000
        2457  4524              CLEAR2
        2460  1047              TAD     XBUFFP          /MEM-1 TO CA
        2461  4475              LCAR
        2462  1165              TAD     PRGFLD          /FIELD TO CM.
        2463  4477              LCMR
        2464  7040              CMA                     /777 TO DB
        2465  4501              LDBR
        2466  4542              RDCOMP                  /R/C FUNC
        2467  4535              BREAK                   /1 BREAK
        2470  4505              RMSR                    /CHECK
        2471  0021              AND     K2
        2472  4533              COMPAR
        2473  7073              ER10T
                     /GD= GOOD R/C ERROR; BD= REAL R/C ERROR (BIT 10)
        2474  5257              JMP     T10T+3          /SUBTEST LOOP

        2475  4243              JMS     T10S
        2476  4300              JMS     T10U
        2477  5311              JMP     T10V


                     /VERIFY CLT CLEARS EMA OVERFLOW BIT.
        2500  0000   T10U,      0
        2501  3154              DCA     GOOD
        2502  4524              CLEAR2
           3  4507              RFSR
```

```
2504  ØØ23                 AND     K1Ø
2505  4533                 COMPAR
2506  7Ø75                 ER1ØU
                   /GD= GOOD EMA OVERFLOW; BD= REAL EMA OVERFLOW (BIT 8)
2507  53Ø2                 JMP     T1ØU+2          /SUBTEST LOOP
251Ø  57ØØ                 JMP I   T1ØU

                   /VERIFY EMA OVERFLOW BIT SETS WHEN EMA SET TO 7
                   /AND CA Ø CHANGES FROM 1 TO Ø,
2511  1Ø23         T1ØV,   TAD     K1Ø
2512  3154                 DCA     GOOD
2513  4524                 CLEAR2
2514  7Ø4Ø                 CMA                     /7777 TO CA
2515  4475                 LCAR
2516  4536                 COMAND                  /7 TO EMA
2517  ØØ7Ø                 7Ø
252Ø  4456                 JMS I   FUNCP           /"GO" AND EMA INCREMENT
2521  Ø14Ø                 Ø14Ø                    /ENABLE TO FUNC,
2522  4476                 CCAR                    /Ø TO CA,
2523  45Ø7                 RFSR                    /CHECK
2524  ØØ23                 AND     K1Ø
2525  4533                 COMPAR
2526  7Ø77                 ER1ØV
                   /GD= GOOD EMA OVERFLOW; BD= REAL EMA OVERFLOW (BIT 8)
2527  5313                 JMP     T1ØV+2          /SUBTEST LOOP

253Ø  43ØØ                 JMS     T1ØU

                   /VERIFY EMA SET TO 7 AND CA Ø CHANGING FROM Ø TO 1
                   /DOES NOT SET EMA OVERFLOW BIT,
2531  4524         T1ØW,   CLEAR2
2532  4476                 CCAR                    /Ø TO CA,
2533  4536                 COMAND                  /7 TO EMA
2534  ØØ7Ø                 7Ø
2535  4456                 JMS I   FUNCP           /"GO" AND EMA INCREMENT
2536  Ø14Ø                 Ø14Ø                    /ENABLE TO FUNC,
2537  7Ø4Ø                 CMA                     /7777 TO CA,
254Ø  4475                 LCAR
2541  45Ø7                 RFSR                    /CHECK
2542  ØØ23                 AND     K1Ø
2543  4533                 COMPAR
2544  71Ø1                 ER1ØW
                   /GD= GOOD EMA OVERFLOW BIT; BD= REAL EMA OVERFLOW (BIT 8)
2545  5331                 JMP     T1ØW            /SUBTEST LOOP,

2546  4531         T1ØLP8, LOOP8
2547  5237                 JMP     T1ØS-4          /*****LOOP 8*****

255Ø  5751                 JMP I   .+1
2551  26ØØ                 T1ØX-2
2552  2346         T1ØKP,  T1ØK-2

      26ØØ                 PAGE

26ØØ  42Ø2                 JMS     T1ØX
```

```
        26Ø1  5213              JMP     T1ØZ



                        /VERIFY CLT CLEARS EF,
        26Ø2  ØØØØ      T1ØX,   Ø
        26Ø3  3154              DCA     GOOD
        26Ø4  4524              CLEAR2
        26Ø5  45Ø5              RMSR
        26Ø6  ØØ34              AND     K4ØØØ
        26Ø7  4533              COMPAR
        261Ø  71Ø3              ER1ØX
                        /GD= GOOD EF; BD= REAL EF (BIT Ø)
        2611  52Ø4              JMP     T1ØX+2          /SUBTEST LOOP.
        2612  56Ø2              JMP I   T1ØX

                        /VERIFY R/C ERROR WITH MTTF SET SETS EF.
        2613  1Ø34      T1ØZ,   TAD     K4ØØØ
        2614  3154              DCA     GOOD
        2615  4524              CLEAR2
        2616  1Ø47              TAD     XBUFFP          /GENERATE R/C ERROR,
        2617  4475              LCAR
        262Ø  1165              TAD     PRGFLD
        2621  4477              LCMR
        2622  7Ø4Ø              CMA
        2623  45Ø1              LDBR
        2624  4542              RDCOMP
        2625  4535              BREAK
        2626  4545              SPCFWD                  /FORCE MTTF
        2627  4553              WCOV
        263Ø  45Ø1              LDBR                    /SET MTTF,
        2631  45Ø5              RMSR                    /CHECK,
        2632  ØØ34              AND     K4ØØØ
        2633  4533              COMPAR
        2634  71Ø5              ER1ØZ
                        /GD=GOOD EF; BD=REAL EF(BIT Ø)
        2635  5215              JMP     T1ØZ+2          /SUBTEST LOOP
        2636  42Ø2              JMS     T1ØX

                        /VERIFY EMA OVERFLOW AND MTTF SET SETS EF.
        2637  1Ø34      T1Ø1,   TAD     K4ØØØ
        264Ø  3154              DCA     GOOD
        2641  4524              CLEAR2
        2642  724Ø              CLA CMA                 /GENERATE EMA OVERFLOW
        2643  4475              LCAR
        2644  4536              COMAND
        2645  ØØ7Ø              7Ø
        2646  4456              JMS I   FUNCP
        2647  Ø14Ø              Ø14Ø
        265Ø  4476              CCAR
        2651  4545              SPCFWD                  /FORCE MTTF
        2652  4553              WCOV
        2653  45Ø5              RMSR                    /CHECK
        2654  ØØ34              AND     K4ØØØ
           5  4533              COMPAR
```

```
      2656  7107            ER1Ø1
                     /GD=GOOD EF; BD=REAL EF(BIT Ø)
      2657  5241            JMP     T1Ø1+2          /SUBTEST LOOP

                     /VERIFY EMA OVERFLOW WITHOUT MTTF SET DOES NOT SET EF,
      266Ø  3154     T1Ø2,  DCA     GOOD
      2661  4524            CLEAR2
      2662  724Ø            CLA CMA                 /FORCE EMA OVERFLOW,
      2663  4475            LCAR
      2664  4536            COMAND
      2665  ØØ7Ø            7Ø
      2666  4456            JMS I FUNCP

      2667  Ø14Ø            Ø14Ø
      267Ø  4476            CCAR
      2671  45Ø5            RMSR                    /CHECK FOR NO EF,
      2672  ØØ34            AND     K4ØØØ
      2673  4533            COMPAR
      2674  7111            ER1Ø2
                     /GD=GOOD EF; BD=REAL EF (BIT Ø)
      2675  5261            JMP     T1Ø2+1          /SUBTEST LOOP,


      2676  4532            LOOP9
      2677  57Ø5            JMP I   T1ØLPP          /*****LOOP9*****

      27ØØ  17Ø4            TAD I   TST1ØP
      27Ø1  33Ø3            DCA     .+2
      27Ø2  57Ø3            JMP I   .+1
      27Ø3  ØØØØ            Ø
      27Ø4  22ØØ     TST1ØP, TEST1Ø
      27Ø5  2546     T1ØLPP, T1ØLP8
```

```
                 /TEST11.  MTTF AND EF SKIP-NO SKIP TEST.

                 /1.  VERIFIES PROPER SKIP OR NO SKIP OF ALL 672X IOT'S
                 /WITH MTTF AND EF BOTH CLEAR, THEN EACH SET WITH THE OTHER CLEAR.

                 /2.  ALL IOT'S USED (672X) ARE SUPPLIED BY A LOCAL
                 /EXECUTIVE AND IF AN ERROR OCCURS THE IOT IN USE IS DISPLAYED
                 /UNDER"I1!" IN THE ERROR PRINTOUT.

                 /3.  ALL SUBTEST LOOPS CYCLE ON IOT IN PRESENT USE.

                 /4.  LOOPS 6-8!
                 /4.1  LOOP 6 CYCLES ON T11A-T11B (MTTF AND EF CLEAR)
                 /4.2  LOOP 7 CYCLES ON T11C-T11D (MTTF SET, EF CLEAR)
                 /4.3  LOOP 8 CYCLES ON T11E-T11F (MTTF SET, EF SET)


      3000                PAGE
3000  0000       TEST11,  0
3001  4523                CLEAR1
3002  3154                DCA     GOOD

                 /VERIFY SKCB SKIPS WHEN MTTF AND EF CLEAR.
3003  4524       T11A,    CLEAR2                  /CLEAR ALL.
3004  4513                SKCB                    /SHOULD SKIP.
3005  7040                CMA
3006  4533                COMPAR
3007  7113                ER11A
3010  5203                JMP     T11A            /SUBTEST LOOP

                 /VERIFY ALL 672X IOT'S WITH THE EXCEPTION OF SKCB DO NOT SKIP
                 /WHEN MTTF AND EF CLEAR.  IOT IS UNDER I1!.
3011  4322       T11B,    JMS     T11EX1          /GET AN IOT.
3012  3147                T11LS1+1
3013  7772                -6
3014  3016                T11B1
3015  4524                CLEAR2
3016  0000       T11B1,   0                       /INSERTED HERE AND SHOULD NOT
3017  7410                SKP                     /SKIP.
3020  7040                CMA
3021  4533                COMPAR
3022  7115                ER11B
                 /I1=IOT WHICH FAILED.
3023  5215                JMP     T11B+4          /SUBTEST LOOP
3024  4341                JMS     T11EX2

3025  4527                LOOP6
3026  5203                JMP     T11A            /*****LOOP6*****

                 /VERIFY SKCB AND SKTD SKIP WHEN MTTF SET AND EF CLEAR.
3027  4322       T11C,    JMS     T11EX1          /GET IOT.
3030  3146                T11LS1
3031  7776                -2
3032  3037                T11C1
```

```
        3Ø33  4524              CLEAR2
        3Ø34  4545              SPCFWD                  /FORCE MTTF
        3Ø35  4553              WCOV
        3Ø36  45Ø1              LDBR                    /SET MTTF,
        3Ø37  ØØØØ      T11C1,  Ø                       /IOT SHOULD SKIP,
        3Ø4Ø  7Ø4Ø              CMA
        3Ø41  4533              COMPAR
        3Ø42  7117              ER11C
                        /I1=FAILING IOT,
        3Ø43  5233              JMP     T11C+4          /SUBTEST LOOP,
        3Ø44  4341              JMS     T11EX2



                        /VERIFY SKTR,SDLE,SBRM AND CLF DO NOT SKIP WHEN
                        /MTTF SET AND EF SET,
        3Ø45  4322      T11D,   JMS     T11EX1          /GET AND IOT
        3Ø46  3151              T11LS1+3
        3Ø47  7774              -4
        3Ø5Ø  3Ø55              T11D1
        3Ø51  4524              CLEAR2
        3Ø52  4545              SPCFWD                  /FORCE MTTF,
        3Ø53  4553              WCOV
        3Ø54  45Ø1              LDBR                    /SET MTTF (ALSO SETS IF AND EF)
        3Ø55  ØØØØ      T11D1,  Ø                       /IOT SHOULD NOT SKIP,
        3Ø56  741Ø              SKP
        3Ø57  7Ø4Ø              CMA
        3Ø6Ø  4533              COMPAR
        3Ø61  7121              ER11D
                        /I1=FAILING IOT
        3Ø62  5251              JMP     T11D+4          /SUBTEST LOOP,
        3Ø63  4341              JMS     T11EX2
        3Ø64  453Ø              LOOP7
        3Ø65  5225              JMP     T11C-2          /*****LOOP7*****

                        /VERIFY SKEF,SKTD AND SKCB SKIP WHEN MTTF AND EF SET,
        3Ø66  4322      T11E,   JMS     T11EX1          /GET AN IOT
        3Ø67  3145              T11LS1-1
        3Ø7Ø  7775              -3
        3Ø71  3Ø74              T11E1
        3Ø72  4524              CLEAR2
        3Ø73  4357              JMS     SETEF           /SET MTTF AND EF,
        3Ø74  ØØØØ      T11E1,  Ø                       /IOT SHOULD SKIP,
        3Ø75  7Ø4Ø              CMA
        3Ø76  4533              COMPAR
        3Ø77  7123              ER11E
                        /I1=FAILING IOT,
        31ØØ  5272              JMP     T11E+4          /SUBTEST LOOP,
        31Ø1  4341              JMS     T11EX2

                        /VERIFY SKTR,SDLE,SBRM AND CLF DO NOT SKIP WHEN MTTF
                        /AND EF SET,
        31Ø2  4322      T11F,   JMS     T11EX1          /GET AN IOT,
        31Ø3  3151              T11LS1+3
```

```
31Ø4  7774            -4
31Ø5  311Ø            T11F1
31Ø6  4524            CLEAR2
31Ø7  4357            JMS      SETEF            /SET MTTF AND EF.
311Ø  ØØØØ    T11F1,  Ø                         /IOT SHOULD NOT SKIP.
3111  741Ø            SKP
3112  7Ø4Ø            CMA
3113  4533            COMPAR
3114  7125            ER11F
              /I1=FAILING IOT.
3115  53Ø6            JMP      T11F+4           /SUBTEST LOOP.
3116  4341            JMS      T11EX2

3117  4531            LOOP8
312Ø  5264            JMP      T11E-2  /*****LOOP8*****
3121  56ØØ            JMP I    TEST11

3122  ØØØØ    T11EX1, Ø                         /TEST 11 672X IOT SELECTOR.
3123  1722            TAD I    T11EX1           /GET T11LS1 STARTING POINTER -1.
3124  3Ø1Ø            DCA      1Ø               /PUT IN AUTO-INDEX.
3125  2322            ISZ      T11EX1
3126  1722            TAD I    T11EX1           /GET NEGATIVE NUMBER OF IOT'S.
3127  3157            DCA      TXXTM1           /SAVE IN TALLY.
313Ø  2322            ISZ      T11EX1
3131  1722            TAD I    T11EX1           /GET POINTER TO INSERT
3132  316Ø            DCA      TXXTM2           /SAVE
3133  2322            ISZ      T11EX1           /UPDATE TO RETURN.
3134  141Ø    T11EX3, TAD I    1Ø               /GET IOT.
3135  356Ø            DCA I    TXXTM2           /INSERT IN TEST.
3136  156Ø            TAD I    TXXTM2
3137  3745            DCA I    I1P              /INSERT IN IOT1 FOR TYPEOUT.
314Ø  5722            JMP I    T11EX1           /RUN TEST.

3141  ØØØØ    T11EX2, Ø                         /ENTER FROM END OF TEST.
3142  2157            ISZ      TXXTM1           /ALL IOT'S DONE?
3143  5334            JMP      T11EX3           /NO,  DO NEXT ONE.
3144  5741            JMP I    T11EX2           /YES EXIT PAST END OF TEST.

3145  Ø566    I1P,    IOT1
3146  4512    T11LS1, SKEF                      /IOT LIST FOR TEST 11.
3147  4513            SKCB
315Ø  4514            SKTD
3151  4512            SKEF
3152  4515            SKTR
3153  452Ø            SDLE
3154  4521            SBRM
3155  4516            CLF
3156  4514            SKTD

3157  ØØØØ    SETEF,  Ø                         /SET ERROR FLAG USING A RDCOMP
316Ø  3Ø51            DCA      XBUFF+1          /FAILURE WITH MTTF SET.
3161  724Ø            CLA CMA
3162  45Ø1            LDBR
3163  1Ø47            TAD      XBUFFP
3164  4475            LCAR
```

```
        3165  1165            TAD     PRGFLD
        3166  4477            LCMR
        3167  4542            RDCOMP
        317Ø  4535            BREAK
        3171  4545            SPCFWD
        3172  4553            WCOV
        3173  45Ø1            LDBR
        3174  5757            JMP I   SETEF
```

```
                /TEST 12,  MTTF-EF INTERRUPT TEST,

                        /1,  VERIFIES CORRECT INTERRUPT NON-OCCURRENCES AND OCCURRENCES
                /WITH ALL INTERESTING COMBINATIONS OF MTTF, EF AND THEIR ENABLES,

                        /2,  IF AN ERROR OCCURS, THE CM AND MS PRINTOUTS SHOW
                /THE CURRENT COMBINATION,

                        /3,  ALL SUBTEST LOOPS CYCLE ON CURRENT COMBINATIONS,

                        /4,  LOOPS 6-8:
                        /4,1  LOOP 6 CYCLES ON T12A (MTTF AND EF CLEAR)
                        /4,2  LOOP 7 CYCLES ON T12B-T12C (MTTF SET, EF CLEAR)
                        /4,3  LOOP 8 CYCLES ON T12D-T12E (MTTF SET, EF SET,)


      3200              PAGE
3200  0000      TEST12, 0
3201  4523              CLEAR1
3202  6002              IOF
3203  4460              JMS I   INSRSP
3204  3335              T12INT
3205  3335              T12INT
3206  3154              DCA     GOOD

                /VERIFY NO INTERRUPT WHEN EF AND MTTF CLEAR REGARDLESS OF STATE OF
                /INTERRUPT ENABLE BITS,
3207  4305      T12A,   JMS     T12EX1          /GET ENABLES,
3210  3336              T12LS1-1
3211  7775              -3
3212  3215              T12A1
3213  4525              CLEAR3
3214  4536              COMAND                  /LOAD CM WITH ENABLES,
3215  0000      T12A1,  0
3216  4326              JMS     T12EX4          /CHECK FOR NO INTERRUPT,
3217  4533              COMPAR
3220  7127              ER12A
                /CM= ENABLE BITS,
3221  5213              JMP     T12A+4          /SUBTEST LOOP,
3222  4322              JMS     T12EX2

3223  4527              LOOP6
3224  5206              JMP     T12A-1          /*****LOOP 6*****

                /VERIFY NO INTERRUPT WHEN MTTF AND EF SET, AND BOTH INTERRUPT ENABLES CLEAR,
3225  3154      T12B,   DCA     GOOD
3226  4525              CLEAR3                  /CLEAR ALL,
3227  4545              SPCFWD                  /FORCE MTTF
3230  4553              WCOV
3231  4501              LDBR
3232  4536              COMAND                  /CLEAR ENABLES
  33  0000              0
  34  4326              JMS     T12EX4          /CHECK   TERRUPT
```

```
3235  4533             COMPAR
3236  7131             ER12B
3237  5226             JMP     T12B+1          /SUBTEST LOOP

                /VERIFY INTERRUPT WITH MTTF SET AND MTTF INT ENABLE SET.
324Ø  724Ø      T12C,  CLA CMA
3241  3154             DCA     GOOD
3242  4525             CLEAR3                  /CLEAR ALL.
3243  4545             SPCFWD                  /FORCE MTTF
3244  4553             WCOV
3245  45Ø1             LDBR
3246  4536             COMAND                  /LOAD ENABLE.
3247  Ø1ØØ             1ØØ
325Ø  4326             JMS     T12EX4          /CHECK INTERRUPT.
3251  4533             COMPAR
3252  7133             ER12C
3253  5242             JMP     T12C+2          /SUBTEST LOOP

3254  453Ø             LOOP7
3255  5223             JMP     T12B-2          /*****LOOP 7*****


                /VERIFY NO INTERRUPT WITH EF AND MTTF SET, AND ENABLES CLEAR.
3256  3154      T12D,  DCA     GOOD
3257  4525             CLEAR3                  /CLEAR ALL
326Ø  4745             JMS I   SETEFP          /FORCE MTTF AND EF.
3261  4536             COMAND                  /CLEAR ENABLES.
3262  ØØØØ             Ø
3263  4326             JMS     T12EX4          /CHECK INTERRUPT
3264  4533             COMPAR
3265  7135             ER12D
3266  5257             JMP     T12D+1          /SUBTEST LOOP

                /VERIFY INTERRUPT WHEN EF AND EF INT ENABLE SET.
3267  724Ø      T12E,  CLA CMA
327Ø  3154             DCA     GOOD
3271  4525             CLEAR3                  /CLEAR ALL.
3272  4745             JMS I   SETEFP          /FORCE EF.
3273  4536             COMAND                  /SET EF INT ENABLE.
3274  Ø2ØØ             2ØØ
3275  4326             JMS     T12EX4          /CHECK INTERRUPT
3276  4533             COMPAR
3277  7137             ER12E
33ØØ  5271             JMP     T12E+2          /SUBTEST LOOP

33Ø1  4531             LOOP8
33Ø2  5254             JMP     T12D-2          /*****LOOP 8*****

33Ø3  4523             CLEAR1                  /EXIT
33Ø4  56ØØ             JMP I   TEST12

33Ø5  ØØØØ      T12EX1, Ø                      /TEST 12 COMMAND REGISTER CONSTANT
33Ø6  17Ø5             TAD I   T12EX1          /SELECTOR,  SAME BASIC METHOD
33Ø7  3Ø1Ø             DCA     1Ø
```

```
3310  2305            ISZ     T12EX1
3311  1705            TAD I   T12EX1
3312  3157            DCA     TXXTM1
3313  2305            ISZ     T12EX1
3314  1705            TAD I   T12EX1
3315  3160            DCA     TXXTM2
3316  2305            ISZ     T12EX1

3317  1410    T12EX3, TAD I   1Ø
3320  3560            DCA I   TXXTM2
3321  5705            JMP I   T12EX1
3322  ØØØØ    T12EX2, Ø
3323  2157            ISZ     TXXTM1
3324  5317            JMP     T12EX3
3325  5722            JMP I   T12EX2
3326  ØØØØ    T12EX4, Ø                         /PUT Ø'S IN INTERRUPT INDICATOR,
3327  72ØØ            CLA                       /(THE AC,)
333Ø  6ØØ1            ION                       /INTERRUPT ON,
3331  7ØØØ            NOP                       /WAIT,
3332  7ØØØ            NOP
3333  6ØØ2            IOF                       /INTERRUPT OFF,
                                                /GET INDICATOR AND
3334  5726            JMP I   T12EX4            /EXIT,
3335  724Ø    T12INT, CLA CMA                   /MTTF, EF INT, HANDLER FOR TEST 12,
3336  5726            JMP I   T12EX4            /JUST SET AC=7777,
3337  ØØØØ    T12LS1, Ø                         /NO INT ENABLE,
334Ø  Ø1ØØ            1ØØ                       /MTTF INT ENABLE
3341  Ø2ØØ            2ØØ                       /EF INT ENABLE,
3342  ØØØØ            Ø
3343  Ø2ØØ            2ØØ
3344  Ø1ØØ            1ØØ
3345  3157    SETEFP, SETEF
```

```
                    /TEST 13,  672X IOT UNIQUENESS FOR CLF AND SBRM,

                            /1,  VERIFIES THAT CLF CLEARS ONLY STATUS BITS AND NOT NON-
                    /STATUS REGISTERS (SINCE TUR HIGH) AND THAT NO OTHER 672X IOT
                    /GENERATES CLF OR SRBM,

                            /2,  PORTIONS OF THIS TEST ARE RUN BY TEST 11 LOCAL EXECUTIVE
                    /IN WHICH CASE THE IOT IN USE IS DISPLAYED UNDER "I1:" IF
                    /AN ERROR OCCURS,

                            /3,  LOOPS 6-7:
                            /3,1  LOOP 6 CYCLES ON T13A-T13C (CLF CHECKS)
                            /3,2  LOOP 7 CYCLES ON T13D-T13E (SBRM UNIQUENESS)

      34ØØ                  PAGE
34ØØ  ØØØØ          TEST13, Ø
34Ø1  4523                  CLEAR1

                    /USING MTTF, VERIFY CLF CLEARS STATUS, MTTF AND EF,  THIS
                    /IS ENOUGH TO VERIFY THAT CLF IS WORKING,
34Ø2  3154          T13A,   DCA     GOOD
34Ø3  4545                  SPCFWD                  /FORCE MTTF
34Ø4  4553                  WCOV
34Ø5  45Ø1                  LDBR
34Ø6  4516                  CLF                     /SHOULD CLEAR MTTF
34Ø7  4514                  SKTD                    /CHECK
341Ø  741Ø                  SKP
3411  7Ø4Ø                  CMA
3412  4533                  COMPAR
3413  7141                  ER13A
3414  52Ø3                  JMP     T13A+1          /SUBTEST LOOP,

                    /VERIFY IOT 672X EXCEPT CLF DOES NOT CLEAR MTTF,
3415  47Ø3          T13B,   JMS I   T13EX1          /GET AN IOT
3416  3146                  T11LS1
3417  7772                  -6
342Ø  3424                  T13B1
3421  4545                  SPCFWD                  /FORCE MTTF
3422  4553                  WCOV
3423  45Ø1                  LDBR
3424  ØØØØ          T13B1,  Ø                       /EXECUTE IOT,  SHOULD NOT CLEAR
3425  7ØØØ                  NOP                     /MTTF,
3426  4514                  SKTD                    /CHECK,
3427  7Ø4Ø                  CMA
343Ø  4533                  COMPAR
3431  7143                  ER13B
                    /I1= FAILING IOT
3432  5221                  JMP     T13B+4          /SUBTEST LOOP
3433  47Ø4                  JMS I   T13EX2

                    /VERIFY THAT CLF DOES NOT CLEAR CA (OR ANY OTHER NON-STATUS
                    /REGISTER) SINCE "TUR" IS HIGH,
3434  724Ø          T13C,   CLA CMA
3435  3154                  DCA     GOOD
```

```
3436  724Ø            CLA CMA                 /7777 TO CA.
3437  4475            LCAR
344Ø  4516            CLF                     /SHOULD NOT CHANGE CA,
3441  45Ø4            RCAR                    /CHECK
3442  4533            COMPAR
3443  7145            ER13C
                /GD= GOOD CA
3444  5236            JMP     T13C+2          /SUBTEST LOOP.

3445  4527            LOOP6
3446  52Ø2            JMP     T13A            /*****LOOP 6*****



                /VERIFY CLF DOES NOT CAUSE A BREAK REQUEST.  CHECK VIA CA
                /INCREMENT.
3447  724Ø      T13D,   CLA CMA
345Ø  3154            DCA     GOOD
3451  724Ø            CLA CMA                 /7777 TO CA
3452  4475            LCAR
3453  4543            WRITE                   /WRITE FUNCTION.
3454  4516            CLF                     /SHOULD NOT CAUSE A BREAK
3455  45Ø4            RCAR                    /AND CA SHOULD REMAIN UNCHANGED.
3456  4533            COMPAR
3457  7147            ER13D
                /GD= GOOD CA
346Ø  5251            JMP     T13D+2          /SUBTEST LOOP.

                /VERIFY SKCB, SKTD, SKEF, SKTR OR SDLE DO NOT GENERATE
                /BREAK REQUEST.  CHECK VIA WC INCREMENT.
3461  47Ø3      T13E,   JMS I   T13EX1          /GET AN IOT.
3462  3146            T11LS1
3463  7773            -5
3464  3471            T13E1
3465  4524            CLEAR2                  /CLEAR ALL.
3466  724Ø            CLA CMA                 /7777 TO WC.
3467  4473            LWCR
347Ø  4545            SPCFWD                  /SET SPACE FORWARD.
3471  ØØØØ      T13E1,  Ø                       /IOT SHOULD NOT CAUSE BREAK
3472  7ØØØ            NOP
3473  45Ø3            RWCR                    /WE SHOULD STILL BE 7777.
3474  4533            COMPAR
3475  7151            ER13E
                /I1= FAILING IOT; GD= GOOD WC.
3476  5265            JMP     T13E+4          /SUBTEST LOOP
3477  47Ø4            JMS I   T13EX2

35ØØ  453Ø            LOOP7
35Ø1  5245            JMP     T13D-2          /*****LOOP 7*****

35Ø2  56ØØ            JMP I   TEST13
35Ø3  3122      T13EX1, T11EX1
35Ø4  3141      T13EX2, T11EX2
```

```
                  /TEST 14,  FULL READ-COMPARE TEST,

                          /1,  VERIFIES:
                          /1.1  NO READ-COMPARE ERROR WHEN MEM AND DB CONTAIN SAME
                  /DATA,
                          /1.2  READ-COMPARE ERROR IS GENERATED WHEN MEM AND DB
                  /DATA DIFFER BY ONLY ONE BIT,
                          /1.3  CA DOES NOT INCREMENT WHEN R-C ERROR EXISTS,

                          /2,  TESTS T14A AND T14B ARE RUN BY A LOCAL EXECUTIVE
                  /WHICH GENERATES DATA FOR MEM AND DB, AND ALSO RUNS THE PORTION
                  /OF THE TEST WHICH SETS UP CA, DB, RDCOMP FUNCTION, THE BREAK
                  /REQUEST, THEN READS MS AND MASKS TO THE R/C BIT,

                          /3,  ALL SUBTEST LOOPS CYCLE ON THE CURRENT DATA,

                          /4,  LOOPS 6-8:
                          /4.1  LOOP 6 CYCLES ON T14A (NO R-C ERROR)
                          /4.2  LOOP 7 CYCLES ON T14B (R-C ERROR)
                          /4.3  LOOP 8 CYCLES ON T14C (CA NO INCREMENT)

      36ØØ                PAGE
36ØØ  ØØØØ        TEST14, Ø
36Ø1  4523                CLEAR1

                  /VERIFY NO READ-COMPARE ERROR WHEN MEM AND DB CONTAIN SAME DATA,
                  /COUNT PATTERN USED.
36Ø2  3154        T14A,   DCA     GOOD
36Ø3  4271                JMS     T14EX1
36Ø4  4533                COMPAR
36Ø5  7153                ER14A
                  /GD=GOOD MS MASKED; BD=REAL MS MASKED; OD=DATA IN MEM,
36Ø6  5277                JMP     T14EX2          /SUBTEST LOOP,
36Ø7  43Ø1                JMS     T14EX3

361Ø  4527                LOOP6
3611  52Ø2                JMP     T14A            /*****LOOP6*****

                  /VERIFY READ-COMPARE ERROR IS GENERATED WHEN ONE BIT DIFFERS IN DB
                  /AND MEM,
3612  1Ø21        T14B,   TAD     K2              /VERIFY READ-COMPARE ERROR
3613  3154                DCA     GOOD            /WHEN ONE BIT DIFFERS IN DB
3614  431Ø                JMS     T14EX4          /AND MEM,
3615  4533                COMPAR                  /DATA IN DB UNDER "DB:",
3616  7155                ER14B
                  /GD=GOOD MS MASKED; BD=REAL MS MASKED; OD=DATA IN MEM; DB=REAL DB
3617  5325                JMP     T14EX5          /SUBTEST LOOP.
362Ø  4327                JMS     T14EX6

3621  453Ø                LOOP7
3622  521Ø                JMP     T14B-2          /*****LOOP7*****

                  /VERIFY CA DOES NOT INCREMENT WHEN READ-COMPARE ERROR EXISTS,
   3  1Ø47        T14C,   TAD     XBUFFP
```

```
      3624  3154              DCA     GOOD
      3625  724Ø              CLA CMA
      3626  3Ø51              DCA     XBUFF+1
      3627  4524              CLEAR2
      363Ø  45Ø1              LDBR                      /GENERATE R/C ERROR,
      3631  1Ø47              TAD     XBUFFP
      3632  4475              LCAR
      3633  1165              TAD     PRGFLD
      3634  4477              LCMR
      3635  4542              RDCOMP
      3636  4535              BREAK
      3637  1Ø47      T14C1,  TAD     XBUFFP            /SET UP FOR NEXT CA INC
      364Ø  4475              LCAR                      /WITH R/C ERROR EXISTING,
      3641  724Ø              CLA CMA
      3642  45Ø1              LDBR
      3643  4535              BREAK
      3644  45Ø4              RCAR
      3645  4533              COMPAR
      3646  7157              ER14C
                      /GD=GOOD CA,
      3647  5237              JMP     T14C1   /SUBTEST LOOP,


                      /VERIFY THAT CM1Ø-11 INHIBIT READ-COMPARE CHECKS ON BITS
                      /Ø-3 OF DB AND MEM. 74ØØ IN MEM, ØØØØ IN DB,  THERE SHOULD BE NO
                      /READ-COMPARE ERROR,
      365Ø  1Ø36      T14D,   TAD     K74ØØ
      3651  3Ø51              DCA     XBUFF+1
      3652  3154              DCA     GOOD
      3653  1262              TAD     .+7
      3654  336Ø              DCA     T14EX8
      3655  4524              CLEAR2
      3656  1Ø47              TAD     XBUFFP
      3657  4475              LCAR
      366Ø  1ØØ5              TAD     K3
      3661  5366              JMP     T14EX8+6
      3662  3663              .+1
      3663  4533              COMPAR
      3664  7161              ER14D
                      /GD=GOOD R/C ERROR BIT; BD=REAL R/C ERROR BIT
      3665  5255              JMP     T14D+5            /SUBTEST LOOP,

      3666  4531              LOOP8
      3667  5221              JMP     T14C-2            /*****LOOP8*****

      367Ø  56ØØ              JMP I   TEST14

      3671  ØØØØ      T14EX1, Ø                         /ENTER FROM T14A,
      3672  4347              JMS     T14EX7            /INITIALIZE DATA AND LOOP COUNTERS,
      3673  1156              TAD     OLD               /PUT OLD IN MEM,
      3674  3Ø51              DCA     XBUFF+1
      3675  1156              TAD     OLD               /PUT OLD IN LOC USED TO LOAD DB,
      3676  316Ø              DCA     TXXTM2
      3677  436Ø      T14EX2, JMS     T14EX8            /SET UP CA, DB, RDCOMP, BREAK AND
      37ØØ  5671              JMP I   T14EX1            /LOOK AT R/C BIT IN MS THEN GO TO
```

```
                                        /TEST 14A COMPAR SECTION,
3701  ØØØØ    T14EX3, Ø                 /ENTER FROM END OF T14A,
3702  2156            ISZ     OLD       /UPDATE OLD DATE,
3703  53Ø7            JMP     .+4       /IF OLD NOT=Ø RUN NEW OLD,
3704  2375            ISZ     T14TM1    /IF OLD=Ø, HAS IT BEEN Ø TWICE?
3705  741Ø            SKP               /NO,
37Ø6  57Ø1            JMP I   T14EX3    /YES,  EXIT T14A,
37Ø7  5273            JMP     T14EX1+2  /NO,  RUN T14A AGAIN WITH NEW DATA,
371Ø  ØØØØ    T14EX4, Ø                 /ENTER FROM T14B,
3711  4347            JMS     T14EX7    /INITIALIZE DATA AND LOOP COUNTERS,
3712  1156            TAD     OLD       /PUT OLD IN MEM,
3713  3Ø51            DCA     XBUFF+1
3714  1156            TAD     OLD       /COMPLEMENT THE BIT IN OLD
3715  1377            TAD     T14BIT    /INDICATED BY CURRENT BIT IN
3716  316Ø            DCA     TXXTM2    /T14BIT AND PUT MODIFIED OLD
3717  1156            TAD     OLD       /IN LOC USED TO LOAD DB,
372Ø  Ø377            AND     T14BIT    /(COMPLEMENTING THE BIT IS
3721  71Ø4            CLL RAL           /ACTUALLY AN EXCLUSIVE OR FUNCTION,)
3722  7Ø41            CIA
3723  116Ø            TAD     TXXTM2
3724  316Ø            DCA     TXXTM2

3725  436Ø    T14EX5, JMS     T14EX8    /SET UP CA, DB, RDCOMP, BREAK, AND LOOK
3726  571Ø            JMP I   T14EX4    /AT R/C BIT IN MS, THEN GO TO T14B COMPAR,
3727  ØØØØ    T14EX6, Ø                 /ENTER FROM END OF T14B,
373Ø  2376            ISZ     T14TM2    /ALL 12 BITS BEEN COMPLEMENTED
3731  741Ø            SKP               /ONE AT A TIME?
3732  5337            JMP     .+5
3733  1377            TAD     T14BIT    /NO,  MOVE BIT ONE LEFT,
3734  71Ø4            CLL RAL
3735  3377            DCA     T14BIT
3736  5314            JMP     T14EX4+4  /GO BACK AND COMPUTE NEW DB DATA,
3737  7ØØ1            IAC               /YES,  PUT BITIN POSITION 12,
374Ø  3377            DCA     T14BIT
3741  2156            ISZ     OLD       /OLD + 1 TO OLD, OLD = Ø?
3742  5346            JMP     .+4
3743  2375            ISZ     T14TM1    /YES, HAS IT BEEN Ø TWICE?
3744  741Ø            SKP
3745  5727            JMP I   T14EX6    /YES, EXIT T14B,
3746  5355            JMP     T14EX7+6  /NO, RUN COMPLEMENTING BITS ON OLD,
3747  ØØØØ    T14EX7, Ø                 /INITIALIZE LOOP COUNTERS AND
375Ø  1Ø37            TAD     M2        /DATA, SET UP FOR 2 RUNS
3751  3375            DCA     T14TM1    /FOR OLD (8192 DATA SETS),
3752  3156            DCA     OLD       /SET OLD TO Ø FOR A STARTER,
3753  7ØØ1            IAC               /SET BIT TO POSITION 12 FOR
3754  3377            DCA     T14BIT    /STARTER
3755  1Ø42            TAD     M14       /SET UP FOR 12 BIT-ROTATES
3756  3376            DCA     T14TM2    /FOR USE IN T14B ONLY,
3757  5747            JMP I   T14EX7    /EXIT
376Ø  ØØØØ    T14EX8, Ø                 /T14A AND T14B TEST POSITION
3761  4524            CLEAR2            /GENERATE "INITIALIZE,"
3762  116Ø            TAD     TXXTM2    /GET DB DATA AND LOAD DB,
3763  45Ø1            LDBR
 764  1Ø47            TAD     XBUFFP    /PUT XBUFF IN CA,
  65  4475            LCAR
```

```
      3766  1165            TAD     PRGFLD
      3767  4477            LCMR
      3770  4542            RDCOMP                  /SET RDCOMP FUNCTION,
      3771  4535            BREAK                   /ONE BREAK REQUEST
      3772  4505            RMSR                    /READ MS AND MASK TO SEE
      3773  ØØ21            AND     K2              /R/C ERROR BIT ONLY
      3774  576Ø            JMP I   T14EX8          /GO TO COMPAR PORTION OF TEST
      3775  ØØØØ    T14TM1, Ø
      3776  ØØØØ    T14TM2, Ø
      3777  ØØØØ    T14BIT, Ø
```

```
                        /TM8E IOT SUBROUTINES,

                                /ALL TM8E IOT'S ARE PLACED IN SUBROUTINES
                                /TO ENABLE EASE IN CHANGING THE IOT DEVICE CODE
                                /SHOULD THE DEVICE CODES NOT BE 7Ø-72,
                                /IF AN IOT SKIPS WHICH SHOULD NEVER SKIP, IT RESULTS
                                /IN ERØØC,

        4ØØØ                    PAGE
  4ØØØ  ØØØØ    S67ØØ,          Ø
  4ØØ1  67ØØ                    67ØØ                    /LOAD IOT,
  4ØØ2  56ØØ                    JMP I   S67ØØ
  4ØØ3  4347                    JMS     SKIPER
  4ØØ4  ØØØØ    SLWCR,          Ø
  4ØØ5  67Ø1                    67Ø1                    /LWCR
  4ØØ6  56Ø4                    JMP I   SLWCR
  4ØØ7  4347                    JMS     SKIPER
  4Ø1Ø  ØØØØ    SCWCR,          Ø
  4Ø11  67Ø2                    67Ø2                    /CWCR
  4Ø12  561Ø                    JMP I   SCWCR
  4Ø13  4347                    JMS     SKIPER
  4Ø14  ØØØØ    SLCAR,          Ø
  4Ø15  67Ø3                    67Ø3                    /LCAR
  4Ø16  5614                    JMP I   SLCAR
  4Ø17  4347                    JMS     SKIPER
  4Ø2Ø  ØØØØ    SCCAR,          Ø
  4Ø21  67Ø4                    67Ø4                    /CCAR
  4Ø22  562Ø                    JMP I   SCCAR
  4Ø23  4347                    JMS     SKIPER
  4Ø24  ØØØØ    SLCMR,          Ø
  4Ø25  67Ø5                    67Ø5                    /LCMR
  4Ø26  5624                    JMP I   SLCMR
  4Ø27  4347                    JMS     SKIPER
  4Ø3Ø  ØØØØ    SLFGR,          Ø
  4Ø31  67Ø6                    67Ø6                    /LFGR
  4Ø32  563Ø                    JMP I   SLFGR
  4Ø33  4347                    JMS     SKIPER
  4Ø34  ØØØØ    SLDBR,          Ø
  4Ø35  67Ø7                    67Ø7                    /LDBR
  4Ø36  5634                    JMP I   SLDBR
  4Ø37  4347                    JMS     SKIPER
  4Ø4Ø  ØØØØ    S671Ø,          Ø
  4Ø41  671Ø                    671Ø                    /READ IOT,
  4Ø42  564Ø                    JMP I   S671Ø
  4Ø43  4347                    JMS     SKIPER
  4Ø44  ØØØØ    SRWCR,          Ø
  4Ø45  6711                    6711                    /RWCR
  4Ø46  5644                    JMP I   SRWCR
  4Ø47  4347                    JMS     SKIPER
  4Ø5Ø  ØØØØ    SRCAR,          Ø
  4Ø51  6713                    6713                    /RCAR
  4Ø52  565Ø                    JMP I   SRCAR
  4Ø53  4347                    JMS     SKIPER
  4Ø54  ØØØØ    SRMSR,          Ø
```

```
4Ø55 6714           6714                /RMSR
4Ø56 5654           JMP I   SRMSR
4Ø57 4347           JMS     SKIPER
4Ø6Ø ØØØØ   SRCMR,  Ø
4Ø61 6715           6715                /RCMR
4Ø62 566Ø           JMP I   SRCMR
4Ø63 4347           JMS     SKIPER
4Ø64 ØØØØ   SRFSR,  Ø
4Ø65 6716           6716                /RFSR
4Ø66 5664           JMP I   SRFSR
4Ø67 4347           JMS     SKIPER
4Ø7Ø ØØØØ   SRDBR,  Ø
4Ø71 6717           6717                /RDBR
4Ø72 567Ø           JMP I   SRDBR
4Ø73 4347           JMS     SKIPER
4Ø74 ØØØØ   S672Ø,  Ø
4Ø75 672Ø           672Ø                /CONTROL IOT,
4Ø76 5674           JMP I   S672Ø
4Ø77 4347           JMS     SKIPER
41ØØ ØØØØ   SSKEF,  Ø
41Ø1 6721           6721                /SKEF
41Ø2 57ØØ           JMP I   SSKEF
41Ø3 23ØØ           ISZ     SSKEF
41Ø4 57ØØ           JMP I   SSKEF
41Ø5 ØØØØ   SSKCB,  Ø
41Ø6 6722           6722                /SKCB
41Ø7 57Ø5           JMP I   SSKCB
411Ø 23Ø5           ISZ     SSKCB
4111 57Ø5           JMP I   SSKCB
4112 ØØØØ   SSKTD,  Ø
4113 6723           6723                /SKTD
4114 5712           JMP I   SSKTD
4115 2312           ISZ     SSKTD
4116 5712           JMP I   SSKTD
4117 ØØØØ   SSKTR,  Ø
412Ø 6724           6724                /SKTR
4121 5717           JMP I   SSKTR
4122 2317           ISZ     SSKTR
4123 5717           JMP I   SSKTR
4124 ØØØØ   SCLF,   Ø
4125 6725           6725                /CLF
4126 5724           JMP I   SCLF
4127 4347           JMS     SKIPER
413Ø ØØØØ   SCLT,   Ø                   /CLT
4131 6712           6712
4132 573Ø           JMP I   SCLT
4133 4347           JMS     SKIPER
4134 ØØØØ   SSDLE,  Ø
4135 6726           6726                /SDLE
4136 5734           JMP I   SSDLE
4137 4347           JMS     SKIPER
414Ø ØØØØ   SSBRM,  Ø
4141 6727           6727                /SBRM
4142 574Ø           JMP I   SSBRM
4143 4347           JMS     SKIPER
```

```
4144  ØØØØ    SCDF,   Ø
4145  62Ø1    SCDFI,  CDF     /N              /HOMING,
4146  5744            JMP I   SCDF


4147  ØØØØ    SKIPER, Ø                       /AN IOT SKIPPED WHICH SHOULD NEVER SKIP,
415Ø  317Ø            DCA     ACLOC           /SAVE AC AT TIME OF FAILURE,
4151  1Ø4Ø            TAD     M4              /GET POINTER TO IOT SURROUTINE,
4152  1347            TAD     SKIPER          /ENTRY AND SAVE,
4153  3347            DCA     SKIPER
4154  7ØØ1            IAC                     /GET POINTER TO IOT CODE,
4155  1347            TAD     SKIPER
4156  3155            DCA     BAD
4157  1555            TAD I   BAD             /GET IOT CODE AND SAVE,
416Ø  3155            DCA     BAD
4161  1747            TAD I   SKIPER          /GET RETURN TO MAIN PROGRAM,
4162  3347            DCA     SKIPER          /SAVE FOR EXIT,
4163  1155            TAD     BAD             /PUT IOT IN EXTERNAL SCOPE LOOP,
4164  3375            DCA     ,+11
4165  4534            ERROR                   /GO TO ERRORS,
4166  67Ø3            ERØØC
              /AN IOT SKIPPED WHICH SHOULD NEVER SKIP,
              /BD=FAILING IOT CODE; AC=CONTENTS OF AC AT TIME OF FAILURE,
4167  5374            JMP     ,+5             /EXECUTE IF SR5=1,
417Ø  4526            LOOP5                   /IF NOT CHECK SR5,
4171  5374            JMP     ,+3             /EXECUTE IF SR5=1,
4172  117Ø            TAD     ACLOC           /NO SCOPE LOOP, EXIT TO CORRECT
4173  5747            JMP I   SKIPER          /POINT IN MAIN PROGRAM,
4174  117Ø            TAD     ACLOC           /SCOPE LOOP, SET UP AC, THIS
4175  ØØØØ            Ø                       /MAY OR MAY NOT BE CORRECT CONDITION,
4176  537Ø            JMP     ,-6             /IOT CODE, NO FAILURE, CHECK SR5,
4177  5365            JMP     ,-12            /FAILURE, CALL ERRORS,
                        /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+1 IN OCTAL.)
                                /"TSTAT" IS AUTOMATICALLY SET TO 7777 WHEN THE PROGRAM IS
                                /STARTED AT 2ØØ, WITH SRØ=Ø.  IF SRØ=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.

              42ØØ              PAGE

        42ØØ  6224      EXEC,   RIF                     /GET INSTRUCTION FIELD.
        42Ø1  3165              DCA     PRGFLD          /MODIFY HOMING CDF ROUTINE.
        42Ø2  1165              TAD     PRGFLD
        42Ø3  1Ø46              TAD     KCDF
        42Ø4  375Ø              DCA I   SCDFIP
        42Ø5  62Ø1              CDF     ØØ              /SET UP INTERRUPT LINKAGE.
        42Ø6  1347              TAD     KRMF            /PUT RMF IN 1.
        42Ø7  3751              DCA I   P1
        421Ø  1344              TAD     KJMPI3          /JMP I 3 IN 2.
        4211  3752              DCA I   P2
        4212  1345              TAD     INTSEP          /INTSEV IN 3.
        4213  3753              DCA I   P3
        4214  4457              JMS I   SCDFP           /SET BACK TO CURRENT FIELD.
        4215  3174              DCA     EXITFL          /CLEAR ERROR EXIT FLAG.
        4216  724Ø              CLA CMA                 /SET THE TTO SOFT FLAG.
        4217  3173              DCA     TTOFLG
        422Ø  743Ø              SZL
        4221  534Ø              JMP     NOEXLP+2
        4222  4461              JMS I   AMG8EP          /PRINT TITLE MESSAGE
        4223  7163              MSG1
        4224  724Ø              CLA CMA                 /SET TEST IN PROGRESS FLAG
        4225  3175              DCA     ACTFLG
        4226  4471              JMS I   EXECFP          /FIND OUT HOW MANY MEM FIELDS
        4227  3177              DCA     TSTNUM
        423Ø  76Ø4              LAS                     /GET SRØ AND
        4231  3157              DCA     TXXTM1          /SAVE.
        4232  4461              JMS I   AMG8EP          /TELL USER THAT SRØ IS NOW A
        4233  7255              MSG9                    /QUICK VERIFY SWITCH
        4234  1157              TAD     TXXTM1          /GET OLD SRØ
        4235  771Ø              SPA CLA                 /GO DIRECTLY TO MONITOR?
        4236  534Ø              JMP     NOEXLP+2        /YES.
        4237  724Ø              CLA CMA                 /NO, SET UP TO RUN ALL TESTS.
        424Ø  3176      REX,    DCA     TSTAT           /PUT AC IN TEST STATUS.
        4241  3174              DCA     EXITFL          /CLEAR ERROR EXIT FLAG
        4242  724Ø              CLA CMA                 /SET TEST IN PROGRESS
        4243  3175              DCA     ACTFLG
        4244  3346              DCA     PASCNT          /CLEAR PASS COUNTER
        4245  1Ø42      EXECL2, TAD     M14             /SET TO CHECK FOR 12 TESTS
        4246  3354              DCA     EXTAL
```

```
      4247  1365             TAD      TSTP            /GET TEST STARTING ADDS, TABLE POINTER
      425Ø  3355             DCA      TSTPP
      4251  1367             TAD      TALP            /GET TEST TALLY TABLE POINTER
      4252  3356             DCA      TALPP
      4253  3177             DCA      TSTNUM          /SET TEST NUMBER TO ØØ
      4254  1176             TAD      TSTAT           /GET TEST STATUS AND PUT IN
      4255  3357             DCA      TSTAEX          /TEMPS,
      4256  1357    EXECL1,  TAD      TSTAEX          /CHECK FOR A TEST BIT SET
      4257  7ØØ4             RAL
      426Ø  3357             DCA      TSTAEX          /SAVE TEST STATUS ROTATED.
      4261  2355             ISZ      TSTPP           /UPDATE POINTERS
      4262  2356             ISZ      TALPP
      4263  2177             ISZ      TSTNUM          /UPDATE TEST NUMBER
      4264  742Ø             SNL                      /RUN THIS TEST?
      4265  5315             JMP      NOTSRN          /NO
      4266  76Ø4             LAS                      /YES TEST PROGRESS REPORT?
      4267  ØØ33             AND      K2ØØØ
      427Ø  764Ø             SZA CLA
      4271  5277             JMP      NOTSPR          /NO
      4272  1177             TAD      TSTNUM          /YES, GET TEST NUMBER
      4273  4463             JMS I    EDITP           /INSERT IN MSG AND PRINT
      4274  376Ø             DCA I    M2P1P
      4275  4461             JMS I    AMG8EP
      4276  7213             MSG2

      4277  1756    NOTSPR,  TAD I    TALPP           /GET THIS TESTS TALLY AND SAVE
      43ØØ  3361             DCA      TSTTAL
      43Ø1  1755             TAD I    TSTPP           /GET TEST STARTING ADDRESS
      43Ø2  3362             DCA      EXTEMP
      43Ø3  4762    EXOMT,   JMS I    EXTEMP          /RUN TEST
      43Ø4  76Ø4             LAS                      /LOOP ON CURRENT TEST
      43Ø5  ØØ21             AND      K2
      43Ø6  764Ø             SZA CLA
      43Ø7  53Ø3             JMP      EXOMT           /YES RUN TEST AGAIN
      431Ø  76Ø4             LAS                      /SRØ SET? (QUICK VERIFY)
      4311  771Ø             SPA CLA                  /NO, GO CHECK TEST TALLY
      4312  5315             JMP      NOTSRN          /YES SEE IF ALL TESTS RUN
      4313  2361             ISZ      TSTTAL          /NO DECREASE TALLY, TEST DONE?
      4314  53Ø3             JMP      EXOMT           /NO RUN AGAIN
      4315  2354    NOTSRN,  ISZ      EXTAL           /YES 12 TESTS CHECKED?
      4316  5256             JMP      EXECL1          /NO CHECK FOR NEXT TEST
      4317  3177             DCA      TSTNUM
      432Ø  76Ø4             LAS                      /YES,  LOOP ON ALL SELECTED TESTS
      4321  ØØ2Ø             AND      K1
      4322  765Ø             SNA CLA
      4323  5336             JMP      NOEXLP          /NO
      4324  2346             ISZ      PASCNT          /YES UPDATE PASS COUNTER
      4325  1346             TAD      PASCNT          /INSERT IN MSG AND
      4326  4463             JMS I    EDITP           /PRINT
      4327  3763             DCA I    M4P2P
      433Ø  1464             TAD I    EDTEMP
      4331  3764             DCA I    M4P1P
      4332  4461             JMS I    AMG8EP
      4333  7233             MSG4
      4334  4465             JMS I    BELLP           /BEL
```

```
4335 5245           JMP     EXECL2          /RUN ALL OVER AGAIN
4336 4461   NOEXLP, JMS I   AMG8EP          /PRINT DONE MESSAGE,
4337 7220           MSG3
4340 3175           DCA     ACTFLG          /CLEAR TEST IN PROGRESS,
4341 7240           CLA CMA
4342 3173           DCA     TTOFLG
4343 5462           JMP I   MONITP          /GO TO MONITOR
4344 5403   KJMPI3, 5403
4345 5400   INTSEP, INTSEV
4346 0000   PASCNT, Ø
4347 6244   KRMF,   RMF
4350 4145   SCDFIP, SCDFI
4351 0001   P1,     1
4352 0002   P2,     2
4353 0003   P3,     3
4354 0000   EXTAL,  Ø
4355 0000   TSTPP,  Ø
4356 0000   TALPP,  Ø
4357 0000   TSTAEX, Ø
4360 7216   M2P1P,  M2P1
4361 0000   TSTTAL, Ø
4362 0000   EXTEMP, Ø
4363 7237   M4P2P,  M4P2
4364 7236   M4P1P,  M4P1
4365 6562   TSTP,   TST-1

4366 0203   TET1P,  TESTØ1
4367 6576   TALP,   TAL-1
```

```
                /MONITOR

                        /1,ENTERED UNDER ONE OF THE FOLLOWING CONDITIONS:
                        /1,1PROGRAM STARTED AT 2ØØ WITH SRØ=1,
                        /1,2ERROR OCCURS WITH SR2=Ø,
                        /1,3ANY FATAL ERROR OCCURS,
                        /1,4"ALTMODE" IS STRUCK ON THE KEYBOARD AFTER TEST START,
                        /1,5SELECTED TESTS ARE RUN TO COMPLETION,

      44ØØ              PAGE
44ØØ  6ØØ2      MONIT,  IOF                     /INTERRUPT SYSTEM OFF,
44Ø1  474Ø              JMS I   SAVEPP          /SAVE SUBROUTINE ONFO,
44Ø2  3171              DCA     TRACE
44Ø3  4461              JMS I   AMG8EP
44Ø4  7253              MSG8                    /PRINT "E",
44Ø5  3337              DCA     TSTATM          /CLEAR TEST STATUS TEMP,
44Ø6  4743              JMS I   LISNP           /GET KEYBOARD,
44Ø7  7475              -3Ø3
441Ø  442Ø              CS                      /C STRING,
4411  7454              -324
4412  4435              TS                      /T STRING,
4413  7473              -3Ø5
4414  4527              ES                      /E STRING,
4415  7474              -3Ø4
4416  46ØØ              DS                      /"D" LOOK AT MEMORY,
4417  ØØØØ              Ø
442Ø  3335      CS,     DCA     TSTEM1          /"C" OR "EX", PUT AC IN TEMP,
4421  1175              TAD     ACTFLG          /IF NO TEST IN PROGRESS: SYNTAX
4422  77ØØ              SMA CLA                 /ERROR,
4423  5742              JMP I   QUESP
4424  4743              JMS I   LISNP           /GET KEYBOARD,
4425  7563              -215                    /"←",
4426  443Ø              .+2
4427  ØØØØ              Ø
443Ø  4466              JMS I   CRLFP           /"C←" OR "EX←",
4431  4741              JMS I   RESTPP          /RESTORE SUBROUTINES,
4432  1335              TAD     TSTEM1          /GET TEMP AND PUT IN EXIT
4433  3174              DCA     EXITFL          /GLAG,
4434  5744              JMP I   INTOKP          /EXIT TO INTERRUPT SERVICE,
4435  4743      TS,     JMS I   LISNP           /T STRING, GET KEYBOARD,
4436  7456              -322
4437  45Ø1              TR                      /"TR"
444Ø  7477              -3Ø1
4441  4517              TA                      /"TA"
4442  ØØØ1              1
4443  4445              .+2                     /"TN"
4444  ØØØØ              Ø
4445  71Ø4              CLL RAL                 /GET FIRST NUMBER, SCALE
4446  7ØØ6              RTL                     /AND SAVE,
4447  3335              DCA     TSTEM1
445Ø  4743              JMS I   LISNP           /GET NEXT NUMBER
4451  ØØØ1              1
4452  4454              .+2
4453  ØØØØ              Ø
```

```
4454  1335            TAD     TSTEM1          /ADD TWO NUMBERS AND CONVERT
4455  7041            CIA                     /SUM TO ISZ SENSITIVE COUNTER,
4456  3335            DCA     TSTEM1
4457  7120            CLL CML                 /DETERMINE TEST BIT,
4460  7010            RAR
4461  2335            ISZ     TSTEM1
4462  5260            JMP     .-2
4463  3335            DCA     TSTEM1          /SAVE TEST BIT,
4464  1335            TAD     TSTEM1          /EXCLUSIVE OR NEW TEST BIT
4465  0337            AND     TSTATM          /WITH OLD TEST STATUS,
4466  7104            CLL RAL
4467  7041            CIA
4470  1335            TAD     TSTEM1
4471  1337            TAD     TSTATM          /SAVE NEW TEST STATUS,
4472  3337            DCA     TSTATM
4473  4743            JMS I   LISNP           /GET INPUT,
4474  7563            -215                    /"TNN-"
4475  4513            TCR
4476  7454            -324                    /"TNNT-"
4477  4435            TS                      /GO BACK TO T STRING,
4500  0000            0
4501  7240    TR,     CLA CMA                 /"TR" TRACE REQUESTED,
4502  3171            DCA     TRACE           /SET TRACE FLAG,
4503  4743            JMS I   LISNP           /GET KEYBOARD,
4504  7475            -303
4505  4420            CS                      /"TRC"
4506  7454            -324
4507  4435            TS                      /"TRT"
4510  7473            -305
4511  4527            ES                      /"TRE"
4512  0000            0
4513  4523    TCR,    CLEAR1                  /NULL MAGTAPE INTERRUPTS, ION,
4514  4466            JMS I   CRLFP
4515  1337            TAD     TSTATM          /PUT TEST STATUS IN AC,
4516  5736            JMP I   REXP            /GO TO "EXEC."
4517  7240    TA,     CLA CMA                 /SET ALL TESTS,
4520  3337            DCA     TSTATM
4521  4743            JMS I   LISNP           /GET KEYBOARD,
4522  7454            -324
4523  4435            TS                      /"TAT"
4524  7563            -215
4525  4513            TCR                     /"TA-"
4526  0000            0
4527  4743    ES,     JMS I   LISNP           /E STRING,
4530  7450            -330                    /"EX"
4531  4533            .+2
4532  0000            0
4533  7240            CLA CMA                 /SET ERROR EXIT FLAG,
4534  5220            JMP     CS              /EXIT VIA C STRING,
4535  0000    TSTEM1, 0
4536  4240    REXP,   REX
4537  0000    TSTATM, 0
4540  6000    SAVEPP, SAVEP
4541  6021    RESTPP, RESTP
```

```
     4542  4673     QUESP,  QUES
     4543  4677     LISNP,  LISN
     4544  5444     INTOKP, INTOK

           46ØØ             PAGE
     46ØØ  1Ø4Ø     DS,     TAD     M4             /SET FOR 4 NUMBERS,
     46Ø1  327Ø             DCA     DS2
     46Ø2  4277             JMS     LISN           /LOOK FOR FIELD NUMBER',
     46Ø3  ØØØ1             1
     46Ø4  46Ø6             ,+2
     46Ø5  ØØØØ             Ø
     46Ø6  71Ø4             CLL RAL                /SCALE AND SAVE,
     46Ø7  7ØØ6             RTL
     461Ø  3267             DCA     DS1
     4611  4277             JMS     LISN           /LOOK FOR :
     4612  75Ø6             -272
     4613  4615             ,+2
     4614  ØØØØ             Ø
     4615  1267             TAD     DS1            /COMPUTE CDF INSTRUCTION,
     4616  1Ø46             TAD     KCDF
     4617  3236             DCA     DLF
     462Ø  3267             DCA     DS1            /CLEAR TEMP FOR NEXT 4 INPUTS,
     4621  4277     DLF2,   JMS     LISN           /GET KEYBOARD,
     4622  ØØØ1             1
     4623  4625             ,+2                    /OCTAL NUMBER,
     4624  ØØØØ             Ø
     4625  1267             TAD     DS1            /ADD TEMP,
     4626  227Ø             ISZ     DS2            /4TH NUMBER?
     4627  741Ø             SKP
     463Ø  5235             JMP     ,+5
     4631  71Ø4             CLL RAL                /NO, MOVE 3 PLACES LEFT,
     4632  7ØØ6             RTL                    /STORE,
     4633  3267             DCA     DS1
     4634  5221             JMP     DLF2           /GO TO NEXT NUMBER',
     4635  3267             DCA     DS1            /YES, 4TH NUMBER, SAVE ADDRESS,
     4636  62Ø1     DLF,    CDF     /N             /CHANGE TO REQUESTED DATA FIELD,
     4637  1667             TAD I   DS1            /GET CONTENTS,
     464Ø  4457     DLF1,   JMS I   SCDFP          /CHANGE BACK TO THIS PROGRAM FIELD,
     4641  4463             JMS I   EDITP          /EDIT AND INSERT IN
     4642  3672             DCA I   M11P2P         /MESSAGE,
     4643  1464             TAD I   EDTEMP
     4644  3671             DCA I   M11P1P
     4645  4461             JMS I   AMG8EP         /PRINT MESSAGE,
     4646  73Ø7             MSG11
     4647  4277             JMS     LISN
     465Ø  7563             -215                   /"D+"
     4651  4665             DCR
     4652  7566             -212
     4653  4655             ,+2                    /"DLF"
     4654  ØØØØ             Ø
     4655  1Ø31             TAD     K215           /"DLF", DUMP NEXT LOCATION,
     4656  447Ø             JMS I   TYPEP
     4657  2267             ISZ     DS1
     466Ø  5236             JMP     DLF
     4661  1236             TAD     DLF
```

```
4662  1Ø23             TAD     K1Ø
4663  3236             DCA     DLF
4664  5236             JMP     DLF
4665  4466   DCR,      JMS I   CRLFP            /"D-". LOOK FOR NEW COMMAND.
4666  5676             JMP I   MONP
4667  ØØØØ   DS1,      Ø
467Ø  ØØØØ   DS2,      Ø
4671  731Ø   M11P1P, M11P1
4672  7311   M11P2P, M11P2
4673  4461   QUES,     JMS I   AMGBEP           /TYPE "?-", RING BELL.
4674  7241             MSG6
4675  5676             JMP I   MONP             /RESTART MONITOR.
4676  44Ø3   MONP,     MONIT+3
4677  ØØØØ   LISN,     Ø                        /GET KEYBOARD AND CHECK SYNTAX.
47ØØ  6Ø31             KSF
47Ø1  53ØØ             JMP     .-1
47Ø2  6Ø36             KRB
47Ø3  3346             DCA     LISNT1           /SAVE INPUT
47Ø4  1346             TAD     LISNT1
47Ø5  447Ø             JMS I   TYPEP            /ECHO.
47Ø6  1346             TAD     LISNT1           /DO NOT CHECK "SPACE" IN SYNTAX.
47Ø7  1347             TAD     M24Ø
471Ø  765Ø             SNA CLA
4711  53ØØ             JMP     LISN+1
4712  1677   LISN1,    TAD I   LISN             /GET COMPARATOR.
4713  745Ø             SNA
4714  5273             JMP     QUES             /IF Ø, SYNTAX ERROR.
4715  75ØØ             SMA
4716  5327             JMP     LISNUM           /IF >Ø, LOOK FOR OCTAL NUMBER.
4717  1346             TAD     LISNT1           /MATCH?
472Ø  764Ø             SZA CLA
4721  5324             JMP     LISN2            /NO.
4722  3346             DCA     LISNT1           /YES.  FAKE TEMP CLEAR AND
4723  534Ø             JMP     LISN3            /EXIT.
4724  2277   LISN2,    ISZ     LISN             /UPDATE CALL POINTERS.
4725  2277             ISZ     LISN
4726  5312             JMP     LISN1            /GET NEXT COMPARATOR.
4727  72ØØ   LISNUM,   CLA                      /OCTAL NUMBER REQUIRED.
473Ø  1346             TAD     LISNT1
4731  1Ø44             TAD     M26Ø
4732  771Ø             SPA CLA
4733  5324             JMP     LISN2
4734  1346             TAD     LISNT1
4735  1Ø45             TAD     M27Ø
4736  77ØØ             SMA CLA
4737  5324             JMP     LISN2            /NOT AN OCTAL NUMBER
474Ø  2277   LISN3,    ISZ     LISN             /YES.  UPDATE CALL.
4741  1677             TAD I   LISN             /GET RETURN ADDRESS.
4742  3277             DCA     LISN
4743  1346             TAD     LISNT1           /GET Ø OR OCTAL NUMBER CODE
4744  3Ø22             AND     K7               /MASK TO LOW ORDER 3 BITS.
4745  5677             JMP I   LISN             /EXIT.
4746  ØØØØ   LISNT1,   Ø
4747  754Ø   M24Ø,     -24Ø
```

```
                  /ERROR SERVICE ROUTINE.
                          /DECODES ERROR STATUS WORDS AND CARRIES OUT THE INDICATED
                  /ACTION.
                          /SAMPLE ERROR TABLE ENTRY:
                  /ERØ5C, STAT1   ERROR STATUS WORD,
                  /       STAT2   PRINT STATUS WORD,

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

                  /STAT2  BIT             PRINT OUT THE FOLLOWING WHEN SET,
                          /Ø              IOT1 (I1)
                          /1              IOT2 (I2)
                          /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)
                          /1Ø             DATA BUFFER (DB)
                          /11             CONTENTS OF AC FOR IOT 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

      5ØØØ                PAGE
5ØØØ  ØØØØ        ERRORS, Ø               /ENTER HERE WITH "ERROR"
5ØØ1  72ØØ                CLA                     /ENTER HERE FROM "COMP" (PC PRESTORED)
5ØØ2  137Ø                TAD     K4352
5ØØ3  3467                DCA I   ERMSGP
5ØØ4  16ØØ                TAD I   ERRORS          /GET ERROR TABLE ADDRESS AND
5ØØ5  3345                DCA     ERRPTR          /SAVE,
5ØØ6  1745                TAD I   ERRPTR          /GET STAT 1,
5ØØ7  771Ø                SPA CLA                 /NON-FATAL ERROR?
5Ø1Ø  5231                JMP     FATERR          /NO,  FATAL ERROR,
5Ø11  424Ø                JMS     OPRNT           /NON-FATAL ERROR,  CHECK PRINTOUT,
5Ø12  4764                JMS I   OHALTP          /CHECK RETURN TO MONITOR,
5Ø13  22ØØ                ISZ     ERRORS          /MOVE POINTER TO SUBTEST LOOP,
5Ø14  4526                LOOP5                   /CHECK SUBTEST LOOP,
5Ø15  56ØØ                JMP I   ERRORS          /EXIT TO LOOP JUMP,
5Ø16  1174                TAD     EXITFL          /"EX-" PENDING?
5Ø17  77ØØ                SMA CLA
```

```
5Ø2Ø  5227          JMP     .+7          /NO.
5Ø21  1745          TAD I   ERRPTR       /YES.  DOES THIS ERROR REQUIRE
5Ø22  7ØØ4          RAL                  /"EX." SERVICE?
5Ø23  77ØØ          SMA CLA
5Ø24  5227          JMP     .+3          /NO.  LEAVE EXITFL SET.
5Ø25  22ØØ          ISZ     ERRORS       /YES.  MOVE POINTER TO RECYCLE JUMP.
5Ø26  3174          DCA     EXITFL       /CLEAR "EX." FLAG.
5Ø27  22ØØ          ISZ     ERRORS       /MOVE POINTER ONE MORE POSITION.
5Ø3Ø  56ØØ          JMP I   ERRORS       /EXIT.

5Ø31  4461  FATERR, JMS I   AMGBEP       /FATAL ERROR, PRINT MSG.
5Ø32  7243          MSG7
5Ø33  4465          JMS I   BELLP
5Ø34  1372          TAD     K6Ø5         /PUT "FE" IN AC.
5Ø35  4253          JMS     PRNT         /PRINT ERROR MESSAGE.
5Ø36  3175          DCA     ACTFLG       /CLEAAR TEST IN PROGRESS FLAG.
5Ø37  5462          JMP I   MONITP       /GO TO MONITOR.
5Ø4Ø  ØØØØ  QPRNT,  Ø                    /PRINTOUT?
5Ø41  76Ø4          LAS
5Ø42  ØØ32          AND     K4ØØ
5Ø43  764Ø          SZA CLA
5Ø44  564Ø          JMP I   QPRNT        /NO.  EXIT.
5Ø45  1371          TAD     K522         /YES.  PUT "ER" IN AC AND
5Ø46  4253          JMS     PRNT         /GENERATE ERROR PRINTOUT.
5Ø47  724Ø          CLA CMA              /BACK BIAS ERROR TABLE POINTER.
5Ø5Ø  1345          TAD     ERRPTR       /TO POINT TO STAT1.
5Ø51  3345          DCA     ERRPTR
5Ø52  564Ø          JMP I   QPRNT        /EXIT.

5Ø53  ØØØØ  PRNT,   Ø                    /GENERATE ERROR MESSAGE.
5Ø54  3746          DCA I   ERMP1P       /SAVE "FE" OR "ER".
5Ø55  1745          TAD I   ERRPTR       /GET TEST NUMBER FROM
5Ø56  4522          BSW                  /STAT1 AND PUT IN ERROR
5Ø57  Ø365          AND     K17          /MESSAGE.
5Ø6Ø  4463          JMS I   EDITP
5Ø61  3747          DCA I   ERMP2P
5Ø62  1745          TAD I   ERRPTR       /GET SUBTEST LETTER FROM STAT1
5Ø63  4522          BSW                  /AND PUT IN ERROR MESSAGE
5Ø64  Ø373          AND     K77ØØ        /AFTER APPENDING "SPACE" CODE.
5Ø65  1Ø25          TAD     K4Ø
5Ø66  375Ø          DCA I   ERMP3P
5Ø67  12ØØ          TAD     ERRORS       /EDIT RETURN JUMP FOR "PC" PRINTOUT.
5Ø7Ø  4463          JMS I   EDITP        /
5Ø71  3753          DCA I   ERMP5P
5Ø72  1464          TAD I   EDTEMP
5Ø73  3752          DCA I   ERMP4P
5Ø74  2345          ISZ     ERRPTR       /MOVE ERROR TABLE POINTER TO STAT2.
5Ø75  1Ø41          TAD     M5           /SET COUNTER FOR 5 SYMBOLS
5Ø76  3354          DCA     PRCNT1       /BEFORE CR-LF.
5Ø77  1Ø42          TAD     M14          /SET COUNTER FOR 12 SYMBOLS
51ØØ  3355          DCA     PRCNT2       /MAXIMUM.
51Ø1  1362          TAD     SYMPTP       /GET SYMBOL ROUTINF TARLE
51Ø2  3357          DCA     SYMBOL       /POINTER AND PUT IN TEMP.
51Ø3  1356          TAD     ERMPTP       /GET ERROR MESSAGE INSERT
51Ø4  336Ø          DCA     PRMPT        /POINTER.
```

```
5105  7130            STL RAR                 /INITIALIZE TO SPACE
5106  3361            DCA     PRNTK
5107  7604            LAS                     /COMPLETE DUMP?
5110  0030            AND     K200
5111  7650            SNA CLA
5112  5315            JMP     .+3
5113  7240            CLA CMA                 /YES,FAKE STAT2 FULL.
5114  7410            SKP
5115  1745            TAD I   ERRPTR          /GET STAT2 AND PUT IN TEMP.
5116  3363            DCA     WHAT
5117  1363    PRNTLP, TAD     WHAT            /BIT BY BIT INSPECT STAT2 FOR
5120  7004            RAL                     /SYMBOL PRINTOUTS,  AS A
5121  3363            DCA     WHAT            /BIT IS SENSED SET, GO TO
5122  7420            SNL                     /SYMBOL ROUTINE AND INSERT
5123  5335            JMP     PRNTCK          /SYMBOL AND DATA IN
5124  1757            TAD I   SYMBOL
5125  3351            DCA     SYMADR
5126  4751            JMS I   SYMADR
5127  2354            ISZ     PRCNT1          /5 SYMBOLS PRINTED?
5130  5333            JMP     .+3
5131  1366            TAD     K4300           /YES, INSERT 1 CR-LF IN SPACING,
5132  7410            SKP
5133  7130            STL RAR                 //NO, INSERT 1 SPACE.
5134  3361            DCA     PRNTK
5135  2357    PRNTCK, ISZ     SYMBOL          /UPDATE SYMBOL TABLE POINTER.
5136  2355            ISZ     PRCNT2          /11 SYMBOLS CHECKED?
5137  5317            JMP     PRNTLP          /NO, CHECK AGAIN.
5140  1366            TAD     K4300           /YES, PUT CR-LF AND END
5141  3760            DCA I   PRMPT           /MSG CODE IN ERROR MSG.
5142  4461            JMS I   AMG8EP          /PRINT ERROR MESSAGE.
5143  7313            ERMSG
5144  5653            JMP I   PRNT            /EXIT
5145  0000    ERRPTR, 0                       /TEMP FOR STATUS WORD POINTER.
5146  7314    ERMP1P, ERMP1                   /ERROR MESSAGE POINTERS: ERROR TYPE,
5147  7315    ERMP2P, ERMP2                   /! TEST NUMBER
5150  7316    ERMP3P, ERMP3                   /! SUBTEST LETTER-SPACE,
5151  0000    SYMADR, 0
5152  7321    ERMP4P, ERMP4                   /!HIGH ORDER PC.
5153  7322    ERMP5P, ERMP5                   /! LOW ORDER PC.
                                              /SYMBOL POINTER.
5154  0000    PRCNT1, 0                       /5 SYMBOL COUNTER.
5155  0000    PRCNT2, 0                       /12 SYMBOL COUNTER.
                                              /SYMBOL TABLE POINTER.
5156  7323    ERMPTP, ERMP5+1                 /ERROR MESSAGE INSERT START POINTER.
5157  0000    SYMBOL, 0                       /SYMBOL TABLE POINTER TEMP.
5160  0000    PRMPT,  0                               /ERROR MESSAGE INSERT POINTER.
5161  0000    PRNTK,  0                       /SPACING CONSTANT.
5162  6663    SYMPTP, SYMPT

5163  0000    WHAT,   0                       /TEMP FOR STAT2 ROTATES.
5164  5200    QHALTP, QHALT                   /POINTER TO RETURN TO MONITOR CHECK.
5165  0017    K17,    17
5166  4300    K4300,  4300
5167  4343    K4343,  4343
5170  4352    K4352,  4352
```

```
        5171  0522     K522,   522
        5172  0605     K605,   605
        5173  7700     K7700,  7700

              5200             PAGE
        5200  0000     QHALT,  Ø                           /RETURN TO MONITOR?
        5201  7604             LAS
        5202  0366             AND     K1000
        5203  7640             SZA CLA
        5204  5600             JMP I   QHALT               /NO,
        5205  6002             IOF                         /YES, TURN INTERRUPT OFF,
        5206  1213             TAD     HALTCP              /PUT CONTINUE EXIT IN Ø
        5207  3000             DCA     Ø                   /SO "C." OR "EX." EXITS TO
        5210  5462             JMP I   MONITP              /HALTC, GO TO MONITOR,
        5211  7200     HALTC,  CLA                         /CONTINUE,
        5212  5600             JMP I   QHALT               /EXIT,
        5213  5211     HALTCP, HALTC                       /HALTC POINTER,

                       /SYMBOL ROUTINES,
        5214  0000     GD,     Ø                           /GOOD,
        5215  1154             TAD     GOOD                /GET GOOD,
        5216  4330             JMS     SYMSEV
        5217  0407             Ø407                        /"GD",
        5220  5614             JMP I   GD
        5221  0000     BD,     Ø
        5222  1155             TAD     BAD                 /GET BAD,
        5223  4330             JMS     SYMSEV
        5224  0402             Ø402                        /"BD",
        5225  5621             JMP I   BD
        5226  0000     OD,     Ø
        5227  1156             TAD     OLD                 /GET OLD
        5230  4330             JMS     SYMSEV
        5231  0417             Ø417                        /"OD",
        5232  5626             JMP I   OD
        5233  0000     WC,     Ø
        5234  4503             RWCR                        /GET WC,
        5235  4330             JMS     SYMSEV
        5236  0327             Ø327                        /"WC",
        5237  5633             JMP I   WC
        5240  0000     CA,     Ø
        5241  4504             RCAR                        /GET CA
        5242  4330             JMS     SYMSEV
        5243  0103             Ø103                        /"CA",
        5244  5640             JMP I   CA
        5245  0000     CM,     Ø
        5246  4506             RCMR                        /GET CM
        5247  4330             JMS     SYMSEV
        5250  1503             1503                        /"CM",
        5251  5645             JMP I   CM
        5252  0000     FS,     Ø
        5253  4507             RFSR                        /GET FS
        5254  4330             JMS     SYMSEV
        5255  2306             2306                        /"FS"
        5256  5652             JMP I   FS
        5257  0000     MS,     Ø
```

```
5260  4505            RMSR                    /GET MS
5261  4330            JMS     SYMSEV
5262  2315            2315                    /"MS"
5263  5657            JMP I   MS
5264  0000    DB,     0
5265  4510            RDBR                    /GET DB
5266  4330            JMS     SYMSEV
5267  0204            0204                    /"DB"
5270  5664            JMP I   DB


5271  0000    AC,     0
5272  1170            TAD     ACLOC
5273  4330            JMS     SYMSEV
5274  0301            0301
5275  5671            JMP I   AC
5276  0000    I1,     0
5277  1703            TAD I   IOT1P
5300  4312            JMS     I1I2
5301  6111            6111
5302  5676            JMP I   I1
5303  0566    IOT1P,  IOT1
5304  0000    I2,     0
5305  1711            TAD I   IOT2P
5306  4312            JMS     I1I2
5307  6211            6211
5310  5704            JMP I   I2
5311  0571    IOT2P,  IOT2
5312  0000    I1I2,   0                       /ENTER FROM I1 OR I2,
5313  0370            AND     K377            /CONVERT INST TO TAD POINTER,
5314  1366            TAD     K1000
5315  3316            DCA     .+1             /INTO TAD POINTER,
5316  0000            0                       /STORE TAD POINTER HERE,
5317  7001            IAC                     /POINTER +1 WHICH IS ADDRESS
5320  3316            DCA     .-2             /OF IOT CODE,
5321  1712            TAD I   I1I2            /GET SYMBOL CODE,
5322  3325            DCA     .+3             /PUT IN CALL TO SYMSEV,
5323  1716            TAD I   .-5             /GET IOT CODE,
5324  4330            JMS     SYMSEV          /GO TO SYMBOL SERVICE,
5325  0000            0                       /SYMBOL CODE GOES HERE,
5326  2312            ISZ     I1I2            /UPDATE RETURN TO I1 OR I2,
5327  5712            JMP I   I1I2            /RETURN,
5330  0000    SYMSEV, 0                       /COMMON SYMBOL ROUTINE SERVICE,
5331  3365            DCA     SYMTEM          /SAVE DATA,
5332  1763            TAD I   ERMPP           /GET ERROR MESSAGA CURRENT POINTER,
5333  3364            DCA     ERMSYM          /PUT ON THIS PAGE OF MEMORY,
5334  1730            TAD I   SYMSEV
5335  0377            AND     (77
5336  1776            TAD I   (PRNTK
5337  3764            DCA I   ERMSYM
5340  2364            ISZ     ERMSYM
5341  1730            TAD I   SYMSEV
5342  0375            AND     (7700
5343  1357            TAD     K72
5344  3764            DCA I   ERMSYM
```

```
5345  2364           ISZ     ERMSYM          /DATA.
5346  1365           TAD     SYMTEM          /GET DATA.
5347  4463           JMS I   EDITP           /EDIT.
535Ø  3365           DCA     SYMTEM          /SAVE LOW ORDER.
5351  1464           TAD I   EDTEMP          /GET HIGH ORDER AND INSERT.
5352  3764           DCA I   ERMSYM
5353  2364           ISZ     ERMSYM          /UPDATE POINTER TO LOW ORDER
5354  1365           TAD     SYMTEM          /DATA, GET LOW ORDER AND INSERT.
5355  3764           DCA I   ERMSYM
5356  2364           ISZ     ERMSYM          /UPDAT POINTER TO SPACING.
5357  1364           TAD     ERMSYM          /PUT CURRENT POINTER ON
536Ø  3763           DCA I   ERMPP           /"ERRORS" MEMORY PAGE.
5361  233Ø           ISZ     SYMSEV          /UPDATE RETURN.
5362  573Ø           JMP I   SYMSEV          /EXIT.
5363  516Ø    ERMPP,  PRMPT
5364  ØØØØ    ERMSYM, Ø
5365  ØØØØ    SYMTEM, Ø
5366  1ØØØ    K1ØØØ,  1ØØØ
5367  ØØ72    K72,    72
537Ø  Ø377    K377,   377
```

```
                /GENERAL INTERRUPT HANDLER,

                        /CHECKS FOR TTO AND TTI INTERRUPTS AND SERVICES SAME,
                        /CHECKS APPLICABILITY OF EF AND MTTF INTERRUPTS AND
                        /GOES TO SPECIFIED EF OR MTTF INTERRUPT HANDLER IF
                        /SPECIFIED IN INTEF AND INTMTF,

5375  772Ø
5376  5161
5377  2Ø77
      54ØØ              PAGE

54ØØ  3162      INTSEV, DCA     SAVEAC          /SAVE AC AND LINK,
54Ø1  7Ø1Ø              RAR
54Ø2  3163              DCA     SAVEL
54Ø3  62Ø1              CDF     ØØ              /GET FIELD Ø LOC Ø
54Ø4  1662              TAD I   PØ              /AND PUT IN THIS FIELD'S
54Ø5  3ØØØ              DCA     Ø               /LOC Ø FOR INT RETURN,
54Ø6  4457              JMS I   SCDFP           /CHANGE TO THIS DF,
54Ø7  6Ø41      INTAGN, TSF                     /TTO INTERRUPT?
541Ø  5215              JMP     .+5
5411  6Ø42              TCF                     /YES, CLEAR HARDWARE FLAG
5412  724Ø              CLA CMA                 /AND SET SOFTWARE FLAG,
5413  3173              DCA     TTOFLG
5414  5244              JMP     INTOK           /TRY TO EXIT,
5415  6Ø31              KSF                     /NO, KEYBOARD INTERRUPT?
5416  5232              JMP     INTNGA
5417  6Ø36              KRB                     /YES, GO TO MONITOR IF INPUT
542Ø  1257              TAD     M233
5421  745Ø              SNA
5422  5462              JMP I   MONITP
5423  126Ø              TAD     M142
5424  745Ø              SNA
5425  5462              JMP I   MONITP
5426  1261              TAD     M1
5427  765Ø              SNA CLA
543Ø  5462              JMP I   MONITP
5431  5244              JMP     INTOK           /OTHERWISE TRY TO EXIT,
5432  1164      INTNGA, TAD     INTEF           /EF INTERRUPT NULL?
5433  764Ø              SZA CLA
5434  5564              JMP I   INTEF           /NO, SERVICE,
5435  1166              TAD     INTMTF          /YES, MTTF INTERRUPT NULL?
5436  764Ø              SZA CLA
5437  5566              JMP I   INTMTF          /NO, SERVICE,
544Ø  1ØØØ              TAD     Ø
5441  3155              DCA     BAD
5442  4534              ERROR                   /ILLEGAL INTERRUPT
5443  6677              FEØØA                   /FATAL ERROR,
                /ILLEGAL INTERRUPT.  THE INTERRUPT CAUSE MAY BE THE TMP-E OR ANY
                /DEVICE EXCEPT THE TTY.  THIS ERROR IS FATAL AND KILLS ALL TEST
                /SELECTIONS,
                /BD=CONTENTS OF LOCATION Ø; ALL OTHERS ARE CURRENT VALUES AND
                /MAY NOT APPLY,
5444  73ØØ      INTOK,  CLA CLL                 /NO, RESTORE AC AND LINK
```

```
5445  1163              TAD     SAVEL           /AND EXIT,
5446  7004              RAL
5447  1172              TAD     INTFLG
5450  7640              SZA CLA
5451  5254              JMP     .+3
5452  1162              TAD     SAVEAC
5453  5400              JMP I   0
5454  1162              TAD     SAVEAC
5455  6001              ION
5456  5400              JMP I   0
5457  7545      M233,   -233
5460  7636      M142,   -142
5461  7777      M1,     -1
5462  0000      P0,     0

                /EF AND MTTF INTERRUPT SERVICE ROUTINES SPECIFIER,
                        /ENTER WITH
                        /JMS I  INSRSP
                        /ADDRESS OF EF SERVICE  PUT IN INTEF
                        /ADDRESS OF MTTF SERVICE        PUT IN INTMTF
5463  0000      INSRS,  0
5464  7340              CLA CLL CMA
5465  3175              DCA     ACTFLG          /SET ACTIVE FLAG,
5466  1663              TAD I   INSRS
5467  3164              DCA     INTEF
5470  2263              ISZ     INSRS
5471  1663              TAD I   INSRS
5472  3166              DCA     INTMTF
5473  1164              TAD     INTEF
5474  1166              TAD     INTMTF
5475  7640              SZA CLA
5476  5305              JMP     INSRSF
5477  3154              DCA     GOOD
5500  4477              LCMR
5501  4506              RCMR
5502  4533              COMPAR
5503  6701              ER00B
                /CM SHOULD CONTAIN 0000 BUT DOES NOT,
                /GD=GOOD CM,
5504  5300              JMP     .-4             /SUBTEST LOOP
5505  2263      INSRSF, ISZ     INSRS
5506  5663              JMP I   INSRS
```

```
                        /EXECUTIVE ADDENDUM.
      5507  0000        EXECFX, Ø                       /ENQUIRE ABOUT EXISTING
      551Ø  4461                JMS I   AMG8EP          /EXTENDED MEMORY FIELDS
      5511  7273                MSG1Ø                   /FOR THE CA MEMORY REFERENCE
      5512  6Ø31                KSF                     /TEST.
      5513  5312                JMP     .-1
      5514  6Ø36                KRB
      5515  3157                DCA     TXXTM1
      5516  1157                TAD     TXXTM1
      5517  447Ø                JMS I   TYPEP
      552Ø  1157                TAD     TXXTM1
      5521  1Ø44                TAD     M26Ø
      5522  771Ø                SPA CLA
      5523  531Ø                JMP     EXECFX+1
      5524  1157                TAD     TXXTM1
      5525  1Ø45                TAD     M27Ø
      5526  77ØØ                SMA CLA
      5527  531Ø                JMP     EXECFX+1
      553Ø  1157                TAD     TXXTM1
      5531  ØØ22                AND     K7
      5532  7ØØ1                IAC
      5533  7Ø41                CIA
      5534  3167                DCA     MEMFLD
      5535  4466                JMS I   CRLFP
      5536  57Ø7                JMP I   EXECFX
```

```
                        /COMPARE AND ERROR DETECTION ROUTINE.

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

                                /COMPAR                 CALL COMPARE.
                                /ERNNX                  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.


          5600                  PAGE
5600  0000              COMP,   Ø
5601  3155                      DCA     BAD             /SAVE AC IN BAD.
5602  1154                      TAD     GOOD            /COMPARE GOOD AND BAD.
5603  7041                      CIA
5604  1155                      TAD     BAD
5605  7650                      SNA CLA
5606  5212                      JMP     .+4
5607  1200                      TAD     COMP            /ERROR, SET UP AND GO TO
5610  3534                      DCA I   ERRORP          /ERRORS AND DON'T
5611  5636                      JMP I   ERR1P           /RETURN HERE.
5612  1171                      TAD     TRACE           /TRACE REQUESTED?
5613  7700                      SMA CLA
5614  5231                      JMP     COMPF
5615  7604                      LAS                     /YES, TRACE INHIBITED?
5616  0032                      AND     K400
5617  7640                      SZA CLA
5620  5231                      JMP     COMPF
5621  1200                      TAD     COMP            /NO, GET ERROR POINTER
5622  3534                      DCA I   ERRORP          /AND PUT IN ERRORS.
5623  1600                      TAD I   COMP            /GET STAT1 ADDRESS AND
5624  3640                      DCA I   TRPP            /PUT IN PRNT.
5625  1242                      TAD     K4324
5626  3467                      DCA I   ERMSGP
5627  1241                      TAD     K2252
5630  4637                      JMS I   PRNTP           /FAKE AN ERROR PRINTOUT.
5631  2200              COMPF,  ISZ     COMP            /UPDATE RETURN JUMP TO SUBTEST LOOP JUMP.
5632  4526                      LOOP5                   /SUBTEST LOOP?
5633  7410                      SKP
5634  2200                      ISZ     COMP            /NO, UPDATE ONE MORE.
5635  5600                      JMP I   COMP            /EXIT.
5636  5001              ERR1P,  ERRORS+1
5637  5053              PRNTP,  PRNT
5640  5145              TRPP,   ERRPTR
5641  2252              K2252,  2252
5642  4324              K4324,  4324
```

```
                        /SAVE AND RESTORE COMMON SUBROUTINE INFORMATION ROUTINES,
                                /LOCATIONS TYPE, TYPTEM, CRLF, AMGRE, AMG8E1, AMG8E2,
                                /AMG8E3, EDIT, EDTEM, EDTEM1, EDTEM2, EDTEM3, ETC. MUST BE SAVED AND RESTORED
                                /SINCE MONITOR MAY INTERRUPT THESE ROUTINES
                                /AND USE THEM ITSELF,
              6ØØØ              PAGE
        6ØØØ  ØØØØ      SAVEP,  Ø
        6ØØ1  72ØØ              CLA
        6ØØ2  1242              TAD     M16
        6ØØ3  3243              DCA     SAVEP1
        6ØØ4  1266              TAD     SRPTS
        6ØØ5  3244              DCA     SAVEP2
        6ØØ6  1247              TAD     SRHOLD
        6ØØ7  3245              DCA     SAVEP3
        6Ø1Ø  2244              ISZ     SAVEP2
        6Ø11  2245              ISZ     SAVEP3
        6Ø12  1644              TAD I   SAVEP2
        6Ø13  3246              DCA     SAVEP4
        6Ø14  1646              TAD I   SAVEP4
        6Ø15  3645              DCA I   SAVEP3
        6Ø16  2243              ISZ     SAVEP1
        6Ø17  521Ø              JMP     .-7
        6Ø2Ø  56ØØ              JMP I   SAVEP
        6Ø21  ØØØØ      RESTP,  Ø
        6Ø22  72ØØ              CLA
        6Ø23  1242              TAD     M16
        6Ø24  3243              DCA     SAVEP1
        6Ø25  1247              TAD     SRHOLD
        6Ø26  3244              DCA     SAVEP2
        6Ø27  1266              TAD     SRPTS
        6Ø3Ø  3245              DCA     SAVEP3
        6Ø31  2244              ISZ     SAVEP2
        6Ø32  2245              ISZ     SAVEP3
        6Ø33  1645              TAD I   SAVEP3
        6Ø34  3246              DCA     SAVEP4
        6Ø35  1644              TAD I   SAVEP2
        6Ø36  3646              DCA I   SAVEP4
        6Ø37  2243              ISZ     SAVEP1
        6Ø4Ø  5231              JMP     .-7
        6Ø41  5621              JMP I   RESTP
        6Ø42  7762      M16,    -16
        6Ø43  ØØØØ      SAVEP1, Ø
        6Ø44  ØØØØ      SAVEP2, Ø
        6Ø45  ØØØØ      SAVEP3, Ø
        6Ø46  ØØØØ      SAVEP4, Ø
        6Ø47  6Ø47      SRHOLD, .
        6Ø5Ø  ØØØØ              Ø
        6Ø51  ØØØØ              Ø
        6Ø52  ØØØØ              Ø
        6Ø53  ØØØØ              Ø
        6Ø54  ØØØØ              Ø
        6Ø55  ØØØØ              Ø
        6Ø56  ØØØØ              Ø
        6Ø57  ØØØØ              Ø
        6Ø6Ø  ØØØØ              Ø
        6Ø61  ØØØØ              Ø
        6Ø62  ØØØØ              Ø
        6Ø63  ØØØØ              Ø
        6Ø64  ØØØØ              Ø
        6Ø65  ØØØØ              Ø
        6Ø66  6Ø66     SRPTS,   .
        6Ø67  627Ø              CRLF
        6Ø7Ø  6244              TYPE
        6Ø71  6261              TYPTEM
        6Ø72  62ØØ              AMG8E
        6Ø73  6237              AMG8E1
        6Ø74  6214              AMG8E2
        6Ø75  624Ø              AMG8E3
        6Ø76  63Ø1              EDIT
        6Ø77  6326              EDTEM
        61ØØ  6327              EDTEM1
        61Ø1  633Ø              EDTEM2
        61Ø2  6331              EDTEM3
        61Ø3  6547              SRBSW
        61Ø4  6561              SRBSWT
```

```
                        /LOOP SWITCH SENSORS.
      61Ø5  ØØØØ    LAS5,   Ø                            /ENTER BY "LOOP5".
      61Ø6  76Ø4            LAS
      61Ø7  ØØ27            AND     K1ØØ
      611Ø  765Ø            SNA CLA
      6111  23Ø5            ISZ     LAS5
      6112  57Ø5            JMP I   LAS5
      6113  ØØØØ    LAS6,   Ø                            /"LOOP6"
      6114  76Ø4            LAS
      6115  ØØ25            AND     K4Ø
      6116  765Ø            SNA CLA
      6117  2313            ISZ     LAS6
      612Ø  5713            JMP I   LAS6
      6121  ØØØØ    LAS7,   Ø                            /"LOOP7"
      6122  76Ø4            LAS
      6123  ØØ24            AND     K2Ø
      6124  765Ø            SNA CLA
      6125  2321            ISZ     LAS7
      6126  5721            JMP I   LAS7
      6127  ØØØØ    LAS8,   Ø                            /"LOOP8"
      613Ø  76Ø4            LAS
      6131  ØØ23            AND     K1Ø
      6132  765Ø            SNA CLA
      6133  2327            ISZ     LAS8
      6134  5727            JMP I   LAS8
      6135  ØØØØ    LAS9,   Ø                            /"LOOP9"
      6136  76Ø4            LAS
      6137  Ø343            AND     K4
      614Ø  765Ø            SNA CLA
      6141  2335            ISZ     LAS9
      6142  5735            JMP I   LAS9
      6143  ØØØ4    K4,     4
```

```
                        /PDP8-E PACKED ANSCII MESSAGE GENERATOR.
                                /ENTERED WITH:
                                /JMS    AMG8E
                                /MSGNP  (MESSAGE POINTER)
                                /
                                /ØØ IS TERMINATOR, 43 (#) IS CARRIAGE RETURN, LINEFEED,
                                /THE FOLLOWING CHARACTERS ARE NOT ALLOWED! @ WHICH GIVES
                                /TERMINATOR CODE, # WHICH RESULTS IN CRLF, AND 2Ø4-2Ø7,
                                /212, 215, 375, 377, 233.
                                /
                                /EXITS WITH AC CLEAR.
        62ØØ                    PAGE

6200    ØØØØ            AMG8E,  Ø
6201    73ØØ                    CLA CLL
6202    16ØØ                    TAD I   AMG8E
6203    3237                    DCA     AMG8E1
6204    22ØØ                    ISZ     AMG8E
6205    1637                    TAD I   AMG8E1
6206    4522                    BSW
6207    4214                    JMS     AMG8E2
6210    1637                    TAD I   AMG8E1
6211    4214                    JMS     AMG8E2
6212    2237                    ISZ     AMG8E1
6213    52Ø5                    JMP     .-6
6214    ØØØØ            AMG8E2, Ø
6215    Ø241                    AND     K77
6216    745Ø                    SNA
6217    56ØØ                    JMP I   AMG8E
6220    324Ø                    DCA     AMG8E3
6221    124Ø                    TAD     AMG8E3
6222    1243                    TAD     M43
6223    764Ø                    SZA CLA
6224    5227                    JMP     .+3
6225    4466                    JMS I   CRLFP
6226    5614                    JMP I   AMG8E2
6227    124Ø                    TAD     AMG8E3
623Ø    1242                    TAD     M4Ø
6231    771Ø                    SPA CLA
6232    1Ø27                    TAD     K1ØØ
6233    1Ø3Ø                    TAD     K2ØØ
6234    124Ø                    TAD     AMG8E3
6235    447Ø                    JMS I   TYPEP
6236    5614                    JMP I   AMG8E2
6237    ØØØØ            AMG8E1, Ø
624Ø    ØØØØ            AMG8E3, Ø
6241    ØØ77            K77,    77
6242    774Ø            M4Ø,    -4Ø
6243    7735            M43,    -43

                        /TYPE SUBROUTINE.
                                /ENTER WITH ANSCII IN AC.  EXIT WITH AC CLEAR
6244    ØØØØ            TYPE,   Ø
6245    3261                    DCA     TYPTEM          /SAVE ANSCII.
```

```
        6246  6Ø41            TSF
        6247  741Ø            SKP
        625Ø  5254            JMP     .+4
        6251  1173            TAD     TTOFLG
        6252  77ØØ            SMA CLA
        6253  5246            JMP     .-5
        6254  3173            DCA     TTOFLG
        6255  1261            TAD     TYPTEM
        6256  6Ø46            TLS                     /TYPE CHARACTER IF REAL TTO FLAG
        6257  72ØØ            CLA                     /IS SET OR SOFTWARE FLAG SET.
        626Ø  5644            JMP I   TYPE
        6261  ØØØØ      TYPTEM, Ø

                        /RING TTY BELL.

        6262  ØØØØ      BELL,   Ø
        6263  72ØØ              CLA
        6264  1267              TAD     K2Ø7
        6265  447Ø              JMS I   TYPEP
        6266  5662              JMP I   BELL
        6267  Ø2Ø7      K2Ø7,   2Ø7

                        /CARRIAGE RETURN-LINE FEED.
        627Ø  ØØØØ      CRLF,   Ø
        6271  72ØØ              CLA
        6272  1Ø31              TAD     K215
        6273  447Ø              JMS I   TYPEP
        6274  13ØØ              TAD     K212
        6275  447Ø              JMS I   TYPEP
        6276  447Ø              JMS I   TYPEP
        6277  567Ø              JMP I   CRLF
        63ØØ  Ø212      K212,   212

                        /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".
        63Ø1  ØØØØ      EDIT,   Ø
        63Ø2  3327              DCA     EDTEM1
        63Ø3  1327              TAD     EDTEM1
        63Ø4  4522              BSW
        63Ø5  4312              JMS     EDIT1
        63Ø6  3326              DCA     EDTEM
        63Ø7  1327              TAD     EDTEM1
        631Ø  4312              JMS     EDIT1
        6311  57Ø1              JMP I   EDIT
        6312  ØØØØ      EDIT1,  Ø
        6313  333Ø              DCA     EDTEM2
        6314  133Ø              TAD     EDTEM2
        6315  ØØ22              AND     K7
        6316  3331              DCA     EDTEM3
        6317  133Ø              TAD     EDTEM2
        632Ø  7ØØ6              RTL
        6321  7ØØ4              RAL
        6322  Ø333              AND     K7ØØ
```

```
6323  1331            TAD     EDTEM3
6324  1332            TAD     K6Ø6Ø
6325  5712            JMP I   EDIT1
6326  ØØØØ    EDTEM,  Ø
6327  ØØØØ    EDTEM1, Ø
633Ø  ØØØØ    EDTEM2, Ø
6331  ØØØØ    EDTEM3, Ø
6332  6Ø6Ø    K6Ø6Ø,  6Ø6Ø
6333  Ø7ØØ    K7ØØ,   7ØØ
```

```
                        /ROUTINES TO CLEAR ALL FLAGS SAFELY,
        64ØØ                    PAGE
  64ØØ  ØØØØ            CLR1,   Ø                       /"CLEAR1"
  64Ø1  446Ø                    JMS I   INSRSP          /NULL EF AND MTTF INTERRUPTS,
  64Ø2  Ø2ØØ                    Ø
  64Ø3  ØØØ2                    2
  64Ø4  42Ø6                    JMS     CLR2            /DO "CLEAR2"
  64Ø5  56ØØ                    JMP I   CLR1
  64Ø6  ØØØØ            CLR2,   Ø                       /"CLEAR2"
  64Ø7  4214                    JMS     CLR3            /DO "CLEAR3"
  641Ø  724Ø                    CLA CMA                 /SET INTERRUPT ACTIVE FLAG
  6411  3172                    DCA     INTFLG
  6412  6ØØ1                    ION                     /INTERRUPT ON,
  6413  56Ø6                    JMP I   CLR2
  6414  ØØØØ            CLR3,   Ø                       /"CLEAR3"
  6415  73ØØ                    CLA CLL                 /ENSURE TTO SOFTWARE FLAG
  6416  1173                    TAD     TTOFLG          /IS SET PRIOR TO CAF
  6417  764Ø                    SZA CLA                 /GENERATE "INITIALIZE"
  642Ø  5223                    JMP     .+3
  6421  6Ø41                    TSF
  6422  5216                    JMP     .-4
  6423  734Ø                    CLA CMA CLL
  6424  3173                    DCA     TTOFLG
  6425  6ØØ2                    IOF
  6426  45Ø1                    LDBR                    /SINCE "INITIALIZE WILL NOT CLEAR DB,
  6427  4517                    CLT
  643Ø  3324                    DCA     FUNCT           /ZERO FR TEMP,
  6431  5614                    JMP I   CLR3
```

```
                  /FUNCTION SUBROUTINES,

                  /TO ENABLE RECALL OF BIT 6 IN FR, EACH TIME THE FR IS LOADED
                  /IN THE FUNC ROUTINE, THE FR CONSTANT LOADED IS SAVED IN CORE, IF INITIALIZE
                  /IS GENERATED, THIS CORE LOCATION IS ZEROED,

6432  ØØØØ        OFFLNE, Ø                 /OFF LINE, (OFFLIN)
6433  4311                JMS     FUNC
6434  ØØØØ                Ø
6435  5632                JMP I   OFFLNE
6436  3ØØØ        REWND,  Ø                          /REWIND, (REWIND)
6437  4311                JMS     FUNC
644Ø  1ØØØ                1ØØØ
6441  5636                JMP I   REWND
6442  ØØØØ        READD,  Ø                          /READ, (READ)
6443  4311                JMS     FUNC
6444  2ØØØ                2ØØØ
6445  5642                JMP I   READD

6446  ØØØØ        RDCMP,  Ø                          /READ COMPARE (RDCOMP)
6447  4311                JMS     FUNC
645Ø  3ØØØ                3ØØØ
6451  5646                JMP I   RDCMP
6452  ØØØØ        WRIT,   Ø                          /WRITE, (WRITE)
6453  4311                JMS     FUNC
6454  4ØØØ                4ØØØ
6455  5652                JMP I   WRIT
6456  ØØØØ        EOFW,   Ø                          /WRITE EOF, (WEOF)
6457  4311                JMS     FUNC
646Ø  5ØØØ                5ØØØ
6461  5656                JMP I   EOFW
6462  ØØØØ        SPFORD, Ø                          /SPACE FORWARD (SPCFWD)
6463  4311                JMS     FUNC
6464  6ØØØ                6ØØØ
6465  5662                JMP I   SPFORD
6466  ØØØØ        SPREV,  Ø                          /SPACE REVERSE, (SPCREV)
6467  4311                JMS     FUNC
647Ø  7ØØØ                7ØØØ
6471  5666                JMP I   SPREV
6472  ØØØØ        XGAP,   Ø                          /SET EXTENDED GAP BIT IN CONSTANT
                                                     /(EXTGAP)

6473  72ØØ                CLA
6474  1Ø32                TAD     K4ØØ
6475  3277                DCA     KXGAP
6476  5672                JMP I   XGAP
6477  ØØØØ        KXGAP,  Ø
65ØØ  ØØØØ        GOE,    Ø                          /SET GO BIT NOT CHANGING FR
65Ø1  72ØØ                CLA                        /OTHERWISE, (GO)
65Ø2  1324                TAD     FUNCT
65Ø3  231Ø                AND     K76ØØA
65Ø4  1Ø27                TAD     K1ØØ
65Ø5  45ØØ                LFGR
65Ø6  72ØØ                CLA
65Ø7  57ØØ                JMP I   GOE
651Ø  76ØØ        K76ØØA, 76ØØ
```

```
6511  ØØØØ    FUNC,   Ø                          /LOAD FR WITH FUNCTION +
6512  72ØØ            CLA                        /EXTENDED GAP BIT CONSTANT
6513  1711            TAD I   FUNC
6514  1277            TAD     KXGAP
6515  3324            DCA     FUNCT              /SAVE FUNCTION,
6516  1324            TAD     FUNCT
6517  45ØØ            LFGR
652Ø  72ØØ            CLA
6521  3277            DCA     KXGAP
6522  2311            ISZ     FUNC
6523  5711            JMP I   FUNC
6524  ØØØØ    FUNCT,  Ø
```

```
                        /LOAD THE COMMAND REGISTER WITH THE CONSTANT IN THE CALLING SEQUENCE,
                                /ENTERED BY:
                                /COMAND         EQUIVALENT JMS I CMDP
                                /"CONSTANT"             CONSTANT TO BE LOADED

6525  ØØØØ      CMD,    Ø                       /"COMAND"
6526  72ØØ              CLA
6527  1725              TAD I   CMD             /GET CONSTANT.
653Ø  4477              LCMR                    /LOAD.
6531  72ØØ              CLA                     /ENSURE AC CLEAR.
6532  2325              ISZ     CMD             /UPDATE RETURN.
6533  5725              JMP I   CMD             /EXIT.

                        /ISSUE ONE BREAK REQUEST
                                /ENTERED BY "BREAK".

6534  ØØØØ      DATBRK, Ø                       /"BREAK"
6535  4521              SBRM
6536  7ØØØ              NOP
6537  7ØØØ              NOP
654Ø  5734              JMP I   DATBRK

6541  ØØØØ      WCOVER, Ø                       /GENERATE CONDITIONS NECESSARY
6542  455Ø              GO                      /FOR WCOV, SET GO BIT, WC TO
6543  724Ø              CLA CMA                 /7777 - LEAVE REMAINDER OF FR ALONE.
6544  4473              LWCR
6545  4535              BREAK
6546  5741              JMP I   WCOVER


                        /BYTE SWAP ROUTINE.

6547  ØØØØ      SRBSW,  Ø
655Ø  71Ø6              CLL RTL
6551  7ØØ6              RTL
6552  7ØØ6              RTL
6553  3361              DCA     SRBSWT
6554  7ØØ4              RAL
6555  1361              TAD     SRBSWT
6556  Ø362              AND     SRBSWN
6557  1361              TAD     SRBSWT
656Ø  5747              JMP I   SRBSW
6561  ØØØØ      SRBSWT, Ø
6562  ØØ77      SRBSWN, 77
```

```
6563  Ø2Ø3    TST,    TESTØ1          /EXEC TEST POINTERS,
6564  Ø6ØØ            TESTØ2
6565  1ØØØ            TESTØ3
6566  1235            TESTØ4
6567  14ØØ            TESTØ5
657Ø  16ØØ            TESTØ6
6571  2ØØØ            TESTØ7
6572  22ØØ            TEST1Ø
6573  3ØØØ            TEST11
6574  32ØØ            TEST12
6575  34ØØ            TEST13
6576  36ØØ            TEST14
6577  ØØØØ    TAL,    Ø               /TEST TALLIES FOR: TØ1
66ØØ  7766            -12             /TØ2
66Ø1  ØØØØ            Ø               /TØ3
66Ø2  7766            -12             /TØ4
66Ø3  7766            -12             /TØ5
66Ø4  7766            -12             /TØ6
66Ø5  ØØØØ            Ø               /TØ7
66Ø6  ØØØØ            Ø               /T1Ø
66Ø7  ØØØØ            Ø               /T11
661Ø  ØØØØ            Ø               /T12
6611  ØØØØ            Ø               /T13
6612  7766            -12             /T14
6613  4473    T1LS1,  LWCR            /T1LS1-T1LS3 ARE UTILIZED BY
6614  4475            LCAR            /THE TEST Ø1 LOCAL EXECUTIVE
6615  4477            LCMR            /FOR IOT DETERMINATION,
6616  45Ø1            LDBR
6617  45ØØ            LFGR
662Ø  4474    T1LS2,  CWCR
6621  4476            CCAR
6622  ØØØØ            Ø
6623  ØØØØ            Ø
6624  ØØØØ            Ø
6625  45Ø3    T1LS3,  RWCR
6626  7777            -1
6627  45Ø4            RCAR
663Ø  7777            -1
6631  45Ø6            RCMR
6632  7777            -1
6633  451Ø            RDBR
6634  7777            -1
6635  45Ø7            RFSR
6636  76ØØ            76ØØ
6637  45Ø3            RWCR
664Ø  7777            -1
6641  45Ø4            RCAR
6642  7777            -1
6643  45Ø6            RCMR
6644  7777            -1
6645  4512            RDBR
6646  7777            -1
              /
6647  4541    T3LS1,  READ            /FUNCTION LIST USED BY
```

```
      665Ø  4542              RDCOMP                        /TEST Ø3 ("T3EX1"),
      6651  4543              WRITE
      6652  4537              OFFLIN
      6653  454Ø              REWIND
      6654  4544              WEOF
      6655  4545              SPCFWD
      6656  4546              SPCREV
      6657  4541              READ
      666Ø  4542              RDCOMP
      6661  4543              WRITE
      6662  4544              WEOF
      6663  5276      SYMPT,  I1                            /SYMBOL ROUTINE POINTER TABLE,
      6664  53Ø4              I2
      6665  5214              GD
      6666  5221              BD
      6667  5226              OD
      667Ø  5233              WC
      6671  524Ø              CA
      6672  5245              CM
      6673  5252              FS
      6674  5257              MS
      6675  5264              DB
      6676  5271              AC
```

```
                    /ERROR TABLE FOR ERRORS OCCURRING OUTSIDE OF FORMAL TESTS, (CODE ØØX)

      6677  4ØØ1    FEØØA,  4ØØ1    /ILLEGAL INTERRUPT IN INTSEV,  CURRENT CONTENTS OF
      67ØØ  7776            7776    /LOCATION Ø IN BAD,
      67Ø1  ØØØ2    ERØØB,  ØØØ2    /BITS 4 AND 5 OF CM DID NOT CLEAR AFTER LCMRØ,
      67Ø2  1Ø2Ø            1Ø2Ø    /DETECTED IN INSRS,
      67Ø3  ØØØ3    ERØØC,  ØØØ3
      67Ø4  76Ø1            76Ø1

                    /ERROR TABLE FOR ERRORS OCCURRING IN FORMAL TESTS (CODE Ø1X-14X),

      67Ø5  Ø1Ø1    ERØ1A,  Ø1Ø1
      67Ø6  14ØØ            14ØØ
      67Ø7  Ø1Ø2    ERØ1B,  Ø1Ø2
      671Ø  14ØØ            14ØØ
      6711  Ø1Ø3    ERØ1C,  Ø1Ø3
      6712  14ØØ            14ØØ
      6713  Ø1Ø4    ERØ1D,  Ø1Ø4
      6714  54ØØ            54ØØ
      6715  Ø1Ø5    ERØ1E,  Ø1Ø5
      6716  6ØØØ            6ØØØ
      6717  Ø1Ø6    ERØ1F,  Ø1Ø6
      672Ø  74ØØ            74ØØ
      6721  Ø1Ø7    ERØ1G,  Ø1Ø7
      6722  6ØØØ            6ØØØ
      6723  Ø11Ø    ERØ1H,  Ø11Ø
      6724  74ØØ            74ØØ
      6725  Ø111    ERØ1I,  Ø111
      6726  74ØØ            74ØØ
      6727  Ø112    ERØ1J,  Ø112
      673Ø  34ØØ            34ØØ
      6731  Ø113    ERØ1K,  Ø113
      6732  6ØØØ            6ØØØ
      6733  Ø114    ERØ1L,  Ø114
      6734  74ØØ            74ØØ
      6735  Ø115    ERØ1M,  Ø115
      6736  54ØØ            54ØØ
      6737  Ø116    ERØ1N,  Ø116
      674Ø  74ØØ            74ØØ
      6741  Ø117    ERØ1O,  Ø117
      6742  74ØØ            74ØØ
      6743  Ø12Ø    ERØ1P,  Ø12Ø
      6744  74ØØ            74ØØ
      6745  22Ø1    ERØ2A,  22Ø1
      6746  13ØØ            13ØØ
      6747  22Ø2    ERØ2B,  22Ø2
      675Ø  124Ø            124Ø
      6751  22Ø3    ERØ2C,  22Ø3
      6752  122Ø            122Ø
      6753  22Ø4    ERØ2D,  22Ø4
      6754  12Ø2            12Ø2
      6755  22Ø5    ERØ2E,  22Ø5
      6756  16ØØ            16ØØ
      6757  Ø3Ø1    ERØ3A,  Ø3Ø1
```

```
6760  111Ø           111Ø
6761  Ø3Ø2    ERØ3B, Ø3Ø2
6762  1252           1252
6763  Ø3Ø3    ERØ3C, Ø3Ø3
6764  1652           1652
6765  Ø3Ø4    ERØ3D, Ø3Ø4
6766  1652           1652
6767  Ø3Ø5    ERØ3E, Ø3Ø5
677Ø  1252           1252
6771  Ø3Ø6    ERØ3F, Ø3Ø6
6772  111Ø           111Ø
6773  Ø3Ø7    ERØ3G, Ø3Ø7
6774  1Ø5Ø           1Ø5Ø
6775  Ø31Ø    ERØ3H, Ø31Ø
6776  111Ø           111Ø
6777  Ø311    ERØ3I, Ø311
7ØØØ  1Ø5Ø           1Ø5Ø
7ØØ1  24Ø1    ERØ4A, 24Ø1
7ØØ2  131Ø           131Ø
7ØØ3  Ø4Ø2    ERØ4B, Ø4Ø2
7ØØ4  1414           1414
7ØØ5  24Ø3    ERØ4C, 24Ø3
7ØØ6  127Ø           127Ø
7ØØ7  24Ø4    ERØ4D, 24Ø4
7Ø1Ø  127Ø           127Ø
7Ø11  Ø4Ø5    ERØ4E, Ø4Ø5
7Ø12  1Ø7Ø           1Ø7Ø
7Ø13  25Ø1    ERØ5A, 25Ø1
7Ø14  1252           1252
7Ø15  25Ø2    ERØ5B, 25Ø2
7Ø16  1652           1652
7Ø17  26Ø1    ERØ6A, 26Ø1
7Ø2Ø  1272           1272
7Ø21  Ø7Ø1    ERØ7A, Ø7Ø1
7Ø22  14ØØ           14ØØ
7Ø23  Ø7Ø2    ERØ7B, Ø7Ø2
7Ø24  14ØØ           14ØØ
7Ø25  Ø7Ø3    ERØ7C, Ø7Ø3
7Ø26  14ØØ           14ØØ
7Ø27  Ø7Ø4    ERØ7D, Ø7Ø4
7Ø3Ø  14ØØ           14ØØ
7Ø31  Ø7Ø5    ERØ7E, Ø7Ø5
7Ø32  14ØØ           14ØØ
7Ø33  1ØØ1    ER1ØA, 1ØØ1
7Ø34  ØØØØ           ØØØØ
7Ø35  1ØØ2    ER1ØB, 1ØØ2
7Ø36  Ø174           Ø174
7Ø37  1ØØ3    ER1ØC, 1ØØ3
7Ø4Ø  Ø174           Ø174
7Ø41  1ØØ4    ER1ØD, 1ØØ4
7Ø42  4174           4174
7Ø43  1ØØ5    ER1ØE, 1ØØ5
7Ø44  Ø174           Ø174
7Ø45  1ØØ6    ER1ØF, 1ØØ6
7Ø46  Ø174           Ø174
```

```
      7Ø47  1ØØ7     ER1ØG,  1ØØ7
      7Ø5Ø  Ø174             Ø174
      7Ø51  1Ø1Ø     ER1ØH,  1Ø1Ø
      7Ø52  1574             1574
      7Ø53  1Ø11     ER1ØI,  1Ø11
      7Ø54  1574             1574
      7Ø55  1Ø12     ER1ØJ,  1Ø12
      7Ø56  1574             1574
      7Ø57  1Ø13     ER1ØK,  1Ø13
      7Ø6Ø  1574             1574
      7Ø61  1Ø17     ER1ØO,  1Ø17
      7Ø62  1574             1574
      7Ø63  1Ø2Ø     ER1ØP,  1Ø2Ø
      7Ø64  1574             1574
      7Ø65  1Ø21     ER1ØQ,  1Ø21
      7Ø66  1574             1574
      7Ø67  1Ø22     ER1ØR,  1Ø22
      7Ø7Ø  1574             1574
      7Ø71  1Ø23     ER1ØS,  1Ø23
      7Ø72  1576             1576
      7Ø73  1Ø24     ER1ØT,  1Ø24
      7Ø74  1576             1576
      7Ø75  1Ø25     ER1ØU,  1Ø25
      7Ø76  1574             1574
      7Ø77  1Ø26     ER1ØV,  1Ø26
      71ØØ  1574             1574
      71Ø1  1Ø27     ER1ØW,  1Ø27
      71Ø2  1574             1574
      71Ø3  1Ø3Ø     ER1ØX,  1Ø3Ø
      71Ø4  1574             1574
      71Ø5  1Ø32     ER1ØZ,  1Ø32
      71Ø6  1576             1576
      71Ø7  1Ø61     ER1Ø1,  1Ø61
      711Ø  1574             1574
      7111  1Ø62     ER1Ø2,  1Ø62
      7112  1574             1574
      7113  11Ø1     ER11A,  11Ø1
      7114  ØØØ4             ØØØ4
      7115  11Ø2     ER11B,  11Ø2
      7116  4ØØ4             4ØØ4
      7117  11Ø3     ER11C,  11Ø3
      712Ø  4114             4114
      7121  11Ø4     ER11D,  11Ø4
      7122  4114             4114
      7123  11Ø5     ER11E,  11Ø5
      7124  4Ø14             4Ø14
      7125  11Ø6     ER11F,  11Ø6
      7126  4Ø14             4Ø14
      7127  12Ø1     ER12A,  12Ø1
      713Ø  ØØ24             ØØ24
      7131  12Ø2     ER12B,  12Ø2
      7132  Ø134             Ø134
      7133  12Ø3     ER12C,  12Ø3
      7134  Ø134             Ø134
      7135  12Ø4     ER12D,  12Ø4
```

```
        7136  ØØ34             ØØ34
        7137  12Ø5   ER12E,    12Ø5
        714Ø  ØØ34             ØØ34
        7141  13Ø1   ER13A,    13Ø1
        7142  ØØ14             ØØ14
        7143  13Ø2   ER13B,    13Ø2
        7144  4Ø14             4Ø14
        7145  13Ø3   ER13C,    13Ø3
        7146  1Ø4Ø             1Ø4Ø
        7147  13Ø4   ER13D,    13Ø4
        715Ø  1Ø5Ø             1Ø5Ø
        7151  13Ø5   ER13E,    13Ø5
        7152  511Ø             511Ø
        7153  34Ø1   ER14A,    34Ø1
        7154  1656             1656
        7155  34Ø2   ER14B,    34Ø2
        7156  1656             1656
        7157  14Ø3   ER14C,    14Ø3
        716Ø  1Ø56             1Ø56
        7161  14Ø4   ER14D,    14Ø4
        7162  14Ø4             14Ø4

                     /MESSAGES!

        7163  4324   MSG1,     TEXT    "#TM8E CONTROL TEST PART 1##MAINDEC-Ø8-DHTMA-A#"
        7164  157Ø
        7165  Ø54Ø
        7166  Ø317
        7167  1624
        717Ø  2217
        7171  144Ø
        7172  24Ø5
        7173  2324
        7174  4Ø2Ø
        7175  Ø122
        7176  244Ø
        7177  6143
        72ØØ  4315
        72Ø1  Ø111
        72Ø2  16Ø4
        72Ø3  45Ø3
        72Ø4  556Ø
        72Ø5  7Ø55
        72Ø6  Ø41Ø
        72Ø7  2415
        721Ø  Ø155
        7211  Ø143
        7212  ØØØØ

                     /MSG2,    TEXT    "#TEST NN#"
        7213  4324   MSG2,     4324                    /CR,T
        7214  Ø523             Ø523                    /E,S
        7215  244Ø             244Ø                    /T,SP
        7216  ØØØØ   M2P1,     ØØØØ                    /TEST NUMBER FILL IN,
        7217  43ØØ             43ØØ                    /*, END MSG
        7220  4323   MSG3,   TEXT    "#SELECTED TESTS DONE#"
        7221  Ø514
        7222  Ø5Ø3
        7223  24Ø5
        7224  Ø44Ø
        7225  24Ø5
        7226  2324
        7227  234Ø
        723Ø  Ø417
        7231  16Ø5
        7232  43ØØ

                     /MSG4,  TEXT    "#PASS NNNN#"
        7233  432Ø   MSG4,   432Ø                    /CR,P
        7234  Ø123           Ø123                    /A,S
        7235  234Ø           234Ø                    /S,SP
        7236  ØØØØ   M4P1,   Ø                       /PASS NUMBER
        7237  ØØØØ   M4P2,   Ø                       /FILL IN,
        724Ø  ØØØØ           ØØØØ
        7241  7743   MSG6,   TEXT    "?#"
        7242  ØØØØ

        7243  4343   MSG7,   TEXT    "##FATAL ERROR#"
        7244  Ø6Ø1
        7245  24Ø1
        7246  144Ø
        7247  Ø522
        725Ø  2217
        7251  2252
        7252  ØØØØ

        7253  4333   MSG8,   TEXT    "#["
        7254  ØØØØ

        7255  4323   MSG9,   TEXT    "#SET SRØ FOR QUICK VERIFY#"
        7256  Ø524
        7257  4Ø23
        726Ø  226Ø
        7261  4ØØ6
        7262  1722
        7263  4Ø21
        7264  2511
        7265  Ø313
        7266  4Ø26
        7267  Ø522
        727Ø  11Ø6
        7271  3143
        7272  ØØØØ

        7273  43Ø5   MSG1Ø,  TEXT    "#EXTENDED MEMORY?(Ø-7) "
        7274  3Ø24
        7275  Ø516
        7276  Ø4Ø5
        7277  Ø44Ø
```

```
7300  1505
7301  1517
7302  2231
7303  775Ø
7304  6055
7305  6751
7306  4ØØØ

              /MSG11, TEXT     "  NNNN"
7307  4040    MSG11,  4Ø4Ø                    /SP,SP
731Ø  ØØØØ    M11P1,  Ø                       /CONTENTS OF MEMORY
7311  ØØØØ    M11P2,  Ø                       /FILL IN.
7312  ØØØØ            Ø                       /END MSG.

              /THE FULL ERROR MESSAGE CALLED BY STA12=7777 WOULD BE:
              /ERMSG, TEXT     "#*ERNNX  PC:NNNN  I1: NNNN  I2: NNNN  GD: NNNN  BD: NNNN  OD: NNNN
              /                WC: NNNN  CA: NNNN  CM: NNNN  FS: NNNN  MS: NNNN  DB: NNNN  AC: NNNN#"
7313  ØØØØ    ERMSG,  Ø                       /SPACING AND SYMBOL FILL IN.
7314  ØØØØ    ERMP1,  Ø                       /ERROR TYPE
7315  ØØØØ    ERMP2,  Ø                       /TEST TYPE
7316  ØØØØ    ERMP3,  Ø                       /SUBTEST LETTER, SPACE
7317  4020            4Ø2Ø                    /SP,P
732Ø  Ø372            Ø372                    /C,:
7321  ØØØØ    ERMP4,  Ø                       /CONTENTS OF PC.
7322  ØØØØ    ERMP5,  Ø                       /DITTO
                      DECIMAL                 /ALL OTHER INFORMATION INSERTED HERE.
7323  ØØØØ            ZBLOCK  65
7324  ØØØØ            ZBLOCK  65
7325  ØØØØ            ZBLOCK  65
7326  ØØØØ            ZBLOCK  65
7327  ØØØØ            ZBLOCK  65
733Ø  ØØØØ            ZBLOCK  65
7331  ØØØØ            ZBLOCK  65
7332  ØØØØ            ZBLOCK  65
7333  ØØØØ            ZBLOCK  65
7334  ØØØØ            ZBLOCK  65
7335  ØØØØ            ZBLOCK  65
7336  ØØØØ            ZBLOCK  65
7337  ØØØØ            ZBLOCK  65
734Ø  ØØØØ            ZBLOCK  65
7341  ØØØØ            ZBLOCK  65
7342  ØØØØ            ZBLOCK  65
7343  ØØØØ            ZBLOCK  65
7344  ØØØØ            ZBLOCK  65
7345  ØØØØ            ZBLOCK  65
7346  ØØØØ            ZBLOCK  65
7347  ØØØØ            ZBLOCK  65
735Ø  ØØØØ            ZBLOCK  65
7351  ØØØØ            ZBLOCK  65
7352  ØØØØ            ZBLOCK  65
7353  ØØØØ            ZBLOCK  65
7354  ØØØØ            ZBLOCK  65
7355  ØØØØ            ZBLOCK  65
7356  ØØØØ            ZBLOCK  65
7357  ØØØØ            ZBLOCK  65
```

```
        7360  ØØØØ              ZBLOCK  65
        7361  ØØØØ              ZBLOCK  65
        7362  ØØØØ              ZBLOCK  65
        7363  ØØØØ              ZBLOCK  65
        7364  ØØØØ              ZBLOCK  65
        7365  ØØØØ              ZBLOCK  65
        7366  ØØØØ              ZBLOCK  65
        7367  ØØØØ              ZBLOCK  65
        737Ø  ØØØØ              ZBLOCK  65
        7371  ØØØØ              ZBLOCK  65
        7372  ØØØØ              ZBLOCK  65
        7373  ØØØØ              ZBLOCK  65
        7374  ØØØØ              ZBLOCK  65
        7375  ØØØØ              ZBLOCK  65
        7376  ØØØØ              ZBLOCK  65
        7377  ØØØØ              ZBLOCK  65
        74ØØ  ØØØØ              ZBLOCK  65
        74Ø1  ØØØØ              ZBLOCK  65
        74Ø2  ØØØØ              ZBLOCK  65
        74Ø3  ØØØØ              ZBLOCK  65
        74Ø4  ØØØØ              ZBLOCK  65
        74Ø5  ØØØØ              ZBLOCK  65
        74Ø6  ØØØØ              ZBLOCK  65
        74Ø7  ØØØØ              ZBLOCK  65
        741Ø  ØØØØ              ZBLOCK  65
        7411  ØØØØ              ZBLOCK  65
        7412  ØØØØ              ZBLOCK  65
        7413  ØØØØ              ZBLOCK  65
        7414  ØØØØ              ZBLOCK  65
        7415  ØØØØ              ZBLOCK  65
        7416  ØØØØ              ZBLOCK  65
        7417  ØØØØ              ZBLOCK  65
        742Ø  ØØØØ              ZBLOCK  65
        7421  ØØØØ              ZBLOCK  65
        7422  ØØØØ              ZBLOCK  65
        7423  ØØØØ              ZBLOCK  65
                                ZBLOCK  65
                                                        /(DONE BY ASSEMBLER.)

                                OCTAL


              75ØØ      *75ØØ
        75ØØ  4472      T1ØLS1, 167ØØ
        75Ø1  4473              LWCR
        75Ø2  4475              LCAR
        75Ø3  4477              LCMR
        75Ø4  45ØØ              LFGR
        75Ø5  4474              CWCR
        75Ø6  4476              CCAR
        75Ø7  45Ø3              RWCR
        751Ø  45Ø4              RCAR
        7511  45Ø6              RCMR
        7512  45Ø7              RFSR
        7513  451Ø              RDBR
        7514  4505          RMSR
        7515  4502          I671Ø
        7516  4511          I672Ø
        7517  4515          SKTR
        752Ø  4514          SKTD
        7521  4513          SKCB
        7522  452Ø          SDLE
        7523  4516          CLF
        7524  4512          SKEF
```

```
            /INSTRUCTION DEFINITIONS:

            /SPECIAL EXT MEM BASIC INSTRUCTIONS:

6201        CDF=6201               /CHANGE DATA FIELD
6224        RIF=6224               /READ INSTRUCTION FIELD.
6244        RMF=6244               /RESTORE MEMORY FIELD.

            /TM8E IOT DEFINITIONS:

                    /LOAD IOT'S:

4472        I67ØØ=JMS I S67ØØP     /IOT 67ØØ,
4473        LWCR=JMS I SLWCRP              /AC TO WC, Ø TO AC,
4474        CWCR=JMS I SCWCRP              /Ø TO WC,
4475        LCAR=JMS I SLCARP              /AC TO CA, Ø TO AC,
4476        CCAR=JMS I SCCARP              /Ø TO CA,
4477        LCMR=JMS I SLCMRP              /AC TO CM, Ø TO AC,
45ØØ        LFGR=JMS I SLFGRP              /ACØ-3, 5 TO FR, Ø TO AC,
45Ø1        LDBR=JMS I SLDBRP              /AC TO DB, Ø TO AC

                    /READ IOT'S:

45Ø2        I671Ø=JMS I S671ØP     /IOT 671Ø,
45Ø3        RWCR=JMS I SRWCRP              /Ø TO AC, WC TO AC,
45Ø4        RCAR=JMS I SRCARP              /Ø TO AC, CA TO AC,
45Ø5        RMSR=JMS I SRMSRP              /Ø TO AC, MS TO AC,
45Ø6        RCMR=JMS I SRCMRP              /Ø TO AC, CM TO AC,
45Ø7        RFSR=JMS I SRFSRP              /Ø TO AC, FR TO ACØ-4, GO BIT TO AC5, S1 TO AC6-11.
451Ø        RDBR=JMS I SRDBRP              /Ø TO AC, DB TO AC,

                    /CONTROL IOT'S:

4511        I672Ø=JMS I S672ØP     /IOT 672Ø,
4512        SKEF=JMS I SSKEFP              /SKIP IF ERROR FLAG SET,
4513        SKCB=JMS I SSKCBP              /SKIP IF CONTROL NOT BUSY,
4514        SKTD=JMS I SSKTDP              /SKIP IF TRANSPORT DONE.
4515        SKTR=JMS I SSKTRP              /SKIP IF TUR,
4516        CLF=JMS I SCLFP                /Ø TO REGISTERS AND FLAGS IF TUR; IF NOT Ø TO MTF, EF, SR
4517        CLT=JMS I SCLTP        /POWER CLEAR TRANSPORT,
452Ø        SDLE=JMS I SSDLEP              /SET DATA LATE ERROR,
4521        SBRM=JMS I SSBRMP              /SET ONE BREAK REQUEST,

            /INSTRUCTION PSEUDO MNEMONICS DEFINITIONS,

4522        BSW=JMS I SRBSWP               /SWAP BITS Ø-5 WITH BITS 6-11 IN THE AC.
4523        CLEAR1=JMS I CLR1P     /GENERATE "INITIALIZE" AND LCMRØ, TURN INTERRUPT ON,
4524        CLEAR2=JMS I CLR2P     /GENERATE "INITIALIZE", LEAVE CM ALONE, TURN INTERRUPT ON,
4525        CLEAR3=JMS I CLR3P     /GENERATE "INITIALIZE" SAFELY WITH RESPECT TO TTO,  DO NOT TURN
                                   /INTERRUPT ON,
4526        LOOP5=JMS I LAS5P      /CHECK SR5,  IF CLEAR, SKIP NEXT INSTRUCTION.
4527        LOOP6=JMS I LAS6P      /CHECK SR6,  IF CLEAR, SKIP NEXT INSTRUCTION.
453Ø        LOOP7=JMS I LAS7P      /CHECK SR7,  IF CLEAR, SKIP NEXT INSTRUCTION.
4531        LOOP8=JMS I LAS8P      /CHECK SR8,  IF CLEAR, SKIP NEXT INSTRUCTION.
```

```
            4532    LOOP9=JMS I LAS9P       /CHECK SR9, IF CLEAR, SKIP NEXT INSTRUCTION,
            4533    COMPAR=JMS I COMPP      /COMPARE "GOOD" AND "BAD".  GO TO "ERRORS" ON FAILURE,  CHECK LOOP 5,
            4534    ERROR=JMS I ERRORP      /GO DIRECTLY TO ERROR ROUTINE,  MAY BEUSED FOR FATAL ERRORS ONLY,
            4535    BREAK=JMS I DATBRP      /ISSUE ONE BREAK REQUEST (SKRM),
            4536    COMAND=JMS I CMDP       /LOAD THE CM WITH THE CONSTANT IN THE NEXT LOCATION,
            4537    OFFLIN=JMS I OFFLNP     /ØØØØ+GAP CONSTANT TO FR,
            454Ø    REWIND=JMS I REWNP      /1ØØØ+GAP CONSTANT FR,
            4541    READ=JMS I READOP       /2ØØØ+GAP CONSTANT TO FR,
            4542    RDCOMP=JMS I RDCMPP     /3ØØØ+GAP CONSTANT TO FR,
            4543    WRITE=JMS I WRITP       /4ØØØ+GAP CONSTANT TO FR,
            4544    WEOF=JMS I EOFP         /5ØØØ+GAP CONSTANT TO FR,
            4545    SPCFWD=JMS I SPFORP     /6ØØØ+GAP CONSTANT TO FR,
            4546    SPCREV=JMS I SPREVP     /7ØØØ+GAP CONSTANT TO FR,
            4547    EXTGAP=JMS I XGAPP      /SET GAP CONSTANT TO Ø4ØØ,  CONSTANT CLEARED AFTER NEXT FR LOAD,
            455Ø    GO=JMS I GOEP           /SET THE "GO" BIT AND LEAVE REST OF FR ALONE,
            4553    WCOV=JMS I WCOVEP       /GENERATE WORD COUNT OVERFLOW USING FUNCTION ALREADY IN FR,
            4551    LXXR=JMS I SIOT1P       /IOT SUBROUTINES FOR TEST Ø1,
            4551    CXXR=JMS I SIOT1P
            4552    RXXR=JMS I SIOT2P
                    $$$$
```

```
ØØØØ  111111ØØ  ØØØØØØØØ  11111111  11111111  11111111  11111111  11111111  11111111
Ø1ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

Ø2ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
Ø3ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

Ø4ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
Ø5ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111ØØØ

Ø6ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
Ø7ØØ  11111111  11111111  11111111  11111111  111111ØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ


1ØØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
11ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

12ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
13ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

14ØØ  11111111  11111111  11111111  11111111  11111111  11ØØØØØØ  ØØØØØØØØ  ØØØØØØØØ
15ØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

16ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
17ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  ØØØØØØØØ


2ØØØ  11111111  11111111  11111111  11111111  11111111  11ØØØØØØ  ØØØØØØØØ  ØØØØØØØØ
21ØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

22ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
23ØØ  11111111  11111111  11111111  11111111  11111111  11111111  1111ØØØØ  ØØØØØØØØ

24ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
25ØØ  11111111  11111111  11111111  11111111  11111111  111ØØØØØ  ØØØØØØØØ  ØØØØØØØØ

26ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
27ØØ  111111ØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ


3ØØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
31ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111ØØØ

32ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
33ØØ  11111111  11111111  11111111  11111111  111111ØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

34ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
35ØØ  11111ØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

36ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
37ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
```

```
4ØØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
41ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

42ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
43ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  ØØØØØØØØ

44ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
45ØØ  11111111  11111111  11111111  11111111  11111ØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

46ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
47ØØ  11111111  11111111  11111111  11111111  11111111  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ


5ØØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
51ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  1111ØØØØ

52ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
53ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  1ØØØØ111

54ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
55ØØ  11111111  11111111  11111111  1111111Ø  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

56ØØ  11111111  11111111  11111111  11111111  111ØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ
57ØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ


6ØØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
61ØØ  11111111  11111111  11111111  11111111  1111ØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

62ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
63ØØ  11111111  11111111  11111111  1111ØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

64ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
65ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

66ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
67ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111


7ØØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
71ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

72ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
73ØØ  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

74ØØ  11111111  11111111  1111ØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ
75ØØ  11111111  11111111  11111ØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ

76ØØ  11111111  11111111  11111111  11111111  1ØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ
77ØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ  ØØØØØØØØ
```

```
AC      5271      EDIT1   6312      ERØ7D   7Ø27      ERMP4   7321
ACLOC   Ø170      EDITP   ØØ63      ERØ7E   7Ø31      ERMP4P  5152
ACTFLG  Ø175      EDTEM   6326      ER1Ø1   71Ø7      ERMP5   7322
AMG8E   622Ø      EDTEM1  6327      ER1Ø2   7111      ERMP5P  5153
AMG8E1  6237      EDTEM2  633Ø      ER1ØA   7Ø33      ERMPP   5363
AMG8E2  6214      EDTEM3  6331      ER1ØB   7Ø35      ERMPTP  5156
AMG8E3  624Ø      EDTEMP  ØØ64      ER1ØC   7Ø37      ERMSG   7313
AMG8EP  ØØ61      EOFP    Ø144      ER1ØD   7Ø41      ERMSGP  ØØ67
BAD     Ø155      EOFW    6456      ER1ØE   7Ø43      ERMSYM  5364
BD      5221      ERØØB   67Ø1      ER1ØF   7Ø45      ERR1P   5636
BELL    6262      ERØØC   67Ø3      ER1ØG   7Ø47      ERROR   4534
BELLP   ØØ65      ERØ1A   67Ø5      ER1ØH   7Ø51      ERRORP  Ø134
BREAK   4535      ERØ1B   67Ø7      ER1ØI   7Ø53      ERRORS  5ØØØ
BSW     4522      ERØ1C   6711      ER1ØJ   7Ø55      ERRPTR  5145
CA      524Ø      ERØ1D   6713      ER1ØK   7Ø57      ES      4527
CCAR    4476      ERØ1E   6715      ER1ØO   7Ø61      EXEC    42ØØ
CDF     62Ø1      ERØ1F   6717      ER1ØP   7Ø63      EXECFP  ØØ71
CLEAR1  4523      ERØ1G   6721      ER1ØQ   7Ø65      EXECFX  55Ø7
CLEAR2  4524      ERØ1H   6723      ER1ØR   7Ø67      EXECL1  4256
CLEAR3  4525      ERØ1I   6725      ER1ØS   7Ø71      EXECL2  4245
CLF     4516      ERØ1J   6727      ER1ØT   7Ø73      EXECP   ØØ55
CLR1    64ØØ      ERØ1K   6731      ER1ØU   7Ø75      EXITFL  Ø174
CLR1P   Ø123      ERØ1L   6733      ER1ØV   7Ø77      EXOMT   43Ø3
CLR2    64Ø6      ERØ1M   6735      ER1ØW   71Ø1      EXTAL   4354
CLR2P   Ø124      ERØ1N   6737      ER1ØX   71Ø3      EXTEMP  4362
CLR3    6414      ERØ1O   6741      ER1ØZ   71Ø5      EXTGAP  4547
CLR3P   Ø125      ERØ1P   6743      ER11A   7113      FATERR  5Ø31
CLT     4517      ERØ2A   6745      ER11B   7115      FEØØA   6677
CM      5245      ERØ2B   6747      ER11C   7117      FS      5252
CMD     6525      ERØ2C   6751      ER11D   7121      FUNC    6511
CMDP    Ø136      ERØ2D   6753      ER11E   7123      FUNCP   ØØ56
COMAND  4536      ERØ2E   6755      ER11F   7125      FUNCT   6524
COMP    56ØØ      ERØ3A   6757      ER12A   7127      GD      5214
COMPAR  4533      ERØ3B   6761      ER12B   7131      GO      455Ø
COMPF   5631      ERØ3C   6763      ER12C   7133      GOE     65ØØ
COMPP   Ø133      ERØ3D   6765      ER12D   7135      GOEP    Ø15Ø
CRLF    627Ø      ERØ3E   6767      ER12E   7137      GOOD    Ø154
CRLFP   ØØ66      ERØ3F   6771      ER13A   7141      HALTC   5211
CS      442Ø      ERØ3G   6773      ER13B   7143      HALTCP  5213
CWCR    4474      ERØ3H   6775      ER13C   7145      I1      5276
CXXR    4551      ERØ3I   6777      ER13D   7147      I1I2    5312
DATBRK  6534      ERØ4A   7ØØ1      ER13E   7151      I1P     3145
DATBRP  Ø135      ERØ4B   7ØØ3      ER14A   7153      I2      53Ø4
DB      5264      ERØ4C   7ØØ5      ER14B   7155      I67ØØ   4472
DCR     4665      ERØ4D   7ØØ7      ER14C   7157      I671Ø   45Ø2
DLF     4636      ERØ4E   7Ø11      ER14D   7161      I672Ø   4511
DLF1    464Ø      ERØ5A   7Ø13      ERMP1   7314      INSRS   5463
DLF2    4621      ERØ5B   7Ø15      ERMP1P  5146      INSRSF  55Ø5
DS      46ØØ      ERØ6A   7Ø17      ERMP2   7315      INSRSP  ØØ6Ø
DS1     4667      ERØ7A   7Ø21      ERMP2P  5147      INTAGN  54Ø7
DS2     467Ø      ERØ7B   7Ø23      ERMP3   7316      INTEF   Ø164
EDIT    63Ø1      ERØ7C   7Ø25      ERMP3P  515Ø      INTFLG  Ø172
```

```
INTMTF   Ø166          KRMF     4347          M4P2P    4363          RDCMP    6446
INTNGA   5432          KXGAP    6477          M5       ØØ41          RDCMPP   Ø142
INTOK    5444          LAS5     6105          MARK1    1622          RDCOMP   4542
INTOKP   4544          LAS5P    Ø126          MARK1P   1762          READ     4541
INTSEP   4345          LAS6     6113          MARK2    1627          READD    6442
INTSEV   54ØØ          LAS6P    Ø127          MARK2P   1763          READDP   Ø141
IOT1     Ø566          LAS7     6121          MARK3    1624          RESTP    6Ø21
IOT1P    53Ø3          LAS7P    Ø13Ø          MEMFLD   Ø167          RESTPP   4541
IOT2     Ø571          LAS8     6127          MONIT    44ØØ          REWIND   454Ø
IOT2P    5311          LAS8P    Ø131          MONITP   ØØ62          REWND    6436
K1       ØØ2Ø          LAS9     6135          MONP     4676          REWNP    Ø14Ø
K1Ø      ØØ23          LAS9P    Ø132          MS       5257          REX      424Ø
K1ØØ     ØØ27          LCAR     4475          MSG1     7163          REXP     4536
K1ØØØ    5366          LCMR     4477          MSG1Ø    7273          RFSR     45Ø7
K17      5165          LDBR     45Ø1          MSG11    73Ø7          RIF      6224
K2       ØØ21          LFGR     45ØØ          MSG2     7213          RMF      6244
K2Ø      ØØ24          LISN     4677          MSG3     722Ø          RMSR     45Ø5
K2ØØ     ØØ3Ø          LISN1    4712          MSG4     7233          RWCR     45Ø3
K2ØØØ    ØØ33          LISN2    4724          MSG6     7241          RXXR     4552
K2Ø7     6267          LISN3    474Ø          MSG7     7243          S67ØØ    4ØØØ
K212     63ØØ          LISNP    4543          MSG8     7253          S67ØØP   ØØ72
K215     ØØ31          LISNT1   4746          MSG9     7255          S671Ø    4Ø4Ø
K2252    5641          LISNUM   4727          NOEXLP   4336          S671ØP   Ø1Ø2
K3       ØØØ5          LOOP5    4526          NOTSPK   4277          S672Ø    4Ø74
K377     537Ø          LOOP6    4527          NOTSRN   4315          S672ØP   Ø111
K3774    2Ø51          LOOP7    453Ø          OD       5226          SAVEAC   Ø162
K4       6143          LOOP8    4531          OFFLIN   4537          SAVEL    Ø163
K4Ø      ØØ25          LOOP9    4532          OFFLNE   6432          SAVEP    6ØØØ
K4ØØ     ØØ32          LWCR     4473          OFFLNP   Ø137          SAVEP1   6Ø43
K4ØØØ    ØØ34          LXXR     4551          OLD      Ø156          SAVEP2   6Ø44
K4Ø4Ø    ØØ35          M1       5461          PØ       5462          SAVEP3   6Ø45
K43ØØ    5166          M11P1    731Ø          P1       4351          SAVEP4   6Ø46
K4324    5642          M11P1P   4671          P2       4352          SAVEPP   454Ø
K4343    5167          M11P2    7311          P3       4353          SBRM     4521
K4352    517Ø          M11P2P   4672          PASCNT   4346          SCCAR    4Ø2Ø
K522     5171          M14      ØØ42          PRCNT1   5154          SCCARP   ØØ76
K6Ø5     5172          M142     546Ø          PRCNT2   5155          SCDF     4144
K6Ø6Ø    6332          M16      6Ø42          PRGFLD   Ø165          SCDFI    4145
K7       ØØ22          M2       ØØ37          PRMPT    516Ø          SCDFIP   435Ø
K7Ø      ØØ26          M2Ø      ØØ43          PRNT     5Ø53          SCDFP    ØØ57
K7ØØ     6333          M233     5457          PRNTCK   5135          SCLF     4124
K72      5367          M24Ø     4747          PRNTK    5161          SCLFP    Ø116
K74ØØ    ØØ36          M26Ø     ØØ44          PRNTLP   5117          SCLT     413Ø
K76ØØ    Ø574          M27Ø     ØØ45          PRNTP    5637          SCLTP    Ø117
K76ØØA   651Ø          M2P1     7216          QHALT    52ØØ          SCWCR    4Ø1Ø
K77      6241          M2P1P    436Ø          QHALTP   5164          SCWCRP   ØØ74
K77ØØ    5173          M4       ØØ4Ø          QPRNT    5Ø42          SDLE     452Ø
KCDF     ØØ46          M4Ø      6242          QUES     4673          SETEF    3157
KJMP61   1757          M43      6243          QUESP    4542          SETEFP   3345
KJMP62   176Ø          M4P1     7236          RCAR     4574          SIOT1    Ø565
KJMPI3   4344          M4P1P    4364          RCMR     4576          SIOT1P   Ø151
KNOP     1756          M4P2     7237          RDBR     4512          SIOT2    Ø572
```

```
SIOI2P  Ø152        START   Ø2ØØ        TØ4T1   1376        T11D    3245
SKCB    4513        SYMADR  5151        TØ4T2   1377        T11D1   3Ø55
SKEF    4512        SYMBOL  5157        TØ5A    14Ø2        T11E    3Ø66
SKIPER  4147        SYMPT   6663        TØ5ASL  1436        T11E1   3Ø74
SKTD    4514        SYMPTP  5162        TØ5B    1424        T11EX1  3122
SKTR    4515        SYMSEV  533Ø        TØ5BSL  1427        T11EX2  3141
SLCAR   4Ø14        SYMTEM  5365        TØ5LDR  1451        T11EX3  3134
SLCARP  ØØ75        TØ1A    Ø211        TØ5LDS  145Ø        T11F    31Ø2
SLCMR   4Ø24        TØ1B    Ø216        TØ6A    16Ø2        T11F1   311Ø
SLCMRP  ØØ77        TØ1C    Ø223        TØ6ADF  1615        T11LS1  3146
SLDBR   4Ø34        TØ1D    Ø235        TØ6AMP  1623        T12A    32Ø7
SLDBRP  Ø1Ø1        TØ1E    Ø243        TØ7A    2ØØ2        T12A1   3215
SLFGR   4Ø3Ø        TØ1F    Ø255        TØ7B    2Ø11        T12B    3225
SLFGRP  Ø1ØØ        TØ1G    Ø27Ø        TØ7C    2Ø21        T12C    324Ø
SLWCR   4ØØ4        TØ1H    Ø3Ø4        TØ7D    2Ø32        T12D    3256
SLWCRP  ØØ73        TØ1I    Ø313        TØ7E    2Ø37        T12E    3267
SPCFWD  4545        TØ1J    Ø323        T1Ø1    2637        T12EX1  33Ø5
SPCREV  4546        TØ1K    Ø336        T1Ø2    266Ø        T12EX2  3322
SPFORD  6462        TØ1L    Ø347        T1ØA    22Ø4        T12EX3  3317
SPFORP  Ø145        TØ1M    Ø355        T1ØB    2216        T12EX4  3326
SPREV   6466        TØ1N    Ø362        T1ØC    2231        T12INT  3335
SPREVP  Ø146        TØ1O    Ø4Ø3        T1ØD    2244        T12LS1  3337
SRBSW   6547        TØ1OL   Ø4ØØ        T1ØE    2264        T13A    34Ø2
SRBSWN  6562        TØ1P    Ø427        T1ØF    2273        T13B    3415
SRBSWP  Ø122        TØ2A    Ø6Ø2        T1ØG    23Ø2        T13B1   3424
SRBSWT  6561        TØ2B    Ø617        T1ØH    2313        T13C    3434
SRCAR   4Ø5Ø        TØ2C    Ø634        T1ØI    2324        T13D    3447
SRCARP  Ø1Ø4        TØ2D    Ø654        T1ØJ    2336        T13E    3461
SRCMR   4Ø6Ø        TØ2E    Ø67Ø        T1ØK    235Ø        T13E1   3471
SRCMRP  Ø1Ø6        TØ2LDM  Ø745        T1ØKP   2552        T13EX1  35Ø3
SRDBR   4Ø7Ø        TØ2LDR  Ø731        T1ØLP8  2546        T13EX2  35Ø4
SRDBRP  Ø11Ø        TØ2LDS  Ø7Ø7        T1ØLPP  27Ø5        T14A    36Ø2
SRFSR   4Ø64        TØ2LDT  Ø716        T1ØLS1  75ØØ        T14B    3612
SRFSRP  Ø1Ø7        TØ2LDU  Ø714        T1ØO    24ØØ        T14BIT  3777
SRHOLD  6Ø47        TØ2LP9  Ø7Ø4        T1ØP    241Ø        T14C    3623
SRMSR   4Ø54        TØ3A    1ØØ4        T1ØQ    2421        T14C1   3637
SRMSRP  Ø1Ø5        TØ3B    1Ø15        T1ØR    243Ø        T14D    365Ø
SRPTS   6Ø66        TØ3C    1Ø27        T1ØS    2443        T14EX1  3671
SRWCR   4Ø44        TØ3D    1Ø42        T1ØT    2454        T14EX2  3677
SRWCRP  Ø1Ø3        TØ3E    1Ø62        T1ØU    25ØØ        T14EX3  37Ø1
SSBRM   414Ø        TØ3F    11Ø1        T1ØV    2511        T14EX4  371Ø
SSBRMP  Ø121        TØ3G    112Ø        T1ØW    2531        T14EX5  3725
SSDLE   4134        TØ3H    1141        T1ØX    26Ø2        T14EX6  3727
SSDLEP  Ø12Ø        TØ3I    1156        T1ØX1P  2361        T14EX7  3747
SSKCB   41Ø5        TØ4A    1237        T1ØX2P  2362        T14EX8  376Ø
SSKCBP  Ø113        TØ4AL   1243        T1ØX3P  2363        T14TM1  3775
SSKEF   41ØØ        TØ4ASL  1252        T1ØZ    2613        T14TM2  3776
SSKEFP  Ø112        TØ4B    1264        T11A    3ØØ3        T1EX1   Ø45Ø
SSKTD   4112        TØ4CD   1276        T11B    3Ø11        T1EX1A  Ø461
SSKTDP  Ø114        TØ4CL   13Ø2        T11B1   3Ø16        T1EX1P  Ø375
SSKTR   4117        TØ4CSL  1324        T11C    3Ø27        T1EX2   Ø473
SSKTRP  Ø115        TØ4E    1357        T11C1   3Ø37        T1EX2P  Ø376
```

```
T1EX3    0501           TEST04   1235
T1EX3P   0377           TEST05   1400
T1EX4    0505           TEST06   1600
T1EX5    0540           TEST07   2000
T1EX6    0524           TEST10   2200
T1EX7    0516           TEST11   3000
T1EX8    0530           TEST12   3200
T1EXT1   0561           TEST13   3400
T1EXT2   0562           TEST14   3600
T1EXT3   0563           TET1P    4366
T1EXT4   0564           TR       4501
T1LP7    0371           TRACE    0171
T1LP7P   0573           TRPP     5640
T1LS1    6613           TS       4435
T1LS1P   0556           TST      6563
T1LS2    6620           TST10P   2704
T1LS2P   0557           TST1P    0447
T1LS3    6625           TSTAEX   4357
T1LS3P   0560           TSTAT    0176
T3EX1    1213           TSTATM   4537
T3EX1P   1176           TSTEM1   4535
T3EX2    1231           TSTNUM   0177
T3EX2P   1175           TSTP     4365
T3EX3    1225           TSTPP    4355
T3EX32   1177           TSTTAL   4361
T3LOAD   1200           TTOFLG   0173
T3LODP   2004           TXXTM1   0157
T3LS1    6647           TXXTM2   0160
T4LP6    1262           TXXTM3   0161
T4LP7    1373           TYPE     6244
T5LP6    1422           TYPEP    0070
T5LP7    1445           TYPTEM   6261
T6CNT1   1764           WC       5233
T6CNT2   1765           WCOV     4553
T6CNT3   1766           WCOVEP   0153
T6EX1    1642           WCOVER   6541
T6EX2    1664           WEOF     4544
T6EX3    1727           WHAT     5163
T6EX4    1747           WRIT     6452
T6EX5    1737           WRITE    4543
T6EX6    1753           WRITP    0143
T6FLAG   1767           XBUFF    0050
T6I1     1713           XBUFFP   0047
T6SAVE   1761           XGAP     6472
TA       4517           XGAPP    0147
TAL      6577
TALP     4367
TALPP    4356
TCR      4513
TEST01   0203
TEST02   0600
TEST03   1000
```

```
ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME:  35 SECONDS

4K CORE USED
```