IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DHTMA-A-D

PRODUCT NAME: TM8-E CONTROL TEST PART 1

DATE CREATED: DECEMBER 4, 1972

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: LEONARD E. BEYERSDORFER

NOTE

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

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

TM8-E DIAGNOSTIC PROGRAMS' ORDER OF EXECUTION

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

TABLE OF CONTENTS

1. ABSTRACT

THE TM8-E CONTROL TEST PART 1 IS AN INTEGRATED SERIES OF SUB-TESTS 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 0200.

B. LOAD THE EXTENDED ADDRESS WITH THE PROGRAM FIELD.

C. SET THE SR TO 4000.

D. DEPRESS CLEAR, THEN CONTINUE.

E. THE PROGRAM WILL PRINT ITS TITLE AND MAINDEC NUMBER, THEN ASK "EXTENDED MEMORY?(0-7)". 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 0201. IN THIS CASE THE PROGRAM BYPASSES ALL PROGRAM INITIALIZATION AND GOES DIRECTLY TO THE PROGRAM MONITOR TO ACCEPT KEYBOARD COMMANDS.

6. STANDARD TEST PROCEDURE

USE OF THE STANDARD TEST PROCEDURE ENSURES PROPER TM8-E 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 TU10 TRANSPORT SYSTEM IS CONNECTED, ENSURE THAT ALL DRIVES ARE OFF LINE. ALL LEVELS INPUT FROM THE TU10 TO THE TM8-E SHOULD BE AT A "HIGH" LEVEL.

C. SET THE SR=0001.

D. TYPE "TA-" WHICH RESULTS IN THE EXECUTION OF TEST 01 THROUGH TEST 14. SEVERAL PASSES WILL BE MADE OF EACH TEST.

E. AFTER ALL TESTS HAVE BEEN EXECUTED, THE PROGRAM WILL PRINT "PASS 0001". 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)
T01T10TNN↵	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.
DMNNNN↵	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 SELECTED TEST).
1	0	ENABLE TEST IN PROGRESS REPORTS, AS SOON AS A TEST IS STARTED THE TEST NUMBER (TESTNN) IS PRINTED ON THE TTY.
	1	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 ALLOW THE USER TO LOOP ON A SMALL GROUP OF SUBTESTS WITHIN THE CURRENT TEST THEREBY ENABLING INCREASING THE TESTING FREQUENCY OF A SMALL SET OF FUNCTIONS. THE LISTING MUST BE REFERENCED TO DETERMINE WHICH SUBTESTS ARE INCLUDED IN EACH LOOP. THE END OF EACH LOOP IS MARKED AS "******LOOP N******".
10	0	INHIBIT LOOP ON CURRENT TEST (TEST 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 ENTIRE TEST SELECTION, THE NUMBER OF PASSES EXECUTED (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=0, THE PROGRAM WILL STOP TESTING AND RETURN TO THE PROGRAM MONITOR TO AWAIT A USER COMMAND. REFER TO PARAGRAPH 7.1 FOR AVAILABLE COMMANDS.

8.2 ERROR REPORTS

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

```
*ER01D  PC:0250 I1:6706 I2:6716 GD:0000 BD:7777 OD:7777
WC:1234 CA:2345 CM:0000 FS:0000 MS:0000 DB:0000 AC:0000
```

THE SYMBOLS USED IN THE ERROR REPORTS ARE DEFINED BELOW.

SYMBOL	DEFINITION
*ERNNX	ERROR OCCURRED IN TEST NN, SUBTEST X. (IF NN=00, 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.

10. EXECUTION TIME

ONE NORMAL PASS OF ALL TESTS TAKES APPROXIMATELY 15 MINUTES.
A QUICK VERIFY PASS (SR0=1) TAKES ONLY 2 MINUTES.

11. TEST ABSTRACTS

TEST 01 VERIFIES THE EXISTENCE OF THE VARIOUS 670X 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 02 IS A WORST CASE DATA TEST. DATA IS LOADED INTO AND READ BACK FROM ALL NON-STATUS REGISTER BITS AND VERIFIED.

TEST 03 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 04 TESTS THE ABILITY OF WORD COUNT, CURRENT ADDRESS AND COMMAND REGISTER BITS 6-8 TO INCREMENT PROPERLY.

TEST 05 VERIFIES THAT PROPER DATA TRANSFERS ARE MADE BETWEEN THE DATA BUFFER AND MEMORY IN BOTH DIRECTIONS.

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-08-DHTMA-A-L

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

                        /*****************************************************************
                        /       WARNING:        THIS PROGRAM SHOULD NOT BE STOPPED UNLESS
                        /                       MONITOR HAS TYPED "C",
                        /*****************************************************************

                        /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 200,
                                /CLEAR ALL SWITCHES,
                                /SET SR BIT 0 IF IT IS DESIRED TO SELECT TESTS WITH THE MONITOR,
                                /IF SR0 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
                                /0:     0       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:     0       TEST PROGRESS REPORTS,
                                /       1       NO TEST PROGRESS REPORTS,
                                /2:     0       GO TO MONITOR ON ANY ERROR OCCURRENCE,
                                /       1       GO TO MONITOR ONLY UPON FATAL ERROR,
                                /3:     0       PRINT ERROR MESSAGES AND IF "TR" IS PENDING ENABLE TRACE MESSAGES,
                                /       1       PRINT ONLY FATAL ERROR MESSAGES,
                                /4:     0       ERROR MESSAGES DISPLAY NECESSARY DATA ONLY,
                                /       1       ERROR MESSAGES DISPLAY ALL DATA WHETHER APPLICABLE
                                /               OR NOT, THIS SWITCH SHOULD BE USED ONLY
```

```
        /               IF TEST 01 PASSES,
/NOTE:  IF SR5=1 WHEN EXITING MONITOR TO RUN A NEW TEST SELECTION, THE
/PROGRAM MAY HANG UP IN THE ER08B TEST, RELEASING THIS SWITCH MOMENTARILY
/WILL ALLOW THE PROGRAM TO CONTINUE WITH THE NEW SELECTED TESTS,
        /5!     0       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:   0       NO LOOP 6-9 RESPECTIVELY,
                                /       1       LOOP 6-9 RESPECTIVELY,  (LOOP ON SMALL GROUP
                                /               OF SUBTESTS),
                                /10:    0       NO LOOP ON CURRENT TEST (T01-T14),
                                /       1       LOOP ON CURRENT TEST (T01-T14),
                                /11:    0       NO LOOP ON COMPLETE TEST AS SELECTED,
                                /       1       LOOP ON COMPLETE TEST AS SELECTED
                                /               (OCTAL PASS NUMBER TYPED AT END OF EACH PASS),
```

```
                        /TTY KEYBOARD COMMAND CONTROL,
                                /NOTES: 1,  THE FOLLOWING COMMANDS MAY BE ENTERED WHEN "C" IS
                        /TYPED,
                        /  (UPON ENTRY MONITOR TYPES THE CONTENTS OF LOCATION 7, AND THE TEST
                        /THAT IS CURRENTLY BEING RUN, THEN "C")
                                /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
                        /"T01T02T05TNN←"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 "C" IS DISPLAYED AGAIN,
                        /                 IT RESULTS IN A MODIFIED ERROR MESSAGE BEING DISPLAYED
                        /                 FOR EACH SUBTEST RUN IF THAT TEST PASSES,  THESE
                        /                 PRINTOUTS ARE CONTROLLED BY THE SAME SR OPTIONS AS
                        /                 ERROR MESSAGES,
                        /"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
          7600      *7600
                            /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:
    7600    0203    TEST01  /670X,671X IOT EXISTENCE AND BASIC WC,CA,CM,DB,FR REGISTER TEST.
    7601    0600    TEST02  /WC,CA,CM,DB,FR REGISTER DATA TEST.
    7602    1000    TEST03  /BASIC FUNCTIONS EXISTENCE TEST.
    7603    1235    TEST04  /WC,CA REGISTERS INCREMENT TEST (INCLUDES CM6-8).
    7604    1400    TEST05  /MEMORY-DB DATA TEST.
    7605    1600    TEST06  /CA REGISTER MEMORY REFERENCE TEST (USES ALL EXISTING MEMORY FIELDS).
    7606    2000    TEST07  /STATUS REGISTERS STAND ALONE TEST.
    7607    2200    TEST10  /BASIC TESTS FOR MTTF, ILLEGAL FUNCTION, READ COMPARE, AND EF.
    7610    3000    TEST11  /MTTF AND EF SKIP-NO SKIP TEST.
    7611    3200    TEST12  /MTTF-EF INTERRUPT TEST.
    7612    3400    TEST13  /672X IOT UNIQUENESS TEST.
    7613    3600    TEST14  /COMPLETE READ COMPARE TEST.

                            /PROGRAM CONTROL SECTION:
    7614    4200    EXEC    /EXECUTIVE.
    7615    4400    MONIT   /MONITOR.
    7616    5400    INTSEV  /INTERRUPT SERVICE.

                            /ERROR HANDLERS:
    7617    5000    ERRORS  /ERROR HANDLING ROUTINE.
    7620    5600    COMP    /ERROR DETECTOR.
    7621    6677    FE00A   /ERROR TABLE.

                            /UTILITIES:
    7622    4000    S6700   /TM8E IOT SUBROUTINES.
    7623    1624    MARK3   /(IF THE DEVICE CODES ARE
                            /NOT 70-72 IN YOUR SYSTEM
                            /CHANGE ONLY THE IOT CODES IN
                            /EACH OF THE 23 SUBROUTINES AND IN THE LOCATION MARK3.)
    7624    6105    LASS    /LOOP SENSORS.
    7625    6200    AMG8E   /PACKED TEXT MESSAGE GENERATOR.
    7626    6244    TYPE    /TTY ROUTINES.
    7627    6301    EDIT    /OCTAL NUMBER OUTPUT EDITOR.
    7630    6400    CLR1    /"INITIALIZE" GENERATORS.
    7631    6432    OFFLNE  /FUNCTION ROUTINES.

                            /DATA SECTION:
    7632    6613    T1LS1   /TEST01 IOT LISTS.
    7633    6647    T3LS1   /TEST03 FUNCTION LIST.
    7634    3146    T11LS1  /TEST11 IOT LIST
    7635    3337    T12LS1  /TEST12 INTERRUPT ENABLE LIST.
    7636    3146    T11LS1  /TEST13 IOT LIST.
    7637    0000    0000    /CONSTANTS, POINTERS, ETC.
    7640    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  0003    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
```

```
0061  6200    AMG8EP, AMG8E
0062  4400    MONITP, MONIT
0063  6301    EDITP,  EDIT
0064  6326    EDTEMP, EDTEM
0065  6262    BELLP,  BELL
0066  6270    CRLFP,  CRLF
0067  7313    ERMSGP, ERMSG
0070  6244    TYPEP,  TYPE
0071  5507    EXECFP, EXECFX

              /TM8E IOT SUBROUTINE POINTERS,
0072  4000    S6700P, S6700                   /LOAD IOT,
0073  4004    SLWCRP, SLWCR                   /LWCR
0074  4010    SCWCRP, SCWCR                   /CWCR
0075  4014    SLCARP, SLCAR                   /LCAR
0076  4020    SCCARP, SCCAR                   /CCAR
0077  4024    SLCMRP, SLCMR                   /LCMR
0100  4030    SLFGRP, SLFGR                   /LFGR
0101  4034    SLDBRP, SLDBR                   /LDBR
0102  4040    S6710P, S6710                   /READ IOT,
0103  4044    SRWCRP, SRWCR                   /RWCR
0104  4050    SRCARP, SRCAR                   /RCAR
0105  4054    SRMSRP, SRMSR                   /RMSR
0106  4060    SRCMRP, SRCMR                   /RCMR
0107  4064    SRFSRP, SRFSR                   /RFSR
0110  4070    SRDBRP, SRDBR                   /RDBR
0111  4074    S6720P, S6720                   /CONTROL IOT,
0112  4100    SSKEFP, SSKEF                   /SKEF
0113  4105    SSKCBP, SSKCB                   /SKCB
0114  4112    SSKTDP, SSKTD                   /SKTD
0115  4117    SSKTRP, SSKTR                   /SKTR
0116  4124    SCLFP,  SCLF                    /CLF
0117  4130    SCLTP,  SCLT                    /CLT
0120  4134    SSDLEP, SSDLE                   /SDLE
0121  4140    SSBRMP, SSBRM                   /SBRM

              /PSEUDO MNEMONIC SUBROUTINE POINTERS,
0122  6547    SRBSWP, SRBSW                   /BSW
0123  6400    CLR1P,  CLR1                    /CLEAR1
0124  6406    CLR2P,  CLR2                    /CLEAR2
0125  6414    CLR3P,  CLR3                    /CLEAR3
0126  6105    LAS5P,  LAS5                    /LOOP5
0127  6113    LAS6P,  LAS6                    /LOOP6
0130  6121    LAS7P,  LAS7                    /LOOP7
0131  6127    LAS8P,  LAS8                    /LOOP8
0132  6135    LAS9P,  LAS9                    /LOOP9
0133  5600    COMPP,  COMP                    /COMPAR
0134  5000    ERRORP, ERRORS                  /ERROR
0135  6534    DATBRP, DATBRK                  /BREAK
0136  6525    CMDP,   CMD                     /COMAND
0137  6432    OFFLNP, OFFLNE                  /OFFLIN
0140  6436    REWNP,  REWND                   /REWIND
0141  6442    READDP, READD                   /READ
0142  6446    RDCMPP, RDCMP                   /RDCOMP
  '3  6452    WRITP,  WRIT                    /WRITE
```

```
0144  6456    EOFP,   EOFW                    /WEOF
0145  6462    SPFORP, SPFORD                  /SPCFWD
0146  6466    SPREVP, SPREV                   /SPCREV
0147  6472    XGAPP,  XGAP                    /EXTGAP
0150  6500    GOEP,   GOE                     /GO
0151  0565    SIOT1P, SIOT1                   /LXXR,CXXR
0152  0570    SIOT2P, SIOT2                   /RXXR
0153  6541    WCOVEP, WCOVER                  /WCOV

                /WIDE USEAGE TEMPORARIES,
0154  0000    GOOD,   0                       /GOOD DATA',
0155  0000    BAD,    0                       /BAD DATA
0156  0000    OLD,    0                       /OLD DATA,
0157  0000    TXXTM1, 0                       /THREE TEMPORARIES FOR
0160  0000    TXXTM2, 0                       /FORMAL TEST USEAGE ONLY,
0161  0000    TXXTM3, 0
0162  0000    SAVEAC, 0                       /AC SAVED,
0163  0000    SAVEL,  0                       /LINK SAVED,
0164  0000    INTEF,  0                       /EF INTERRUPT SERVICE ADDRESS,
0165  0000    PRGFLD, 0                       /MEMORY FIELD WHICH PROGRAM OCCUPIES,
0166  0000    INTMTF, 0                       /MTTF INTERRUPT SERVICE ADDRESS,
0167  0000    MEMFLD, 0                       /NUMBER OF EXTENDED MEM FIELDS PRESENT,
0170  0000    ACLOC,  0                       /CONTENTS OF AC FOR ILLEGAL SKIP,

                /WIDE USEAGE SOFTWARE FLAGS AND INDICATORS,
0171  0000    TRACE,  0                       /SET IF "TR" PENDING,
0172  0000    INTFLG, 0                       /SET IF INTERRUPT SHOULD BE TURNED ON
                                              /AFTER INTERRUPT SERVICE,
0173  0000    TTOFLG, 0                       /SET IF TTO FLAG GETS SET,
0174  0000    EXITFL, 0                       /SET IF AUTO-EXIT "EX-",
0175  0000    ACTFLG, 0                       /SET IF TEST IN PROGRESS,
0176  0000    TSTAT,  0                       /BIT N SET MEANS RUN TEST
                                              /N+1 (IN OCTAL),
0177  0000    TSTNUM, 0                       /CURRENT TEST NUMBER,


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

      0200      *200

0200  5455    START,  JMP I   EXECP           /START HERE, GO TO EXEC,
0201  7120            STL
0202  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 01, 670X, 671X IOT EXISTENCE AND BASIC NON-STATUS REGISTERS
                    /TEST.

                    /1.  VERIFIES THAT 670X AND 671X IOT'S EXIST EXCEPT FOR
                    /RMSR AND THAT WC, CA, CM, DB AND FR0-4 CAN BE
                    /LOADED AND READ WITH ALL 1'S AND ALL 0'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 T01D-T01P 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 T01A-T01C (AC CLEARING)
                    /3.2  LOOP 7 CYCLES ON T01D-T01N (IOT EXISTENCE & BASIC REGISTER)
                    /3.3  LOOP 8 CYCLES ON T01O-T01P (IOT UNIQUENESS)


0203  0000    TEST01, 0
0204  7001            IAC
0205  3177            DCA     TSTNUM
0206  6002            IOF                     /RUN WITH INTERRUPT SYSTEM OFF.
0207  3172            DCA     INTFLG
0210  3154            DCA     GOOD

                    /VERIFY 6700 CLEARS AC.
0211  7240    T01A,   CLA CMA
0212  4472            I6700
0213  4533            COMPAR
0214  6705            ER01A
                    /IOT 6700 DID NOT CLEAR AC,  GD=GOOD AC; BD=REAL AC.
0215  5211            JMP     T01A            /SUBTEST LOOP.

                    /VERIFY 6710 CLEARS AC.
0216  7240    T01B,   CLA CMA
0217  4502            I6710
0220  4533            COMPAR
0221  6707            ER01B
                    /IOT 6710 DID NOT CLEAR AC,  GD=GOOD AC; BD=REAL AC
0222  5216            JMP     T01B            /SUBTEST LOOP.

                    /VERIFY 6720 DOES NOT CLEAR AC
0223  7240    T01C,   CLA CMA
0224  3154            DCA     GOOD
0225  7240            CLA CMA
0226  4511            I6720
0227  4533            COMPAR
0230  6711            ER01C
                    /IOT 6720 CLEARED AC, GD=GOOD AC, BD=REAL AC.
0231  5225            JMP     T01C+2          /SUBTEST LOOP

0232  4527            LOOP6
0233  5210            JMP     T01A-1          /*****LOOP 6*****
```

```
0234  4775          JMS I   T1EX1P          /TO EXECUTIVE.

                    /VERIFY LOAD IOT LXXR CLEARS AC.
0235  3154    T01D, DCA     GOOD
0236  1157          TAD     TXXTM1
0237  4551          LXXR
0240  4533          COMPAR
0241  6713          ER01D
                    /LOAD IOT LXXR DID NOT CLEAR AC,  GD=GOOD AC; BD=REAL AC;
                    /I1=LXXR.
0242  5236          JMP     T01D+1          /SUBTEST LOOP.


                    /VERIFY THAT LXXR WITH AC=7777, THEN RXXR RESULTS IN A
                    /NON-ZERO AC.
0243  1157    T01E, TAD     TXXTM1
0244  4551          LXXR
0245  7200          CLA                     /CLEAR AC TO ENSURE GOOD TEST.
0246  4552          RXXR
0247  0157          AND     TXXTM1          /MASK BITS 0-4 FOR FS ONLY.
0250  7650          SNA CLA
0251  7040          CMA
0252  4533          COMPAR
0253  6715          ER01E
                    /LXXR WITH AC=7777, THEN RXXR DID NOT RESULT IN A NON-ZERO
                    /AC.  I1=LXXR; I2=RXXR
0254  5243          JMP     T01E            /SUBTEST LOOP

                    /VERIFY XX REGISTER TO BE LOADED AND READ ALL 1'S BY LXXR
                    /THEN RXXR.
0255  7240    T01F, CLA CMA
0256  3154          DCA     GOOD
0257  1157          TAD     TXXTM1
0260  4551          LXXR
0261  7200          CLA                     /AC CLEARED TO ENSURE GOOD TEST.
0262  4552          RXXR
0263  0157          AND     TXXTM1          /MASK TO BITS 0-4 FOR FS ONLY.
0264  1160          TAD     TXXTM2          /ADD IN COMPLEMENT MASK.
0265  4533          COMPAR
0266  6717          ER01F
                    /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.
0267  5257          JMP     T01F+2          /SUBTEST LOOP

                    /VERIFY THAT LXXR WITH AC=0000, THEN RXXR RESULTS IN
                    /AT LEAST ONE BIT CLEAR IN AC.
0270  3154    T01G, DCA     GOOD
0271  4551          LXXR
0272  7200          CLA
0273  4552          RXXR
0274  0157          AND     TXXTM1          /MASK BITS 0-4 FOR FS ONLY.
0275  1160          TAD     TXXTM2          /ADD IN COMPLEMENT MASK.
0276  7001          IAC                     /ADD 1 TO CHECK FLIP TO 0.
  77  7650          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
```

```
0343  7040            CMA
0344  4533            COMPAR
0345  6731            ER01K
                  /CXXR DID NOT CLEAR AT LEAST ONE BIT IN XX REGISTER
                  /I1=CXXR; I2=RXXR
0346  5337            JMP     T01K+1          /SUBTEST LOOP

                  /VERIFY THAT CXXR CLEARS THE XX REGISTER
0347  7200    T01L, CLA
0350  4551            CXXR
0351  4552            RXXR
0352  4533            COMPAR
0353  6733            ER01L
                  /CXXR DID NOT CLEAR XX REGISTER,  GD=GOOD AC; BD=REAL AC;
                  /I1=CXXR; I2=RXXR,
0354  5350            JMP     T01L+1          /SUBTEST LOOP

                  /VERIFY CXXR CLEARS AC.
0355  7240    T01M, CLA CMA
0356  4551            CXXR
0357  4533            COMPAR
0360  6735            ER01M
                  /CXXR DOES NOT CLEAR AC,  GD=GOOD AC; BD=REAL AC;
                  /I1=CXXR,
0361  5355            JMP     T01M            /SUBTEST LOOP

                  /VERIFY CXXR WITH AC=7777 DOES NOT SET ANY BIT IN XX
                  /REGISTER.
0362  7240    T01N, CLA CMA
0363  4551            CXXR
0364  4552            RXXR
0365  4533            COMPAR
0366  6737            ER01N
                  /CXXR SET AT LEAST ONE BIT IN XX REGISTER,  GD=GOOD AC;
                  /BD=REAL AC; I1=CXXR; I2=RXXR,
0367  5362            JMP     T01N            /SUBTEST LOOP

0370  4777            JMS I   T1EX3P          /BACK TO EXECUTIVE.


0371  4530    T1LP7,  LOOP7
0372  5232            JMP     T01D-3          /*****LOOP 7*****

0373  5774            JMP I   .+1             /GO TO T010 WHEN D=N DONE
0374  0400            T010L
0375  0450    T1EX1P, T1EX1
0376  0473    T1EX2P, T1EX2
0377  0501    T1EX3P, T1EX3

      0400            PAGE


0400  4305    T010L,  JMS     T1EX4           /TO EXECUTIVE TO INSERT
0401  4525            CLEAR3                  /IOT'S,
0402  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,
0403  7240    T010,     CLA CMA
0404  4551              LXXR
0405  4552              RXXR
0406  0157              AND       TXXTM1
0407  4533              COMPAR
0410  6741              ER010
                        /IOT LXXR AFFECTS REGISTER READ BY RXXR, OR RXXR READS
                        /REGISTER LOADED BY LXXR.  GD=GOOD AC; BD=REAL AC;
                        /I1=LXXR; I2=RXXR,
0411  5203              JMP       T010            /SUBTEST LOOP

                        /VERIFY THAT SPECIFIC LOAD, CLEAR AND READ IOT'S AFFECT
                        /ONLY THEIR RELATED REGISTERS.  AC SET TO 0000 PRIOR TO
                        /LOAD OR CLEAR WITH ALL OTHER REGISTERS SET TO 1'S,
                        /SAME METHOD AS T010,
0412  4525              CLEAR3
0413  7240              CLA CMA                   /LOAD ALL REGISTERS WITH 1'S,
0414  3154              DCA       GOOD
0415  7240              CLA CMA
0416  4473              LWCR
0417  7240              CLA CMA
0420  4475              LCAR
0421  7240              CLA CMA
0422  4477              LCMR
0423  7240              CLA CMA
0424  4501              LDBR
0425  1374              TAD       K7600
0426  4500              LFGR
0427  7200    T01P,     CLA
0430  4551              LXXR                      /LOAD ONE REGISTER
0431  4552              RXXR                      /READ ANOTHER
0432  0157              AND       TXXTM1          /MASK BITS 0-4 FOR FS ONLY,
0433  1160              TAD       TXXTM2          /ADD COMPLEMENT MASK,
0434  4533              COMPAR
0435  6743              ER01P
                        /IOT LXXR AFFECTS REGISTER READ BY RXXR, OR RXXR READS
                        /REGISTER LOADED BY LXXR.  GD=GOOD AC; BD=REAL AC;
                        /I1=LXXR; I2=RXXR,
0436  5230              JMP       T01P+1          /SUBTEST LOOP

0437  4340              JMS       T1EX5           /TO EXECUTIVE

0440  4531              LOOP8
0441  5773              JMP I     T1LP7P          /*****LOOP 8*****

0442  4523              CLEAR1                    /CLEAR REGISTERS + ION,
0443  1647              TAD I     TST1P           /EXIT,
0444  3246              DCA       .+2
0445  5646              JMP I     .+1
0446  0000              0
0447  0203    TST1P,    TEST01
```

```
                    /TEST 01 LOCAL EXECUTIVE TO SERVICE TESTS T01D-T01N,

0450  0000  T1EX1,  0                       /ENTER PRIOR TO T01D,
0451  1356          TAD     T1LS1P          /SET UP LIST POINTERS
0452  3010          DCA     10
0453  1357          TAD     T1LS2P
0454  3011          DCA     11
0455  1360          TAD     T1LS3P
0456  3012          DCA     12
0457  1041          TAD     M5
0460  3361          DCA     T1EXT1
0461  1410  T1EX1A, TAD I    10              /SET IOT1 FOR LXXR,
0462  3366          DCA     IOT1
0463  1412          TAD I   12              /SET IOT2 FOR RXXR,
0464  3371          DCA     IOT2
0465  1412          TAD I   12              /SET MASK AND COMPLEMENT
0466  3157          DCA     TXXTM1          /MASK,
0467  1157          TAD     TXXTM1
0470  7040          CMA
0471  3160          DCA     TXXTM2
0472  5650          JMP I   T1EX1           /GO BACK AND RUN T01D-T01J,
0473  0000  T1EX2,  0                       /ENTER AFTER T01J,
0474  1411          TAD I   11              /SEE IF CXXR APPLIES,
0475  7450          SNA
0476  5302          JMP     .+4
0477  3366          DCA     IOT1            /CXXR APPLIES,  PUT IN IOT1,
0500  5673          JMP I   T1EX2           /GO BACK AND RUN T01K-T01N,
0501  0000  T1EX3,  0                       /ENTER AFTER T01N,
0502  2361          ISZ     T1EXT1          /ALL IOT'S CHECKED?
0503  5261          JMP     T1EX1A          /NO,  RESET IOT'S
0504  5701          JMP I   T1EX3           /YES,  EXIT TO END OF T01N,

                    /TEST 01 LOCAL EXECUTIVE TO SERVICE TESTS T01O-T01P,

0505  0000  T1EX4,  0
0506  1356          TAD     T1LS1P          /SET LIST POINTERS,
0507  3010          DCA     10
0510  1357          TAD     T1LS2P
0511  3011          DCA     11
0512  1360          TAD     T1LS3P
0513  3361          DCA     T1EXT1
0514  1041          TAD     M5              /SET FOR 5 LOAD, CLEAR COMBINATIONS,
0515  3362          DCA     T1EXT2

0516  1410  T1EX7,  TAD I   10              /GET LOAD,
0517  3366          DCA     IOT1            /PUT IN IOT1,
0520  1037          TAD     M2              /SET FOR PAIR COUNTER,
0521  3364          DCA     T1EXT4
0522  2361          ISZ     T1EXT1          /UPDATE READ POINTER,
0523  2361          ISZ     T1EXT1
0524  1361  T1EX6,  TAD     T1EXT1          /GET READ POINTER AND
0525  3012          DCA     12              /STORE IN AUTO-INDEX,
  26  1040          TAD     M4              /SET FOR 4 READS,
```

```
0527  3363            DCA     T1EXT3
0530  1412  T1EX8,     TAD I   12              /GET READ,
0531  3371            DCA     IOT2            /PUT IN IOT2,
0532  1412            TAD I   12              /GET MASK,  SAVE MASK
0533  3157            DCA     TXXTM1          /AND COMPLEMENT MASK,
0534  1157            TAD     TXXTM1
0535  7040            CMA
0536  3160            DCA     TXXTM2
0537  5705            JMP I   T1EX4           /RUN TEST,
0540  0000  T1EX5,     0
0541  2363            ISZ     T1EXT3          /ENTER AFTER TEST,  4 READS?
0542  5330            JMP     T1EX8           /NO,  DO NEXT READ,
0543  2364            ISZ     T1EXT4          /YES,  DONE WITH THIS LOAD, CLEAR PAIR?
0544  7410            SKP
0545  5353            JMP     .+6             /YES,
0546  1411            TAD I   11              /NO,  GET CLEAR,
0547  7450            SNA
0550  5353            JMP     .+3
0551  3366            DCA     IOT1            /CLEAR APPLIES,  RUN TEST AFTER
0552  5324            JMP     T1EX6           /SETTING UP READS,
0553  2362            ISZ     T1EXT2          /ALL DONE?
0554  5316            JMP     T1EX7           /NO,  RUN NEXT PAIR,
0555  5740            JMP I   T1EX5           /YES,  EXIT,
0556  6612  T1LS1P,    T1LS1-1
0557  6617  T1LS2P,    T1LS2-1
0560  6624  T1LS3P,    T1LS3-1
0561  0000  T1EXT1,    0
0562  0000  T1EXT2,    0
0563  0000  T1EXT3,    0
0564  0000  T1EXT4,    0
0565  0000  SIOT1,     0                       /IOT1 SUBROUTINE,
0566  0000  IOT1,      0
0567  5765            JMP I   SIOT1
0570  0000  SIOT2,     0                       /IOT2 SUBROUTINE,
0571  0000  IOT2,      0
0572  5770            JMP I   SIOT2
0573  0371  T1LP7P,    T1LP7
0574  7600  K7600,     7600
```

```
                        /TEST 02,  NON-STATUS REGISTER DATA TEST,

                                /1,  VERIFIES THAT WC, CA, CM, DB AND FR0-4 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:  T02A(WC)
                                /3.2  LOOP 7:  T02B(CA)
                                /3.3  LOOP 8:  T02C(CM)
                                /3.4  LOOP 9:  T02D-T02E(DB,FR0-4)



      0600                      PAGE
0600  0000  TEST02, 0
0601  4523                      CLEAR1                    /CLEAR ALL-ION,

                        /VERIFY WC CAN BE LOADED AND READ USING INCREMENTAL
                        /COMPLEMENTED PATTERN,
0602  4307  T02A,      JMS       T02LDS                  /GENERATE DATA,
0603  7777              7777                              /MASK TO CHECK ALL BITS,
0604  5215              JMP       T02B-2                  /DONE,
0605  1157              TAD       TXXTM1                  /GET DATA
0606  4473              LWCR                              /LOAD WC,
0607  7200              CLA                               /CLEAR AC TO ENSURE GOOD TEST,
0610  4503              RWCR                              /READ WC,
0611  4533              COMPAR
0612  6745              ER02A
                        /WC NOT LOADED OR READ CORRECTLY, GD=GOOD WC; OD=PREVIOUS
                        /GOOD WC, SUBTEST LOOP CYCLES ON PRESENT DATA,
0613  5205              JMP       .-6                     /SUBTEST LOOP
0614  5331              JMP       T02LDR                  /GET MORE DATA,

0615  4527              LOOP6
0616  5202              JMP       T02A                    /*****LOOP 6*****

                        /VERIFY CA CAN BE LOADED AND READ USING INCREMENTAL
                        /COMPLEMENTED DATA,
0617  4307  T02B,      JMS       T02LDS
0620  7777              7777
0621  5232              JMP       T02C-2
0622  1157              TAD       TXXTM1
0623  4475              LCAR
0624  7200              CLA                               /CLEAR AC TO ENSURE GOOD TEST,
0625  4504              RCAR
0626  4533              COMPAR
0627  6747              ER02B
                        /CA NOT LOADED OR READ CORRECTLY,  GD=GOOD CA; OD=PREVIOUS
                        /GOOD CA,  SUBTEST LOOP CYCLES ON PRESENT DATA,
0630  5222              JMP       .-6                     /SUBTEST LOOP,
0631  5331              JMP       T02LDR
```

```
0632  4530            LOOP7
0633  5215            JMP     T02B-2          /*****LOOP 7*****

                      /VERIFY CM CAN BE LOADED AND READ USING INCREMENTAL
                      /COMPLEMENTED DATA.
0634  3172  T02C,     DCA     INTFLG          /ALLOW NO INTERRUPTS,
0635  6002            IOF
0636  4307            JMS     T02LDS
0637  7777            7777
0640  5251            JMP     T02D-3
0641  1157            TAD     TXXTM1
0642  4477            LCMR
0643  7200            CLA                     /CLEAR AC TO INSURE GOOD TEST,
0644  4506            RCMR
0645  4533            COMPAR
0646  6751            ER02C
                      /CM NOT LOADED OR READ CORRECTLY, GD=GOOD CM; OD=PREVIOUS
                      /GOOD CM, SUBTEST LOOP CYCLES ON PRESENT DATA,
0647  5241            JMP     .-6             /SUBTEST LOOP
0650  5331            JMP     T02LDR

0651  4523            CLEAR1                  /CLEAR ALL - ION,
0652  4531            LOOP8
0653  5232            JMP     T02C-2          /*****LOOP 8*****

                      /VERIFY DB CAN BE LOADED AND READ USING INCREMENTAL
                      /COMPLEMENTED DATA.
0654  4524  T02D,     CLEAR2
0655  4307            JMS     T02LDS
0656  7777            7777
0657  5270            JMP     T02E
0660  1157            TAD     TXXTM1
0661  4501            LDBR
0662  7200            CLA                     /CLEAR AC TO ENSURE GOOD TEST,
0663  4510            RDBR
0664  4533            COMPAR
0665  6753            ER02D
                      /DB NOT LOADED OR READ CORRECTLY, GD=GOOD DB; OD=PREVIOUS
                      /GOOD DB, SUBTEST LOOP CYCLES ON PRESENT DATA,
0666  5260            JMP     .-6             /SUBTEST LOOP
0667  5331            JMP     T02LDR

                      /VERIFY FS 0-4 CAN BE LOADED AND READ USING INCREMENTAL
                      /COMPLEMENTED DATA.
0670  4307  T02E,     JMS     T02LDS
0671  7600            7600                    /MASK FOR BITS 0-4
0672  5304            JMP     T02LP9
0673  1157            TAD     TXXTM1
0674  4500            LFGR
0675  7200            CLA
0676  4507            RFSR
0677  0271            AND     T02E+1          /MASK
0700  4533            COMPAR
0701  6755            ER02E
                      /FS 0-4 NOT LOADED OR READ CORRECTLY, GD=GOOD FS 0-4;
```

```
                    /BD=REAL FS 0-4 (MASKED); OD=PREVIOUS GOOD FS 0-4,
0702  5273          T02D,   JMP     .-7             /SUBTEST LOOP
0703  5331                  JMP     T02LDR

0704  4532          T02LP9, LOOP9
0705  5252                  JMP     T02D-2          /*****LOOP 9*****

0706  5600                  JMP I   TEST02          /EXIT,


                    /DATA SUPPLIER FOR WC, CA, CM, DB, FR,
                    /STARTING FROM 0 OUTPUTS THE FOLLOWING INCREMENTAL COMPLEMENTED
                    /DATA PATTERN,
                            /0:  7777,7777,0,0,7777,7777,0,0
                            /1:  7776,7776,1,1,7776,7776,1,1, ETC,
                    /THE CALLING SEQUENCE IS:
                    /JMS    T02LDS
                    /DATA MASK CONSTANT
                    /EXIT
                    /TEST INSTRUCTIONS
                    /ERROR CHECK
                    /JMP T02LDR TO GET NEW DATA,

0707  0000          T02LDS, 0                       /ENTER,
0710  3157                  DCA     TXXTM1          /0 FOR A STARTER,
0711  1021                  TAD     K2              /COMPUTE RETURN EXIT
0712  1307                  TAD     T02LDS          /AND SAVE,
0713  3160                  DCA     TXXTM2
0714  1040          T02LDU, TAD     M4              /SET FOR FOUR COMPLEMENTS,
0715  3161                  DCA     TXXTM3
0716  1037          T02LDT, TAD     M2              /SET FOR TWO SENDS OF ONE PATTERN,
0717  3345                  DCA     T02LDM
0720  1154                  TAD     GOOD            /PUT PREVIOUS DATA
0721  3156                  DCA     OLD             /IN OLD,
0722  1157                  TAD     TXXTM1          /SET LAST DATA,
0723  7040                  CMA                     /COMPLEMENT,
0724  3157                  DCA     TXXTM1          /SAVE,
0725  1157                  TAD     TXXTM1          /GET LAST DATA,
0726  0707                  AND I   T02LDS          /MASK,
0727  3154                  DCA     GOOD            /THIS IS WHAT WE SHOULD READ,
0730  5560                  JMP I   TXXTM2          /RUN THE TEST,
0731  2345          T02LDR, ISZ     T02LDM          /THIS PATTERN SENT 2 TIMES,
0732  7410                  SKP
0733  5337                  JMP     .+4
0734  1154                  TAD     GOOD            /YES,  PUT OLD DATA IN OLD,
0735  3156                  DCA     OLD
0736  5560                  JMP I   TXXTM2          /RUN PATTERN AGAIN,
0737  2161                  ISZ     TXXTM3          /COMPLEMENTING DONE 4 TIMES?
0740  5316                  JMP     T02LDT          /NO,  SET UP NEXT DATA,
0741  2157                  ISZ     TXXTM1          /YES,  INCREMENT DATA,
0742  5314                  JMP     T02LDU          /SEND OUT NEXT DATA TRAIN,
0743  2307                  ISZ     T02LDS          /ALL DONE,  UPDATE RETURN,
0744  5707                  JMP I   T02LDS          /EXIT,
0745  0000          T02LDM, 0
```

```
                    /TEST 03,  BASIC FUNCTIONS EXISTENCE TEST,

                            /1,  REFER TO SUBTESTS FOR SPECIFIC TEST INFORMATION,
                            /2,  TESTS T03D-T03I ARE GENERALIZED AND ARE CONTROLLED BY A
                    /LOCAL EXECUTIVE,
                            /3,  LOOPS 6-9:
                            /3,1  LOOP 6 CYCLES ON T03A-T03C,
                            /3,2  LOOP 7 CYCLES ON T03D-T03E,
                            /3,3  LOOP 8 CYCLES ON T03F-T03G,
                            /3,4  LOOP 9 CYCLES ON T03H-T03I,
                            /4,  EACH SUBTEST LOOP CYCLES ON THE FUNCTION IN CURRENT USE,


          1000              PAGE
1000      0000    TEST03,   0
1001      4523              CLEAR1                      /CLEAR ALL REGISTERS-ION
1002      1047              TAD       XBUFFP
1003      3156              DCA       OLD

                    /VERIFY WRITE-GO DOES NOT CAUSE A DATA BREAK WITH NO
                    /TRANSOPRT READY, WC STARTS AT 0000 AND SHOULD
                    /REMAIN 0000,
1004      3154    T03A,     DCA       GOOD
1005      4524              CLEAR2                      /CLEAR REGISTERS
1006      4473              LWCR                        /0 TO WC
1007      4543              WRITE
1010      4550              GO
1011      4503              RWCR
1012      4533              COMPAR
1013      6757              ER03A
                    /WC CHANGED WHEN WRITE-GO ISSUED WITH NO TRANSPORT READY,
                    /GD=GOOD WC,
1014      5205              JMP       T03A+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,)
1015      7240    T03B,     CLA CMA
1016      3154              DCA       GOOD
1017      4404              JMS I     T3LODP            /LOAD UP REGISTERS,
1020      4501              LDBR                        /0 TO DB
1021      4543              WRITE
1022      4535              BREAK                       /1 DATA BREAK
1023      4510              RDBR
1024      4533              COMPAR
1025      6761              ER03B
                    /GD=GOOD DB; OD=ADDRESS XBUFF (CA SHOULD EQUAL THIS
                    /QUANTITY OR ONE MORE-CA INCREMENT IS NOT BEING
                    /TESTED,)
1026      5217              JMP       T03B+2            /SUBTEST LOOP,

                    /VERIFY READ CAUSES TRANSFER INTO MEMORY, CA REFERENCES
                    /ADDRESSES XBUFF OR XBUFF+1, AND THAT DB CAN SEND ALL
                    /0'S. (XBUFF AND XBUFF+1 CONTAIN 7777 PRIOR TO
```

```
                    /BREAK.)
1027  4404  T03C,   JMS I   T3LODP          /LOAD UP REGISTERS
1030  4501          LDBR                    /0 TO DB
1031  4541          READ
1032  4535          BREAK                   /1 DATA BREAK
1033  1050          TAD     XBUFF           /SUM
1034  1051          TAD     XBUFF+1
1035  4533          COMPAR
1036  6763          ER03C
                    /GD=GOOD SUM OF THE CONTENTS OF XBUFF AND XBUFF+1 (ONE OF
                    /THE VALUES SHOULD BE 7777, THE OTHER 0000 YIELDING A
                    /SUM OF 7777); BD=ACTUAL SUM; OD=ADDRESS XBUFF
                    /(CA SHOULD EQUAL THIS QUANTITY OR 1 MORE = CA INCREMENT
                    /NOT BEING TESTED.)
1037  5227          JMP     T03C            /SUBTEST LOOP

1040  4527          LOOP6
1041  5204          JMP     T03A            /*****LOOP6*****

                    /VERIFY ALL FUNCTIONS EXCEPT READ DO NOT CAUSE TRANSFER
                    /INTO PROCESSOR MEMORY. XBUFF AND XBUFF+1 CONTAIN
                    /7777 PRIOR TO BREAK. DB CONTAINS 0000.
1042  3154  T03D,   DCA     GOOD
1043  4776          JMS I   T3EX1P          /TO EXECUTIVE TO INSERT FUNCTION.
1044  6647          T3LS1                   /FUNCTION LIST POINTER
1045  7771          -7                      /NUMBER OF FUNCTIONS TO USE.
1046  1051          .+3                     /WHERE TO PUT FUNCTION.
1047  4404          JMS I   T3LODP          /LOAD ALL REGISTERS.
1050  4501          LDBR                    /0 TO DB.
1051  0000          0                       /VARIABLE FUNCTION HERE.
1052  4535          BREAK                   /1 DATA BREAK.
1053  1050          TAD     XBUFF           /XBUFF AND XBUFF+1 SHOULD
1054  7041          CIA                     /BE EQUAL AND UNCHANGED.
1055  1051          TAD     XBUFF+1
1056  4533          COMPAR
1057  6765          ER03D
                    /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.
1060  5777          JMP I   T3EX32          /SUBTEST LOOP.
1061  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 0000.
1062  4776  T03E,   JMS I   T3EX1P          /TO EXECUTIVE TO GET FUNCTION.
1063  6651          T3LS1+2                 /FUNCTION LIST POINTER.
1064  7772          -6                      /# OF FUNCTIONS
1065  1070          .+3                     /WHERE TO INSERT FUNCTION
1066  4404          JMS I   T3LODP          /LOAD REGISTERS


1067  4501          LDBR                    /0 TO DB
1070  0000          0                       /VARIABLE FUNCTION
      4535          BREAK                   /1 DATA BREAK
```

```
1072  4510          RDBR                    /DB SHOULD STILL BE 0000,
1073  4533          COMPAR
1074  6767          ER03E
                /GD=GOOD DB; OD=ADDRESS XBUFF; FS=FUNCTION WHICH FAILED
1075  5777          JMP I   T3EX32          /SUBTEST LOOP
1076  4775          JMS I   T3EX2P          /GET ANOTHER FUNCTION

1077  4530          LOOP7
1100  5240          JMP     T03D-2          /*****LOOP7*****

                /VERIFY SPCFWD, SPCREV, READ, RDCOMP AND WRITE CAUSE
                /WC INCREMENT FROM 0000 TO 0001,
1101  7201    T03F, CLA IAC
1102  3154          DCA     GOOD
1103  4776          JMS I   T3EX1P          /SET VARIABLE FUNCTION,
1104  6654          T3LS1+5
1105  7773          -5
1106  1111          ,+3
1107  4404          JMS I   T3LODP          /LOAD REGISTERS
1110  4473          LWCR                    /0 TO WC
1111  0000          0                       /FUNCTION INSERTED
1112  4535          BREAK                   / 1 DATA BREAK
1113  4503          RWCR
1114  4533          COMPAR
1115  6771          ER03F
                /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
1120  7201    T03G, CLA IAC
1121  1047          TAD     XBUFFP
1122  3154          DCA     GOOD
1123  4776          JMS I   T3EX1P          /GET A FUNCTION,
1124  6646          T3LS1-1
1125  7775          -3
1126  1130          ,+2
1127  4404          JMS I   T3LODP          /LOAD REGISTERS
1130  0000          0                       /FUNCTION INSERTED HERE,
1131  4535          BREAK                   /1 DATA BREAK,
1132  4504          RCAR
1133  4533          COMPAR
1134  6773          ER03G
                /GD=GOOD CA; FS=FAILING FUNCTION,
1135  5777          JMP I   T3EX32          /SUBTEST LOOP
1136  4775          JMS I   T3EX2P          /GET ANOTHER FUNCTION

1137  4531          LOOP8
1140  5277          JMP     T03F-2          /*****LOOP 8*****

                /VERIFY OFFLINE, REWIND, WEOF DO NOT CAUSE WC INCREMENT,
                /WC PRESET TO 0000,
1141  3154    T03H, DCA     GOOD
1142  4776          JMS I   T3EX1P          /GET FUNCTION
```

```
1143  6651            T3LS1+2
1144  7775            -3
1145  1147            .+2
1146  4473            LWCR                    /0 TO WC
1147  0000            0                       /FUNCTION INSERTED
1150  4535            BREAK                   /1 DATA BREAK
1151  4503            RWCR
1152  4533            COMPAR
1153  6775            ER03H
                /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 0000,
1156  4776    T03I,   JMS I   T3EX1P          /GET A FUNCTION,
1157  6651            T3LS1+2
1160  7773            -5
1161  1163            .+2
1162  4475            LCAR                    /0 TO CA
1163  0000            0                       /FUNCTION
1164  4535            BREAK                   /1 DATA BREAK
1165  4504            RCAR
1166  4533            COMPAR
1167  6777            ER03I
                /GD=GOOD CA; FS=FAILING FUNCTION
1170  5777            JMP I   T3EX32          /SUBTEST LOOP
1171  4775            JMS I   T3EX2P          /GET ANOTHER FUNCTION

1172  4532            LOOP9
1173  5337            JMP     T03H-2          /*****LOOP 9*****

1174  5600            JMP I   TEST03          /EXIT,

1175  1231    T3EX2P, T3EX2
1176  1213    T3EX1P, T3EX1
1177  1227    T3EX32, T3EX3+2
      1200            PAGE

1200  0000    T3LOAD, 0                       /LOAD XBUFF AND XBUFF+1
1201  4524            CLEAR2
1202  7240            CLA CMA                 /WITH 7777, AND LCAR WITH
1203  3050            DCA     XBUFF           /XBUFF ADDRESS,
1204  7240            CLA CMA
1205  3051            DCA     XBUFF+1
1206  1047            TAD     XBUFFP
1207  4475            LCAR
1210  1165            TAD     PRGFLD
1211  4477            LCMR
1212  5600            JMP I   T3LOAD

1213  0000    T3EX1,  0                       /TEST 03 FUNCTION SELECTOR EXECUTIVE,
1214  1613            TAD I   T3EX1           /GET CALL +1 = THIS IS STARTING
1215  3010            DCA     10              /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   10              /WHICH FUNCTION TO BE INSERTED AND
1226  3560            DCA I   TXXTM2          /POINT OF EXIT,
1227  4524            CLEAR2
1230  5613            JMP I   T3EX1           /RUN TEST WITH THIS FUNCTION,
1231  0000    T3EX2,  0                       /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 04,  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 T04A,  (WC INCREMENT)
                                /2,2  LOOP 7 CYCLES ON T04B, CD, AND E (CA AND EMA TESTS),
                                /3,  THE SUBTEST LOOPS CYCLE ON THE PRESENT OLD VALUE
                                /BEING INCREMENTED TO THE GOOD VALUE,

1235  0000      TEST04, 0
1236  4523              CLEAR1


                        /VERIFY COMPLETE WC INCREMENT USING SPCFWD FUNCTION,
                        /START WITH WC=0 AND INCREMENT A FULL COUNT FOUR
                        /TIMES DISREGARDING WC OVERFLOW,
1237  4545      T04A,   SPCFWD                  /SPACE FORWARD,
1240  1040              TAD     M4
1241  3157              DCA     TXXTM1
1242  3154              DCA     GOOD
1243  1154      T04AL,  TAD     GOOD            /GOOD TO OLD,
1244  3156              DCA     OLD
1245  2154              ISZ     GOOD            /+1 TO GOOD,
1246  5252              JMP     T04ASL
1247  2157              ISZ     TXXTM1
1250  7410              SKP
1251  5262              JMP     T4LP6           /YES,
1252  1156      T04ASL, TAD     OLD             /LOAD WC WITH OLD,
1253  4473              LWCR
1254  4535              BREAK                   /1 BREAK,
1255  4503              RWCR                    /CHECK
1256  4533              COMPAR
1257  7001              ER04A
                        /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,
1260  5252              JMP     T04ASL          /SUBTEST LOOP,
1261  5243              JMP     T04AL

1262  4527      T4LP6,  LOOP6
1263  5237              JMP     T04A            /*****LOOP 6*****

                        /VERIFY THAT"GO"BIT CAN BE SET WITH MITF CLEAR,
1264  1027      T04B,   TAD     K100
1265  3154              DCA     GOOD
1266  4524              CLEAR2                  /CLEAR ALL REGISTERS INCLUDING
1267  4543              WRITE                   /MTTF,
1270  4550              GO                      /GO,
1271  4507              RFSR                    /GO SHOULD BE SET,
1272  0027              AND     K100
1273  4533              COMPAR
1274  7003              ER04B
                        /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, T04D 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            ER04D
                /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     T04CSL          /SUBTEST LOOP,
1356  5302            JMP     T04CL

                /VERIFY THAT EMA DOES NOT INCREMENT WHEN EMA INC
                /DISABLED, CA SET TO 7777, EMA TO 0, THEN CAUSE CA
                /INCREMENT AND EMA SHOULD REMAIN 0,
1357  4524    T04E,   CLEAR2                  /CLEAR ALL;
1360  3154            DCA     GOOD
1361  7240            CLA CMA                 /7777 TO CA,
1362  4475            LCAR
1363  4477            LCMR                    /0 TO EMA
1364  4456            JMS I   FUNCP           /WRITE-GO, EMA INC
1365  4100            4100                    /DISABLED,
1366  4535            BREAK                   /1 BREAK,
1367  4506            RCMR                    /CHECK EMA=0,
1370  4533            COMPAR
1371  7011            ER04E
                /EMA CHANGED WHEN EMA INC DISABLED, GD=GOOD EMA;
                /CM=REAL EMA IN BITS 6-8,
1372  5357            JMP     T04E            /SUBTEST LOOP,

1373  4530    T4LP7,  LOOP7
1374  5262            JMP     T4LP6           /*****LOOP 7*****
1375  5635            JMP I   TEST04          /EXIT
1376  0000    T04T1,  0
1377  0000    T04T2,  0
```

```
                    /TEST 05,  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 T05A (MEM TO DB-WRITE)
                            /3.2  LOOP 7 CYCLES ON T05B (DB TO MEM-READ)

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


          1400              PAGE
    1400  0000    TEST05, 0
    1401  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,
    1402  4543    T05A,   WRITE                   /SET WRITE FUNCTION,
    1403  4650            JMS I   T05LDS          /GET DATA,
    1404  7777            7777
    1405  5222            JMP     T5LP6           /DONE EXIT,
    1406  1047    T05ASL, TAD     XBUFFP          /LOAD CA-XBUFF,
    1407  4475            LCAR
    1410  1157            TAD     TXXTM1          /LOAD XBUFF+1 WITH DATA WORD,
    1411  3051            DCA     XBUFF+1
    1412  1165            TAD     PRGFLD          /LOAD CM WITH PROPER MEMORY
    1413  4477            LCMR                    /FIELD,
    1414  4535            BREAK                   /1 DATA BREAK,
    1415  4510            RDBR                    /READ AND CHECK DB,
    1416  4533            COMPAR
    1417  7013            ER05A
                    /GD GOOD DB; OD-PREVIOUS GOOD DB; DB-REAL DB,
    1420  5206            JMP     T05ASL          /SUBTEST LOOP,
    1421  5651            JMP I   T05LDR

    1422  4527    T5LP6,  LOOP6
    1423  5202            JMP     T05A            /*****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  4650    T05B,   JMS I   T05LDS          /GET DATA,
    1425  7777            7777
    1426  5245            JMP     T5LP7           /DONE EXIT,
    1427  4524    T05BSL, CLEAR2                  /CLEAR REGISTERS,
    1430  1047            TAD     XBUFFP          /LOAD CA WITH XBUFF,
    1431  4475            LCAR
```

```
1432  1157            TAD     TXXTM1          /DATA TO DB,
1433  4501            LDBR
1434  1165            TAD     PRGFLD          /FIELD TO CM,
1435  4477            LCMR
1436  4541            READ                    /SET READ FUNCTION
1437  4535            BREAK                   /1 DATA BREAK,
1440  1051            TAD     XBUFF+1         /COMPARE MEMORY


1441  4533            COMPAR
1442  7015            ER05B
                /GD= GOOD CONTENTS OF XBUFF+1; BD:REAL CONTENTS OF XBUFF+1;
                /OD= PREVIOUS GOOD CONTENTS OF XBUFF+1,
1443  5227            JMP     T05BSL          /SUBTEST LOOP,
1444  5651            JMP I   T05LDR

1445  4530    T5LP7,  LOOP7
1446  5222            JMP     T5LP6           /*****LOOP 7*****
1447  5600            JMP I   TEST05

1450  0707    T05LDS, T02LDS
1451  0731    T05LDR, T02LDR
```

```
                    /TEST 06,   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 ARE USED FOR DATA,
                            /3,   THE SUBTEST LOOP CYCLES ON THE SAME MEMORY LOCATION
                    /USING THE SAME DATA,


      1600                  PAGE
1600  0000    TEST06, 0
1601  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.
1602  4242    T06A,   JMS     T6EX1
1603  4517            CLT
1604  1156            TAD     OLD             /SET UP CA AND CM,
1605  4475            LCAR
1606  1160            TAD     TXXTM2
1607  4477            LCMR
1610  4456            JMS I   FUNCP           /WRITE-GO-EMA INCREMENT ENABLED
1611  4140            4140
1612  6002            IOF                     /INTERRUPT SYSTEM TURNED
1613  1161            TAD     TXXTM3          /OFF,
1614  3154            DCA     GOOD
1615  6201    T06ADF, CDF     /N              /CHANGE TO CURRENT DATA FIELD,
1616  1623            TAD I   T06AMP          /GET ORIGINAL MEM AND SAVE,
1617  3361            DCA     T6SAVE
1620  1161            TAD     TXXTM3          /GET DATA AND PUT IN MEM,
1621  3623            DCA I   T06AMP
1622  7410    MARK1,  SKP
1623  0000    T06AMP, 0
1624  6727    MARK3,  6727                    /DATA BREAK,
1625  7000            NOP
1626  1361            TAD     T6SAVE          /RESTORE MEMORY,
1627  3623    MARK2,  DCA I   T06AMP
1630  4457            JMS I   SCDFP           /CHANGE TO THIS PROGRAMS DATA FIELD,
1631  6001            ION                     /INTERRUPT SYSTEM ON,
1632  4510            RDBR                    /DATA SHOULD BE IN DB,
1633  4533            COMPAR
1634  7017            ER06A
                    /GD=GOOD DB; OD=CURRENT MEMORY LOCATION BEING USED MINUS ONE; DB=REAL DB
1635  5203            JMP     T06A+1          /SUBTEST LOOP,
1636  4264            JMS     T6EX2

1637  4527            LOOP6
1640  5201            JMP     T06A-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
  ?2  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     T06ADF          /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   T06AMP          /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     T06AMP          /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,   0
1762  1622  MARK1P,   MARK1
1763  1631  MARK2P,   MARK2+2               /BIASED BY +2 TO YIELD CORRECT TALLY
                                              /NUMBER FOR (MARK2-MARK1)+2,
1764  0000  T6CNT1,   0                     /MARK1 COUNTER,
1765  0000  T6CNT2,   0                     /(MARK2-MARK1)+2 COUNTER,
1766  0000  T6CNT3,   0                     /COMPLEMENT DATA COUNTER,
1767  0000  T6FLAG,   0                     /SET TO 7777 IF TIME FOR COMPLEMENT DATA,
```

```
                    /TEST 07:  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 (T07A-T07B)
                            /2,2  LOOP 7 CYCLES ON S1 CHECK (T07C-T07D)

        2000                PAGE
2000    0000        TEST07, 0
2001    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,
2002    1032        T07A,   TAD     K400            /SELECT REMOTE BIT TO GOOD,
2003    3154                DCA     GOOD
2004    4505                RMSR                    /READ MS AND MASK OUT
2005    0251                AND     K3774           /EF, R/C AND IF BITS,
2006    4533                COMPAR
2007    7021                ER07A
                    /GD=GOOD MS MASKED; BD=REAL MS MASKED,
2010    5204                JMP     T07A+2          /SUBTEST LOOP

                    /VERIFY THAT RMSR CLEARS AC BEFORE READING,  SELECT REMOTE
                    /SHOULD BE ONLY BIT SET,
2011    7240        T07B,   CLA CMA                 /AC=7777,
2012    4505                RMSR                    /READ MS AND MASK OUT
2013    0251                AND     K3774           /EF,R/C AND IF BITS,
2014    4533                COMPAR
2015    7023                ER07B
                    /GD=GOOD MS MASKED; BD=REAL MS MASKED,
2016    5211                JMP     T07B            /SUBTEST LOOP,

2017    4527                LOOP6
2020    5202                JMP     T07A            /*****LOOP6******

                    /FS SHOULD INDICATE BIT 6 SET SINCE NO DRIVE ON LINE
                    /DEFAULTS TO 9 TRK,
2021    4537        T07C,   OFFLIN                  /FUNCTION TO 00,
2022    1025                TAD     K40
2023    3154                DCA     GOOD
2024    1005                TAD     K3              /LOAD 9 TRK 800BPI
2025    4477                LCMR
2026    4507                RFSR
2027    4533                COMPAR
2030    7025                ER07C
                    /GD=GOOD FS; BD=REAL FS,
2031    5224                JMP     T07C+3          /SUBTEST LOOP,

                    /VERIFY THAT RFSR CLEARS AC PRIOR TO RE[illegible]NG,
```

```
2032  7240  T07D,   CLA CMA                  /AC=7777,
2033  4507           RFSR                    /READ FS-9TRK BIT SHOULD
2034  4533           COMPAR                  /STILL BE SET,
2035  7027           ER07D
                /GD=GOOD FS; BD=REAL FS,
2036  5232           JMP     T07D            /SUBTEST LOOP

                /VERIFY THAT CM BIT 11 = 0 FORCES 7 CHANNEL,  (FS BIT 6 = 0)
2037  3154  T07E,   DCA     GOOD
2040  7126           STL RTL                 /DENSITY BITS=10
2041  4477           LCMR
2042  4507           RFSR
2043  4533           COMPAR
2044  7031           ER07E
                /GD=GOOD FS;  BD=REAL FS,
2045  5240           JMP     T07E+1          /SUBTEST LOOP,

2046  4530           LOOP7
2047  5217           JMP     T07C-2          /*****LOOP 7*****

2050  5600           JMP I   TEST07
2051  3774  K3774,   3774
```

```
                    /TEST 10,       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 (T10A-T10J)
                            /LOOP 7 CYCLES ON IF TESTS (T10K-T10R)
                            /LOOP 8 CYCLES ON R/C AND EMA CHECKS (T10S-T10W)
                            /LOOP 9 CYCLES ON EF CHECKS (T10X-T103)



        2200                PAGE
2200    0000    TEST10, 0
2201    4523            CLEAR1

2202    4204            JMS     T10A            /EXECUTE T10A
2203    5216            JMP     T10B            /THEN GO TO T10B

                    /VERIFY CLT CLEARS MTTF,  CHECK VIA SKTD NOT SKIPPING,
2204    0000    T10A,   0                       /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            ER10A
2214    5206            JMP     T10A+2          /SUBTEST LOOP,
2215    5604            JMP I   T10A

                    /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    T10B,   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            ER10B
2225    5216            JMP     T10B            /SUBTEST LOOP,

2226    4204            JMS     T10A            /T10A 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    T10C,   CLEAR2                  /CLEAR ALL,
2232    4473            LWCR                    /ENSURE WC=0000
  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            ER10C
2243  5231            JMP     T10C            /SUBTEST LOOP,

                /VERIFY WCOV AND AN IOT OTHER THAN LDBR
                /DOES NOT SET MTTF.  VARIABLE IOT (11) INSERTED BY EXECUTIVE,
2244  4524      T10D, CLEAR2
2245  4761            JMS I   T10X1P          /GET IOT,
2246  7477            T10LS1-1
2247  7753            -25
2250  2253            .+3
2251  4545            SPCFWD
2252  4553            WCOV
2253  0000            0                       /PUT IT HERE,

2254  7000            NOP
2255  7200            CLA
2256  4514            SKTD                    /SKTD SHOULD NOT SKIP,
2257  7040            CMA
2260  4533            COMPAR
2261  7041            ER10D
                /I1=VARIABLE IOT
2262  5762            JMP I   T10X2P          /SUBTEST LOOP,
2263  4763            JMS I   T10X3P          /GET MORE IOT'S,



                /VERIFY THAT REWIND FUNCTION WITHOUT RW STATUS (SINCE
                /NO DRIVE ON LINE) DOES NOT SET MTTF,
2264  4524      T10E, CLEAR2
2265  4540            REWIND
2266  4514            SKTD                    /SHOULD NOT SKIP,
2267  7040            CMA
2270  4533            COMPAR
2271  7043            ER10E
2272  5264            JMP     T10E            /SUBTEST LOOP,

                /VERIFY SPACE REVERSE WITHOUT BOT DOES NOT SET MTTF,
2273  4524      T10F, CLEAR2
2274  4546            SPCREV
2275  4514            SKTD
2276  7040            CMA
2277  4533            COMPAR
2300  7045            ER10F
2301  5273            JMP     T10F            /SUBTEST LOOP,

                /VERIFY READ, READ COMPARE OR WRITE WITHOUT LPCS DOES NOT SET
                /MTTF,
2302  4524      T10G, CLEAR2
2303  4541            READ
2304  4542            RDCOMP
```

```
2305  4543            WRITE
2306  4514            SKTD                    /SHOULD NOT SKIP.
2307  7040            CMA
2310  4533            COMPAR
2311  7047            ER10G
2312  5302            JMP     T10G            /SUBTEST LOOP.

                /VERIFY "GO" BIT SETS WHEN LOADED AND MTTF CLEAR.
2313  1027    T10H,   TAD     K100
2314  3154            DCA     GOOD
2315  4524            CLEAR2
2316  4550            GO
2317  4507            RFSR
2320  0027            AND     K100
2321  4533            COMPAR
2322  7051            ER10H
                /GD= GOOD "GO" BIT; BD= REAL "GO" BIT.
2323  5315            JMP     T10H+2          /SUBTEST LOOP.

                /VERIFY "GO" BIT CLEARS AFTER LOADED WITH MTTF SET.
2324  3154    T10I,   DCA     GOOD
2325  4545            SPCFWD                  /SET MTTF.
2326  4553            WCOV
2327  4501            LDBR                    /SET MTTF.
2330  4550            GO                      /SET "GO"; SHOULD CLEAR
2331  4507            RFSR                    /IMMEDIATELY.
2332  0027            AND     K100
2333  4533            COMPAR
2334  7053            ER10I
                /GD= GOOD "GO" BIT; BD= REAL GO BIT.
2335  5325            JMP     T10I+1          /SUBTEST LOOP.



                /VERIFY CLT CLEARS "GO" BIT.
2336  4524    T10J,   CLEAR2
2337  4550            GO                      /SET "GO"
2340  4524            CLEAR2                  /CLT
2341  4507            RFSR                    /CHECK CLEAR.
2342  0027            AND     K100
2343  4533            COMPAR
2344  7055            ER10J
                /GD= GOOD "GO" BIT; BD= REAL "GO" BIT.
2345  5337            JMP     T10J+1          /SUBTEST LOOP.

2346  4527            LOOP6
2347  5201            JMP     TEST10+1        /*****LOOP 6*****

                /VERIFY CLT CLEARS ILLEGAL FUNCTION (IF), ACTUALLY IT SHOULD
                /NEVER BE SET IN THIS TEST - SEE T10Q.
2350  3154    T10K,   DCA     GOOD
2351  4524            CLEAR2
2352  4505            RMSR
2353  0020            AND     K1
      1 4533            COMPAR
```

```
2355  7057            ER10K
                    /GD= GOOD "IF"; BD= REAL "IF"
2356  5351            JMP     T10K+1          /SUBTEST LOOP

2357  5760            JMP I   .+1
2360  2400            T10O
2361  1213    T10X1P, T3EX1
2362  1227    T10X2P, T3EX3+2
2363  1231    T10X3P, T3EX2

      2400            PAGE

                    /VERIFY WRITE OR WEOF WITHOUT FILE PROTECT DOES NOT SET IF,
2400  4524    T10O,   CLEAR2
2401  4543            WRITE
2402  4544            WEOF
2403  4505            RMSR
2404  0020            AND     K1
2405  4533            COMPAR
2406  7061            ER10O
                    /GD= GOOD IF; BD= REAL IF (BIT 11)
2407  5200            JMP     T10O            /SUBTEST LOOP

                    /VERIFY LCMR, LFGR OR LDBR WITHOUT CONTROL BUSY DO NOT
                    /SET IF,
2410  4524    T10P,   CLEAR2
2411  4477            LCMR
2412  4500            LFGR
2413  4501            LDBR
2414  4505            RMSR
2415  0020            AND     K1
2416  4533            COMPAR
2417  7063            ER10P
                    /GD= GOOD IF; BD= REAL IF (BIT 11)
2420  5210            JMP     T10P            /SUBTEST LOOP

                    /VERIFY 9 CHANNEL AND CM10=0 AND "GO" DO NOT SET IF,
                    /("GO" SHOULD NOT GENERATE "PRESET" NOW,)



2421  4524    T10Q,   CLEAR2
2422  4550            GO                      /CM10=0 FROM CLEAR',
2423  4505            RMSR
2424  0020            AND     K1
2425  4533            COMPAR
2426  7065            ER10Q
                    /GD= GOOD IF; BD= REAL IF (BIT 11)
2427  5221            JMP     T10Q            /SUBTEST LOOP,

                    /VERIFY SPCREV WITHOUT BOT DOES NOT SET IF,
2430  4524    T10R,   CLEAR2
2431  4546            SPCREV
2432  4505            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  0023            AND     K10
2505  4533            COMPAR
2506  7075            ER10U
                /GD= GOOD EMA OVERFLOW; BD= REAL EMA OVERFLOW (BIT 8)
2507  5302            JMP     T10U+2          /SUBTEST LOOP
2510  5700            JMP I   T10U

                /VERIFY EMA OVERFLOW BIT SETS WHEN EMA SET TO 7
                /AND CA 0 CHANGES FROM 1 TO 0,
2511  1023    T10V,   TAD     K10
2512  3154            DCA     GOOD
2513  4524            CLEAR2
2514  7040            CMA                     /7777 TO CA
2515  4475            LCAR
2516  4536            COMAND                  /7 TO EMA
2517  0070            70
2520  4456            JMS I   FUNCP           /"GO" AND EMA INCREMENT
2521  0140            0140                    /ENABLE TO FUNC,
2522  4476            CCAR                    /0 TO CA,
2523  4507            RFSR                    /CHECK
2524  0023            AND     K10
2525  4533            COMPAR
2526  7077            ER10V
                /GD= GOOD EMA OVERFLOW; BD= REAL EMA OVERFLOW (BIT 8)
2527  5313            JMP     T10V+2          /SUBTEST LOOP

2530  4300            JMS     T10U

                /VERIFY EMA SET TO 7 AND CA 0 CHANGING FROM 0 TO 1
                /DOES NOT SET EMA OVERFLOW BIT,
2531  4524    T10W,   CLEAR2
2532  4476            CCAR                    /0 TO CA,
2533  4536            COMAND                  /7 TO EMA
2534  0070            70
2535  4456            JMS I   FUNCP           /"GO" AND EMA INCREMENT
2536  0140            0140                    /ENABLE TO FUNC,
2537  7040            CMA                     /7777 TO CA,
2540  4475            LCAR
2541  4507            RFSR                    /CHECK
2542  0023            AND     K10
2543  4533            COMPAR
2544  7101            ER10W
                /GD= GOOD EMA OVERFLOW BIT; BD= REAL EMA OVERFLOW (BIT 8)
2545  5331            JMP     T10W            /SUBTEST LOOP,

2546  4531    T10LP8, LOOP8
2547  5237            JMP     T10S-4          /*****LOOP 8*****

2550  5751            JMP I   .+1
2551  2600            T10X-2
2552  2346    T10KP,  T10K-2

      2600            PAGE

2600  4202            JMS     T10X
```

```
2601  5213            JMP     T10Z



                /VERIFY CLT CLEARS EF,
2602  0000      T10X, 0
2603  3154            DCA     GOOD
2604  4524            CLEAR2
2605  4505            RMSR
2606  0034            AND     K4000
2607  4533            COMPAR
2610  7103            ER10X
                /GD= GOOD EF; BD= REAL EF (BIT 0)
2611  5204            JMP     T10X+2          /SUBTEST LOOP,
2612  5602            JMP I   T10X

                /VERIFY R/C ERROR WITH MTTF SET SETS EF,
2613  1034      T10Z, TAD     K4000
2614  3154            DCA     GOOD
2615  4524            CLEAR2
2616  1047            TAD     XBUFFP          /GENERATE R/C ERROR,
2617  4475            LCAR
2620  1165            TAD     PRGFLD
2621  4477            LCMR
2622  7040            CMA
2623  4501            LDBR
2624  4542            RDCOMP
2625  4535            BREAK
2626  4545            SPCFWD                  /FORCE MTTF
2627  4553            WCOV
2630  4501            LDBR                    /SET MTTF,
2631  4505            RMSR                    /CHECK,
2632  0034            AND     K4000
2633  4533            COMPAR
2634  7105            ER10Z
                /GD=GOOD EF; BD=REAL EF(BIT 0)
2635  5215            JMP     T10Z+2          /SUBTEST LOOP
2636  4202            JMS     T10X

                /VERIFY EMA OVERFLOW AND MTTF SET SETS EF,
2637  1034      T101, TAD     K4000
2640  3154            DCA     GOOD
2641  4524            CLEAR2
2642  7240            CLA CMA                 /GENERATE EMA OVERFLOW
2643  4475            LCAR
2644  4536            COMAND
2645  0070            70
2646  4456            JMS I   FUNCP
2647  0140            0140
2650  4476            CCAR
2651  4545            SPCFWD                  /FORCE MTTF
2652  4553            WCOV
2653  4505            RMSR                    /CHECK
2654  0034            AND     K4000
    5 4533            COMPAR
```

```
2656  7107            ER101
                /GD=GOOD EF; BD=REAL EF(BIT 0)
2657  5241            JMP     T101+2          /SUBTEST LOOP

                /VERIFY EMA OVERFLOW WITHOUT MTTF SET DOES NOT SET EF,
2660  3154    T102,   DCA     GOOD
2661  4524            CLEAR2
2662  7240            CLA CMA                 /FORCE EMA OVERFLOW,
2663  4475            LCAR
2664  4536            COMAND
2665  0070            70
2666  4456            JMS I FUNCP

2667  0140            0140
2670  4476            CCAR
2671  4505            RMSR                    /CHECK FOR NO EF,
2672  0034            AND     K4000
2673  4533            COMPAR
2674  7111            ER102
                /GD=GOOD EF; BD=REAL EF (BIT 0)
2675  5261            JMP     T102+1          /SUBTEST LOOP,


2676  4532            LOOP9
2677  5705            JMP I   T10LPP          /*****LOOP9*****

2700  1704            TAD I   TST10P
2701  3303            DCA     .+2
2702  5703            JMP I   .+1
2703  0000            0
2704  2200    TST10P, TEST10
2705  2546    T10LPP, T10LP8
```

```
                    /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
```

```
3033  4524            CLEAR2
3034  4545            SPCFWD                  /FORCE MTTF
3035  4553            WCOV
3036  4501            LDBR                    /SET MTTF,
3037  0000  T11C1,    0                       /IOT SHOULD SKIP,
3040  7040            CMA
3041  4533            COMPAR
3042  7117            ER11C
                /I1=FAILING IOT,
3043  5233            JMP     T11C+4          /SUBTEST LOOP,
3044  4341            JMS     T11EX2



                /VERIFY SKTR,SDLE,SBRM AND CLF DO NOT SKIP WHEN
                /MTTF SET AND EF SET,
3045  4322  T11D,     JMS     T11EX1          /GET AND IOT
3046  3151            T11LS1+3
3047  7774            -4
3050  3055            T11D1
3051  4524            CLEAR2
3052  4545            SPCFWD                  /FORCE MTTF,
3053  4553            WCOV
3054  4501            LDBR                    /SET MTTF (ALSO SETS IF AND EF)
3055  0000  T11D1,    0                       /IOT SHOULD NOT SKIP,
3056  7410            SKP
3057  7040            CMA
3060  4533            COMPAR
3061  7121            ER11D
                /I1=FAILING IOT
3062  5251            JMP     T11D+4          /SUBTEST LOOP,
3063  4341            JMS     T11EX2
3064  4530            LOOP7
3065  5225            JMP     T11C-2          /*****LOOP7*****

                /VERIFY SKEF,SKTD AND SKCB SKIP WHEN MTTF AND EF SET,
3066  4322  T11E,     JMS     T11EX1          /GET AN IOT
3067  3145            T11LS1-1
3070  7775            -3
3071  3074            T11E1
3072  4524            CLEAR2
3073  4357            JMS     SETEF           /SET MTTF AND EF,
3074  0000  T11E1,    0                       /IOT SHOULD SKIP,
3075  7040            CMA
3076  4533            COMPAR
3077  7123            ER11E
                /I1=FAILING IOT,
3100  5272            JMP     T11E+4          /SUBTEST LOOP,
3101  4341            JMS     T11EX2

                /VERIFY SKTR,SDLE,SBRM AND CLF DO NOT SKIP WHEN MTTF
                /AND EF SET,
3102  4322  T11F,     JMS     T11EX1          /GET AN IOT,
3103  3151            T11LS1+3
```

```
3104  7774            -4
3105  3110            T11F1
3106  4524            CLEAR2
3107  4357            JMS     SETEF           /SET MTTF AND EF,
3110  0000  T11F1,    0                       /IOT SHOULD NOT SKIP,
3111  7410            SKP
3112  7040            CMA
3113  4533            COMPAR
3114  7125            ER11F
                /I1=FAILING IOT,
3115  5306            JMP     T11F+4          /SUBTEST LOOP,
3116  4341            JMS     T11EX2

3117  4531            LOOP8
3120  5264            JMP     T11E-2  /*****LOOP8*****
3121  5600            JMP I   TEST11

3122  0000  T11EX1,   0                       /TEST 11 672X IOT SELECTOR,
3123  1722            TAD I   T11EX1          /GET T11LS1 STARTING POINTER -1,
3124  3010            DCA     10              /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,
3130  2322            ISZ     T11EX1
3131  1722            TAD I   T11EX1          /GET POINTER TO INSERT
3132  3160            DCA     TXXTM2          /SAVE
3133  2322            ISZ     T11EX1          /UPDATE TO RETURN,
3134  1410  T11EX3,   TAD I   10              /GET IOT,
3135  3560            DCA I   TXXTM2          /INSERT IN TEST,
3136  1560            TAD I   TXXTM2
3137  3745            DCA I   I1P             /INSERT IN IOT1 FOR TYPEOUT,
3140  5722            JMP I   T11EX1          /RUN TEST,

3141  0000  T11EX2,   0                       /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  0566  I1P,      IOT1
3146  4512  T11LS1,   SKEF                    /IOT LIST FOR TEST 11,
3147  4513            SKCB
3150  4514            SKTD
3151  4512            SKEF
3152  4515            SKTR
3153  4520            SDLE
3154  4521            SBRM
3155  4516            CLF
3156  4514            SKTD

3157  0000  SETEF,    0                       /SET ERROR FLAG USING A RDCOMP
3160  3051            DCA     XBUFF+1         /FAILURE WITH MTTF SET,
3161  7240            CLA CMA
3162  4501            LDBR
      1063  1047      TAD     XBUFFP
        64  4475      LCAR
```

```
3165  1165              TAD     PRGFLD
3166  4477              LCMR
3167  4542              RDCOMP
3170  4535              BREAK
3171  4545              SPCFWD
3172  4553              WCOV
3173  4501              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
3233    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 MITF INT ENABLE SET.
3240  7240    T12C,   CLA CMA
3241  3154            DCA     GOOD
3242  4525            CLEAR3                  /CLEAR ALL,
3243  4545            SPCFWD                  /FORCE MTTF
3244  4553            WCOV
3245  4501            LDBR
3246  4536            COMAND                  /LOAD ENABLE,
3247  0100            100
3250  4326            JMS     T12EX4          /CHECK INTERRUPT,
3251  4533            COMPAR
3252  7133            ER12C
3253  5242            JMP     T12C+2          /SUBTEST LOOP

3254  4530            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
3260  4745            JMS I   SETEFP          /FORCE MTTF AND EF,
3261  4536            COMAND                  /CLEAR ENABLES,
3262  0000            0
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  7240    T12E,   CLA CMA
3270  3154            DCA     GOOD
3271  4525            CLEAR3                  /CLEAR ALL,
3272  4745            JMS I   SETEFP          /FORCE EF,
3273  4536            COMAND                  /SET EF INT ENABLE,
3274  0200            200
3275  4326            JMS     T12EX4          /CHECK INTERRUPT
3276  4533            COMPAR
3277  7137            ER12E
3300  5271            JMP     T12E+2          /SUBTEST LOOP

3301  4531            LOOP8
3302  5254            JMP     T12D-2          /*****LOOP 8*****

3303  4523            CLEAR1                  /EXIT
3304  5600            JMP I   TEST12

3305  0000    T12EX1, 0                       /TEST 12 COMMAND REGISTER CONSTANT
3306  1705            TAD I   T12EX1          /SELECTOR,  SAME BASIC METHOD
3307  3010            DCA     10
```

```
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   10
3320  3560            DCA I   TXXTM2
3321  5705            JMP I   T12EX1
3322  0000    T12EX2, 0
3323  2157            ISZ     TXXTM1
3324  5317            JMP     T12EX3
3325  5722            JMP I   T12EX2
3326  0000    T12EX4, 0                       /PUT 0'S IN INTERRUPT INDICATOR,
3327  7200            CLA                     /(THE AC,)
3330  6001            ION                     /INTERRUPT ON,
3331  7000            NOP                     /WAIT,
3332  7000            NOP
3333  6002            IOF                     /INTERRUPT OFF,
                                              /GET INDICATOR AND
3334  5726            JMP I   T12EX4          /EXIT,
3335  7240    T12INT, CLA CMA                 /MTTF, EF INT, HANDLER FOR TEST 12,
3336  5726            JMP I   T12EX4          /JUST SET AC=7777,
3337  0000    T12LS1, 0                       /NO INT ENABLE,
3340  0100            100                     /MTTF INT ENABLE
3341  0200            200                     /EF INT ENABLE,
3342  0000            0
3343  0200            200
3344  0100            100
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)

        3400                    PAGE
3400    0000    TEST13, 0
3401    4523            CLEAR1

                        /USING MTTF, VERIFY CLF CLEARS STATUS, MTTF AND EF',  THIS
                        /IS ENOUGH TO VERIFY THAT CLF IS WORKING,
3402    3154    T13A,   DCA     GOOD
3403    4545            SPCFWD                  /FORCE MTTF
3404    4553            WCOV
3405    4501            LDBR
3406    4516            CLF                     /SHOULD CLEAR MTTF
3407    4514            SKTD                    /CHECK
3410    7410            SKP
3411    7040            CMA
3412    4533            COMPAR
3413    7141            ER13A
3414    5203            JMP     T13A+1          /SUBTEST LOOP,

                        /VERIFY IOT 672X EXCEPT CLF DOES NOT CLEAR MTTF,
3415    4703    T13B,   JMS I   T13EX1          /GET AN IOT
3416    3146            T11LS1
3417    7772            -6
3420    3424            T13B1
3421    4545            SPCFWD                  /FORCE MTTF
3422    4553            WCOV
3423    4501            LDBR
3424    0000    T13B1,  0                       /EXECUTE IOT,  SHOULD NOT CLEAR
3425    7000            NOP                     /MTTF,
3426    4514            SKTD                    /CHECK,
3427    7040            CMA
3430    4533            COMPAR
3431    7143            ER13B
                        /I1= FAILING IOT
3432    5221            JMP     T13B+4          /SUBTEST LOOP
3433    4704            JMS I   T13EX2

                        /VERIFY THAT CLF DOES NOT CLEAR CA (OR ANY OTHER NON-STATUS
                        /REGISTER) SINCE "TUR" IS HIGH,
3434    7240    T13C,   CLA CMA
3435    3154            DCA     GOOD
```

```
3436  7240          CLA CMA                 /7777 TO CA,
3437  4475          LCAR
3440  4516          CLF                     /SHOULD NOT CHANGE CA,
3441  4504          RCAR                    /CHECK
3442  4533          COMPAR
3443  7145          ER13C
                /GD= GOOD CA
3444  5236          JMP     T13C+2          /SUBTEST LOOP,

3445  4527          LOOP6
3446  5202          JMP     T13A            /*****LOOP 6*****



                /VERIFY CLF DOES NOT CAUSE A BREAK REQUEST,  CHECK VIA CA
                /INCREMENT,
3447  7240    T13D, CLA CMA
3450  3154          DCA     GOOD
3451  7240          CLA CMA                 /7777 TO CA
3452  4475          LCAR
3453  4543          WRITE                   /WRITE FUNCTION,
3454  4516          CLF                     /SHOULD NOT CAUSE A BREAK
3455  4504          RCAR                    /AND CA SHOULD REMAIN UNCHANGED,
3456  4533          COMPAR
3457  7147          ER13D
                /GD= GOOD CA
3460  5251          JMP     T13D+2          /SUBTEST LOOP,

                /VERIFY SKCB, SKTD, SKEF, SKTR OR SDLE DO NOT GENERATE
                /BREAK REQUEST,  CHECK VIA WC INCREMENT,
3461  4703    T13E, JMS I   T13EX1          /GET AN IOT,
3462  3146          T11LS1
3463  7773          -5
3464  3471          T13E1
3465  4524          CLEAR2                  /CLEAR ALL,
3466  7240          CLA CMA                 /7777 TO WC,
3467  4473          LWCR
3470  4545          SPCFWD                  /SET SPACE FORWARD,
3471  0000    T13E1, 0                      /IOT SHOULD NOT CAUSE BREAK
3472  7000          NOP
3473  4503          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  4704          JMS I   T13EX2

3500  4530          LOOP7
3501  5245          JMP     T13D-2          /*****LOOP 7*****

3502  5600          JMP I   TEST13
3503  3122    T13EX1, T11EX1
3504  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)

        3600                PAGE
3600    0000    TEST14,     0
3601    4523                CLEAR1

                    /VERIFY NO READ-COMPARE ERROR WHEN MEM AND DB CONTAIN SAME DATA,
                    /COUNT PATTERN USED,
3602    3154    T14A,       DCA     GOOD
3603    4271                JMS     T14EX1
3604    4533                COMPAR
3605    7153                ER14A
                    /GD=GOOD MS MASKED; BD=REAL MS MASKED; OD=DATA IN MEM,
3606    5277                JMP     T14EX2          /SUBTEST LOOP,
3607    4301                JMS     T14EX3

3610    4527                LOOP6
3611    5202                JMP     T14A            /*****LOOP6*****

                    /VERIFY READ-COMPARE ERROR IS GENERATED WHEN ONE BIT DIFFERS IN DB
                    /AND MEM,
3612    1021    T14B,       TAD     K2              /VERIFY READ-COMPARE ERROR
3613    3154                DCA     GOOD            /WHEN ONE BIT DIFFERS IN DB
3614    4310                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,
3620    4327                JMS     T14EX6

3621    4530                LOOP7
3622    5210                JMP     T14B-2          /*****LOOP7*****

                    /VERIFY CA DOES NOT INCREMENT WHEN READ-COMPARE ERROR EXISTS,
           3    1047    T14C,       TAD     XBUFFP
```

```
3624  3154            DCA     GOOD
3625  7240            CLA CMA
3626  3051            DCA     XBUFF+1
3627  4524            CLEAR2
3630  4501            LDBR                    /GENERATE R/C ERROR,
3631  1047            TAD     XBUFFP
3632  4475            LCAR
3633  1165            TAD     PRGFLD
3634  4477            LCMR
3635  4542            RDCOMP
3636  4535            BREAK
3637  1047  T14C1,    TAD     XBUFFP          /SET UP FOR NEXT CA INC
3640  4475            LCAR                    /WITH R/C ERROR EXISTING,
3641  7240            CLA CMA
3642  4501            LDBR
3643  4535            BREAK
3644  4504            RCAR
3645  4533            COMPAR
3646  7157            ER14C
                /GD=GOOD CA,
3647  5237            JMP     T14C1   /SUBTEST LOOP,


                /VERIFY THAT CM10-11 INHIBIT READ-COMPARE CHECKS ON BITS
                /0-3 OF DB AND MEM, 7400 IN MEM, 0000 IN DB,  THERE SHOULD BE NO
                /READ-COMPARE ERROR,
3650  1036  T14D,    TAD     K7400
3651  3051            DCA     XBUFF+1
3652  3154            DCA     GOOD
3653  1262            TAD     .+7
3654  3360            DCA     T14EX8
3655  4524            CLEAR2
3656  1047            TAD     XBUFFP
3657  4475            LCAR
3660  1005            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******

3670  5600            JMP I   TEST14

3671  0000  T14EX1,  0                       /ENTER FROM T14A,
3672  4347            JMS     T14EX7          /INITIALIZE DATA AND LOOP COUNTERS,
3673  1156            TAD     OLD             /PUT OLD IN MEM,
3674  3051            DCA     XBUFF+1
3675  1156            TAD     OLD             /PUT OLD IN LOC USED TO LOAD DB,
3676  3160            DCA     TXXTM2
3677  4360  T14EX2,   JMS     T14EX8          /SET UP CA, DB, RDCOMP, BREAK AND
3700  5671            JMP I   T14EX1          /LOOK AT R/C BIT IN MS THEN GO TO
```

```
                                        /TEST 14A COMPAR SECTION,
3701  0000  T14EX3, 0                   /ENTER FROM END OF T14A,
3702  2156        ISZ     OLD           /UPDATE OLD DATE,
3703  5307        JMP     .+4           /IF OLD NOT=0 RUN NEW OLD,
3704  2375        ISZ     T14TM1        /IF OLD=0, HAS IT BEEN 0 TWICE?
3705  7410        SKP                   /NO,
3706  5701        JMP I   T14EX3        /YES,  EXIT T14A,
3707  5273        JMP     T14EX1+2      /NO,  RUN T14A AGAIN WITH NEW DATA,
3710  0000  T14EX4, 0                   /ENTER FROM T14B,
3711  4347        JMS     T14EX7        /INITIALIZE DATA AND LOOP COUNTERS,
3712  1156        TAD     OLD           /PUT OLD IN MEM,
3713  3051        DCA     XBUFF+1
3714  1156        TAD     OLD           /COMPLEMENT THE BIT IN OLD
3715  1377        TAD     T14BIT        /INDICATED BY CURRENT BIT IN
3716  3160        DCA     TXXTM2        /T14BIT AND PUT MODIFIED OLD
3717  1156        TAD     OLD           /IN LOC USED TO LOAD DB,
3720  0377        AND     T14BIT        /(COMPLEMENTING THE BIT IS
3721  7104        CLL RAL               /ACTUALLY AN EXCLUSIVE OR FUNCTION,)
3722  7041        CIA
3723  1160        TAD     TXXTM2
3724  3160        DCA     TXXTM2

3725  4360  T14EX5, JMS     T14EX8        /SET UP CA, DB, RDCOMP, BREAK, AND LOOK
3726  5710        JMP I   T14EX4        /AT R/C BIT IN MS, THEN GO TO T14B COMPAR,
3727  0000  T14EX6, 0                   /ENTER FROM END OF T14B,
3730  2376        ISZ     T14TM2        /ALL 12 BITS BEEN COMPLEMENTED
3731  7410        SKP                   /ONE AT A TIME?
3732  5337        JMP     .+5
3733  1377        TAD     T14BIT        /NO,  MOVE BIT ONE LEFT,
3734  7104        CLL RAL
3735  3377        DCA     T14BIT
3736  5314        JMP     T14EX4+4      /GO BACK AND COMPUTE NEW DB DATA,
3737  7001        IAC                   /YES,  PUT BITIN POSITION 12,
3740  3377        DCA     T14BIT
3741  2156        ISZ     OLD           /OLD + 1 TO OLD, OLD = 0?
3742  5346        JMP     .+4
3743  2375        ISZ     T14TM1        /YES, HAS IT BEEN 0 TWICE?
3744  7410        SKP
3745  5727        JMP I   T14EX6        /YES, EXIT T14B,
3746  5355        JMP     T14EX7+6      /NO, RUN COMPLEMENTING BITS ON OLD,
3747  0000  T14EX7, 0                   /INITIALIZE LOOP COUNTERS AND
3750  1037        TAD     M2            /DATA, SET UP FOR 2 RUNS
3751  3375        DCA     T14TM1        /FOR OLD (8192 DATA SETS),
3752  3156        DCA     OLD           /SET OLD TO 0 FOR A STARTER,
3753  7001        IAC                   /SET BIT TO POSITION 12 FOR
3754  3377        DCA     T14BIT        /STARTER
3755  1042        TAD     M14           /SET UP FOR 12 BIT-ROTATES
3756  3376        DCA     T14TM2        /FOR USE IN T14B ONLY,
3757  5747        JMP I   T14EX7        /EXIT
3760  0000  T14EX8, 0                   /T14A AND T14B TEST POSITION
3761  4524        CLEAR2                /GENERATE "INITIALIZE,"
3762  1160        TAD     TXXTM2        /GET DB DATA AND LOAD DB,
3763  4501        LDBR
 764  1047        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  0021            AND     K2              /R/C ERROR BIT ONLY
3774  5760            JMP I   T14EX8          /GO TO COMPAR PORTION OF TEST
3775  0000    T14TM1, 0
3776  0000    T14TM2, 0
3777  0000    T14BIT, 0
```

```
                    /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 70-72,
                            /IF AN IOT SKIPS WHICH SHOULD NEVER SKIP, IT RESULTS
                            /IN ER00C,

        4000                PAGE
4000    0000    S6700,      0
4001    6700                6700                        /LOAD IOT,
4002    5600                JMP I   S6700
4003    4347                JMS     SKIPER
4004    0000    SLWCR,      0
4005    6701                6701                        /LWCR
4006    5604                JMP I   SLWCR
4007    4347                JMS     SKIPER
4010    0000    SCWCR,      0
4011    6702                6702                        /CWCR
4012    5610                JMP I   SCWCR
4013    4347                JMS     SKIPER
4014    0000    SLCAR,      0
4015    6703                6703                        /LCAR
4016    5614                JMP I   SLCAR
4017    4347                JMS     SKIPER
4020    0000    SCCAR,      0
4021    6704                6704                        /CCAR
4022    5620                JMP I   SCCAR
4023    4347                JMS     SKIPER
4024    0000    SLCMR,      0
4025    6705                6705                        /LCMR
4026    5624                JMP I   SLCMR
4027    4347                JMS     SKIPER
4030    0000    SLFGR,      0
4031    6706                6706                        /LFGR
4032    5630                JMP I   SLFGR
4033    4347                JMS     SKIPER
4034    0000    SLDBR,      0
4035    6707                6707                        /LDBR
4036    5634                JMP I   SLDBR
4037    4347                JMS     SKIPER
4040    0000    S6710,      0
4041    6710                6710                        /READ IOT,
4042    5640                JMP I   S6710
4043    4347                JMS     SKIPER
4044    0000    SRWCR,      0
4045    6711                6711                        /RWCR
4046    5644                JMP I   SRWCR
4047    4347                JMS     SKIPER
4050    0000    SRCAR,      0
4051    6713                6713                        /RCAR
4052    5650                JMP I   SRCAR
4053    4347                JMS     SKIPER
4054    0000    SRMSR,      0
```

```
4055  6714            6714                    /RMSR
4056  5654            JMP I   SRMSR
4057  4347            JMS     SKIPER
4060  0000    SRCMR,  0
4061  6715            6715                    /RCMR
4062  5660            JMP I   SRCMR
4063  4347            JMS     SKIPER
4064  0000    SRFSR,  0
4065  6716            6716                    /RFSR
4066  5664            JMP I   SRFSR
4067  4347            JMS     SKIPER
4070  0000    SRDBR,  0
4071  6717            6717                    /RDBR
4072  5670            JMP I   SRDBR
4073  4347            JMS     SKIPER
4074  0000    S6720,  0
4075  6720            6720                    /CONTROL IOT,
4076  5674            JMP I   S6720
4077  4347            JMS     SKIPER
4100  0000    SSKEF,  0
4101  6721            6721                    /SKEF
4102  5700            JMP I   SSKEF
4103  2300            ISZ     SSKEF
4104  5700            JMP I   SSKEF
4105  0000    SSKCB,  0
4106  6722            6722                    /SKCB
4107  5705            JMP I   SSKCB
4110  2305            ISZ     SSKCB
4111  5705            JMP I   SSKCB
4112  0000    SSKTD,  0
4113  6723            6723                    /SKTD
4114  5712            JMP I   SSKTD
4115  2312            ISZ     SSKTD
4116  5712            JMP I   SSKTD
4117  0000    SSKTR,  0
4120  6724            6724                    /SKTR
4121  5717            JMP I   SSKTR
4122  2317            ISZ     SSKTR
4123  5717            JMP I   SSKTR
4124  0000    SCLF,   0
4125  6725            6725                    /CLF
4126  5724            JMP I   SCLF
4127  4347            JMS     SKIPER
4130  0000    SCLT,   0                       /CLT
4131  6712            6712
4132  5730            JMP I   SCLT
4133  4347            JMS     SKIPER
4134  0000    SSDLE,  0
4135  6726            6726                    /SDLE
4136  5734            JMP I   SSDLE
4137  4347            JMS     SKIPER
4140  0000    SSBRM,  0
4141  6727            6727                    /SBRM
4142  5740            JMP I   SSBRM
4143  4347            JMS     SKIPER
```

```
4144  0000    SCDF,   0
4145  6201    SCDFI,  CDF     /N              /HOMING,
4146  5744            JMP I   SCDF


4147  0000    SKIPER, 0                       /AN IOT SKIPPED WHICH SHOULD NEVER SKIP,
4150  3170            DCA     ACLOC           /SAVE AC AT TIME OF FAILURE,
4151  1040            TAD     M4              /GET POINTER TO IOT SUBROUTINE,
4152  1347            TAD     SKIPER          /ENTRY AND SAVE,
4153  3347            DCA     SKIPER
4154  7001            IAC                     /GET POINTER TO IOT CODE,
4155  1347            TAD     SKIPER
4156  3155            DCA     BAD
4157  1555            TAD I   BAD             /GET IOT CODE AND SAVE',
4160  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  6703            ER00C
                /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,
4170  4526            LOOP5                   /IF NOT CHECK SR5,
4171  5374            JMP     .+3             /EXECUTE IF SR5=1,
4172  1170            TAD     ACLOC           /NO SCOPE LOOP, EXIT TO CORRECT
4173  5747            JMP I   SKIPER          /POINT IN MAIN PROGRAM',
4174  1170            TAD     ACLOC           /SCOPE LOOP, SET UP AC, THIS
4175  0000            0                       /MAY OR MAY NOT BE CORRECT CONDITION,
4176  5370            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 200, WITH SR0=0.  IF SR0=1 AT TEST START TIME,
                            /USER MUST SELECT TESTS WITH MONITOR.  MONITOR THEN SETS THE
                            /SELECTED TEST BITS AND ENTERS EXEC AT REX.  ALL TESTS
                            /WILL BE RUN THE NUMBER OF TIMES SPECIFIED BY THEIR
                            /RESPECTIVE TALLY CONSTANTS UNLESS QUICK VERIFY IS SET,
                            /WHEN ALL TESTS ARE COMPLETE, EXEC GOES TO MONITOR.

      4200                  PAGE

4200  6224    EXEC,   RIF                     /GET INSTRUCTION FIELD,
4201  3165            DCA     PRGFLD          /MODIFY HOMING CDF ROUTINE,
4202  1165            TAD     PRGFLD
4203  1046            TAD     KCDF
4204  3750            DCA I   SCDFIP
4205  6201            CDF     00              /SET UP INTERRUPT LINKAGE,
4206  1347            TAD     KRMF            /PUT RMF IN 1,
4207  3751            DCA I   P1
4210  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  7240            CLA CMA                 /SET THE TTO SOFT FLAG,
4217  3173            DCA     TTOFLG
4220  7430            SZL
4221  5340            JMP     NOEXLP+2
4222  4461            JMS I   AMGBEP          /PRINT TITLE MESSAGE
4223  7163            MSG1
4224  7240            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
4230  7604            LAS                     /GET SR0 AND
4231  3157            DCA     TXXTM1          /SAVE,
4232  4461            JMS I   AMGBEP          /TELL USER THAT SR0 IS NOW A
4233  7255            MSG9                    /QUICK VERIFY SWITCH
4234  1157            TAD     TXXTM1          /GET OLD SR0
4235  7710            SPA CLA                 /GO DIRECTLY TO MONITOR?
4236  5340            JMP     NOEXLP+2        /YES.
4237  7240            CLA CMA                 /NO, SET UP TO RUN ALL TESTS,
4240  3176    REX,    DCA     TSTAT           /PUT AC IN TEST STATUS,
4241  3174            DCA     EXITFL          /CLEAR ERROR EXIT FLAG
4242  7240            CLA CMA                 /SET TEST IN PROGRESS
4243  3175            DCA     ACTFLG
4244  3346            DCA     PASCNT          /CLEAR PASS COUNTER
4245  1042    EXECL2, TAD     M14             /SET TO CHECK FOR 12 TESTS
4246  3354            DCA     EXTAL
```

```
4247  1365            TAD     TSTP            /GET TEST STARTING ADDS, TABLE POINTER
4250  3355            DCA     TSTPP
4251  1367            TAD     TALP            /GET TEST TALLY TABLE POINTER
4252  3356            DCA     TALPP
4253  3177            DCA     TSTNUM          /SET TEST NUMBER TO 0
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  7004            RAL
4260  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  7420            SNL                     /RUN THIS TEST?
4265  5315            JMP     NOTSRN          /NO
4266  7604            LAS                     /YES TEST PROGRESS REPORT?
4267  0033            AND     K2000
4270  7640            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  3760            DCA I   M2P1P
4275  4461            JMS I   AMG8EP
4276  7213            MSG2

4277  1756  NOTSPR,   TAD I   TALPP           /GET THIS TESTS TALLY AND SAVE
4300  3361            DCA     TSTTAL
4301  1755            TAD I   TSTPP           /GET TEST STARTING ADDRESS
4302  3362            DCA     EXTEMP
4303  4762  EXOMT,    JMS I   EXTEMP          /RUN TEST
4304  7604            LAS                     /LOOP ON CURRENT TEST
4305  0021            AND     K2
4306  7640            SZA CLA
4307  5303            JMP     EXOMT           /YES RUN TEST AGAIN
4310  7604            LAS                     /SR0 SET? (QUICK VERIFY)
4311  7710            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  5303            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
4320  7604            LAS                     /YES,  LOOP ON ALL SELECTED TESTS
4321  0020            AND     K1
4322  7650            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
4330  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   AMGBEP          /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,     0
4347  6244  KRMF,       RMF
4350  4145  SCOFIP,     SCOFI
4351  0001  P1,         1
4352  0002  P2,         2
4353  0003  P3,         3
4354  0000  EXTAL,      0
4355  0000  TSTPP,      0
4356  0000  TALPP,      0
4357  0000  TSTAEX,     0
4360  7216  M2P1P,      M2P1
4361  0000  TSTTAL,     0
4362  0000  EXTEMP,     0
4363  7237  M4P2P,      M4P2
4364  7236  M4P1P,      M4P1
4365  6562  TSTP,       TST-1

4366  0203  TET1P,      TEST01
4367  6576  TALP,       TAL-1
```

```
                    /MONITOR

                            /1,ENTERED UNDER ONE OF THE FOLLOWING CONDITIONS:
                            /1,1PROGRAM STARTED AT 200 WITH SR0=1,
                            /1,2ERROR OCCURS WITH SR2=0,
                            /1,3ANY FATAL ERROR OCCURS,
                            /1,4"ALTMODE" IS STRUCK ON THE KEYBOARD AFTER TEST START,
                            /1,5SELECTED TESTS ARE RUN TO COMPLETION,

      4400                  PAGE
4400  6002    MONIT,        IOF                     /INTERRUPT SYSTEM OFF,
4401  4740                  JMS I   SAVEPP          /SAVE SUBROUTINE ONE,
4402  3171                  DCA     TRACE
4403  4461                  JMS I   AMG8EP
4404  7253                  MSG8                    /PRINT "E",
4405  3337                  DCA     TSTATM          /CLEAR TEST STATUS TEMP,
4406  4743                  JMS I   LISNP           /GET KEYBOARD,
4407  7475                  -303
4410  4420                  CS                      /C STRING,
4411  7454                  -324
4412  4435                  TS                      /T STRING,
4413  7473                  -305
4414  4527                  ES                      /E STRING,
4415  7474                  -304
4416  4600                  DS                      /"D" LOOK AT MEMORY,
4417  0000                  0
4420  3335    CS,           DCA     TSTEM1          /"C" OR "EX", PUT AC IN TEMP,
4421  1175                  TAD     ACTFLG          /IF NO TEST IN PROGRESS; SYNTAX
4422  7700                  SMA CLA                 /ERROR,
4423  5742                  JMP I   QUESP
4424  4743                  JMS I   LISNP           /GET KEYBOARD,
4425  7563                  -215                    /"-",
4426  4430                  ,+2
4427  0000                  0
4430  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          /FLAG,
4434  5744                  JMP I   INTOKP          /EXIT TO INTERRUPT SERVICE,
4435  4743    TS,           JMS I   LISNP           /T STRING, GET KEYBOARD,
4436  7456                  -322
4437  4501                  TR                      /"TR"
4440  7477                  -301
4441  4517                  TA                      /"TA"
4442  0001                  1
4443  4445                  ,+2                     /"TN"
4444  0000                  0
4445  7104                  CLL RAL                 /GET FIRST NUMBER, SCALE
4446  7006                  RTL                     /AND SAVE,
4447  3335                  DCA     TSTEM1
4450  4743                  JMS I   LISNP           /GET NEXT NUMBER
4451  0021                  1
4452  4454                  ,+2
4453  0000                  0
```

```
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                    /"TNN1-"
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

      4600            PAGE
4600  1040    DS,     TAD     M4              /SET FOR 4 NUMBERS,
4601  3270            DCA     DS2
4602  4277            JMS     LISN            /LOOK FOR FIELD NUMBER,
4603  0001            1
4604  4606            .+2
4605  0000            0
4606  7104            CLL RAL                 /SCALE AND SAVE,
4607  7006            RTL
4610  3267            DCA     DS1
4611  4277            JMS     LISN            /LOOK FOR :
4612  7506            -272
4613  4615            .+2
4614  0000            0
4615  1267            TAD     DS1             /COMPUTE CDF INSTRUCTION,
4616  1046            TAD     KCDF
4617  3236            DCA     DLF
4620  3267            DCA     DS1             /CLEAR TEMP FOR NEXT 4 INPUTS,
4621  4277    DLF2,   JMS     LISN            /GET KEYBOARD,
4622  0001            1
4623  4625            .+2                     /OCTAL NUMBER,
4624  0000            0
4625  1267            TAD     DS1             /ADD TEMP,
4626  2270            ISZ     DS2             /4TH NUMBER?
4627  7410            SKP
4630  5235            JMP     .+5
4631  7104            CLL RAL                 /NO, MOVE 3 PLACES LEFT,
4632  7006            RTL                     /STORE,
4633  3267            DCA     DS1
4634  5221            JMP     DLF2            /GO TO NEXT NUMBER,
4635  3267            DCA     DS1             /YES, 4TH NUMBER, SAVE ADDRESS,
4636  6201    DLF,    CDF     /N              /CHANGE TO REQUESTED DATA FIELD,
4637  1667            TAD I   DS1             /GET CONTENTS,
4640  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  7307            MSG11
4647  4277            JMS     LISN
4650  7563            -215                    /"D-"
4651  4665            DCR
4652  7566            -212
4653  4655            .+2                     /"DLF"
4654  0000            0
4655  1031            TAD     K215            /"DLF", DUMP NEXT LOCATION,
4656  4470            JMS I   TYPEP
4657  2267            ISZ     DS1
4660  5236            JMP     DLF
4661  1236            TAD     DLF
```

```
4662  1023          TAD     K10
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  0000  DS1,    0
4670  0000  DS2,    0
4671  7310  M11P1P, M11P1
4672  7311  M11P2P, M11P2
4673  4461  QUES,   JMS I   AMG8EP          /TYPE "?+", RING BELL,
4674  7241          MSG6
4675  5676          JMP I   MONP            /RESTART MONITOR,
4676  4403  MONP,   MONIT+3
4677  0000  LISN,   0                       /GET KEYBOARD AND CHECK SYNTAX,
4700  6031          KSF
4701  5300          JMP     .-1
4702  6036          KRB
4703  3346          DCA     LISNT1          /SAVE INPUT
4704  1346          TAD     LISNT1
4705  4470          JMS I   TYPEP           /ECHO,
4706  1346          TAD     LISNT1          /DO NOT CHECK "SPACE" IN SYNTAX,
4707  1347          TAD     M240
4710  7650          SNA CLA
4711  5300          JMP     LISN+1
4712  1677  LISN1,  TAD I   LISN            /GET COMPARATOR,
4713  7450          SNA
4714  5273          JMP     QUES            /IF 0, SYNTAX ERROR,
4715  7500          SMA
4716  5327          JMP     LISNUM          /IF >0, LOOK FOR OCTAL NUMBER,
4717  1346          TAD     LISNT1          /MATCH?
4720  7640          SZA CLA
4721  5324          JMP     LISN2           /NO,
4722  3346          DCA     LISNT1          /YES,  FAKE TEMP CLEAR AND
4723  5340          JMP     LISN3           /EXIT,
4724  2277  LISN2,  ISZ     LISN            /UPDATE CALL POINTERS,
4725  2277          ISZ     LISN
4726  5312          JMP     LISN1           /GET NEXT COMPARATOR,
4727  7200  LISNUM, CLA                     /OCTAL NUMBER REQUIRED,
4730  1346          TAD     LISNT1
4731  1044          TAD     M260
4732  7710          SPA CLA
4733  5324          JMP     LISN2
4734  1346          TAD     LISNT1
4735  1045          TAD     M270
4736  7700          SMA CLA
4737  5324          JMP     LISN2           /NOT AN OCTAL NUMBER
4740  2277  LISN3,  ISZ     LISN            /YES,  UPDATE CALL,
4741  1677          TAD I   LISN            /GET RETURN ADDRESS,
4742  3277          DCA     LISN
4743  1346          TAD     LISNT1          /GET 0 OR OCTAL NUMBER CODE
4744  0022          AND     K7              /MASK TO LOW ORDER 3 BITS,
4745  5677          JMP I   LISN            /EXIT,
4746  0000  LISNT1, 0
4747  7540  M240,   -240
```

```
                    /ERROR SERVICE ROUTINE.
                    /DECODES ERROR STATUS WORDS AND CARRIES OUT THE INDICATED
                    /ACTION.
                    /SAMPLE ERROR TABLE ENTRY:
                    /ER05C, STAT1   ERROR STATUS WORD,
                    /       STAT2   PRINT STATUS WORD,

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

                    /STAT2  BIT             PRINT OUT THE FOLLOWING WHEN SET,
                            /0              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)
                            /10             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

        5000                PAGE
5000    0000    ERRORS, 0                   /ENTER HERE WITH "ERROR"
5001    7200            CLA                         /ENTER HERE FROM "COMP" (PC PRESTORED)
5002    1370            TAD     K4352
5003    3467            DCA I   ERMSGP
5004    1600            TAD I   ERRORS              /GET ERROR TABLE ADDRESS AND
5005    3345            DCA     ERRPTR              /SAVE,
5006    1745            TAD I   ERRPTR              /GET STAT 1,
5007    7710            SPA CLA                     /NON-FATAL ERROR?
5010    5231            JMP     FATERR              /NO,  FATAL ERROR,
5011    4240            JMS     QPRNT               /NON-FATAL ERROR,  CHECK PRINTOUT,
5012    4764            JMS I   QHALTP              /CHECK RETURN TO MONITOR,
5013    2200            ISZ     ERRORS              /MOVE POINTER TO SUBTEST LOOP,
5014    4526            LOOPS                       /CHECK SUBTEST LOOP,
5015    5600            JMP I   ERRORS              /EXIT TO LOOP JUMP,
5016    1174            TAD     EXITFL              /"EX-" PENDING?
5017    7700            SMA CLA
```

```
5020  5227            JMP     .+7             /NO,
5021  1745            TAD I   ERRPTR          /YES,  DOES THIS ERROR REQUIRE
5022  7004            RAL                     /"EX-" SERVICE?
5023  7700            SMA CLA
5024  5227            JMP     .+3             /NO,  LEAVE EXITFL SET,
5025  2200            ISZ     ERRORS          /YES,  MOVE POINTER TO RECYCLE JUMP,
5026  3174            DCA     EXITFL          /CLEAR "EX-" FLAG,
5027  2200            ISZ     ERRORS          /MOVE POINTER ONE MORE POSITION,
5030  5600            JMP I   ERRORS          /EXIT,

5031  4461  FATERR,   JMS I   AMG8EP          /FATAL ERROR, PRINT MSG,
5032  7243            MSG7
5033  4465            JMS I   BELLP
5034  1372            TAD     K605            /PUT "FE" IN AC,
5035  4253            JMS     PRNT            /PRINT ERROR MESSAGE,
5036  3175            DCA     ACTFLG          /CLEAAR TEST IN PROGRESS FLAG,
5037  5462            JMP I   MONITP          /GO TO MONITOR,
5040  0000  QPRNT,    0                       /PRINTOUT?
5041  7604            LAS
5042  0032            AND     K400
5043  7640            SZA CLA
5044  5640            JMP I   QPRNT           /NO,  EXIT,
5045  1371            TAD     K522            /YES,  PUT "ER" IN AC AND
5046  4253            JMS     PRNT            /GENERATE ERROR PRINTOUT,
5047  7240            CLA CMA                 /BACK BIAS ERROR TABLE POINTER,
5050  1345            TAD     ERRPTR          /TO POINT TO STAT1',
5051  3345            DCA     ERRPTR
5052  5640            JMP I   QPRNT           /EXIT,

5053  0000  PRNT,     0                       /GENERATE ERROR MESSAGE,
5054  3746            DCA I   ERMP1P          /SAVE "FE" OR "ER",
5055  1745            TAD I   ERRPTR          /GET TEST NUMBER FROM
5056  4522            BSW                     /STAT1 AND PUT IN ERROR
5057  0365            AND     K17             /MESSAGE,
5060  4463            JMS I   EDITP
5061  3747            DCA I   ERMP2P
5062  1745            TAD I   ERRPTR          /GET SUBTEST LETTER FROM STAT1
5063  4522            BSW                     /AND PUT IN ERROR MESSAGE
5064  0373            AND     K7700           /AFTER APPENDING "SPACE" CODE,
5065  1025            TAD     K40
5066  3750            DCA I   ERMP3P
5067  1200            TAD     ERRORS          /EDIT RETURN JUMP FOR "PC" PRINTOUT,
5070  4463            JMS I   EDITP           /
5071  3753            DCA I   ERMP5P
5072  1464            TAD I   EDTEMP
5073  3752            DCA I   ERMP4P
5074  2345            ISZ     ERRPTR          /MOVE ERROR TABLE POINTER TO STAT2,
5075  1041            TAD     M5              /SET COUNTER FOR 5 SYMBOLS
5076  3354            DCA     PRCNT1          /BEFORE CR-LF,
5077  1042            TAD     M14             /SET COUNTER FOR 12 SYMBOLS
5100  3355            DCA     PRCNT2          /MAXIMUM,
5101  1362            TAD     SYMPTP          /GET SYMBOL ROUTINE TABLE
5102  3357            DCA     SYMBOL          /POINTER AND PUT IN TEMP,
5103  1356            TAD     ERMPTP          /GET ERROR MESSAGE INSERT
5104  3360            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,  0                       /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 0
5207  3000            DCA     0               /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,     0                       /GOOD,
5215  1154            TAD     GOOD            /GET GOOD,
5216  4330            JMS     SYMSEV
5217  0407            0407                    /"GD",
5220  5614            JMP I   GD
5221  0000    BD,     0
5222  1155            TAD     BAD             /GET BAD,
5223  4330            JMS     SYMSEV
5224  0402            0402                    /"BD",
5225  5621            JMP I   BD
5226  0000    OD,     0
5227  1156            TAD     OLD             /GET OLD
5230  4330            JMS     SYMSEV
5231  0417            0417                    /"OD",
5232  5626            JMP I   OD
5233  0000    WC,     0
5234  4503            RWCR                    /GET WC,
5235  4330            JMS     SYMSEV
5236  0327            0327                    /"WC",
5237  5633            JMP I   WC
5240  0000    CA,     0
5241  4504            RCAR                    /GET CA
5242  4330            JMS     SYMSEV
5243  0103            0103                    /"CA",
5244  5640            JMP I   CA
5245  0000    CM,     0
5246  4506            RCMR                    /GET CM
5247  4330            JMS     SYMSEV
5250  1503            1503                    /"CM",
5251  5645            JMP I   CM
5252  0000    FS,     0
5253  4507            RFSR                    /GET FS
5254  4330            JMS     SYMSEV
5255  2306            2306                    /"FS"
5256  5652            JMP I   FS
5257  0000    MS,     0
```

```
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  1367            TAD     K72
5344  3764            DCA I   ERMSYM
```

```
5345  2364              ISZ     ERMSYM          /DATA,
5346  1365              TAD     SYMTEM          /GET DATA,
5347  4463              JMS I   EDITP           /EDIT,
5350  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
5360  3763              DCA I   ERMPP           /"ERRORS" MEMORY PAGE,
5361  2330              ISZ     SYMSEV          /UPDATE RETURN,
5362  5730              JMP I   SYMSEV          /EXIT,
5363  5160    ERMPP,    PRMPT
5364  0000    ERMSYM,   0
5365  0000    SYMTEM,   0
5366  1000    K1000,    1000
5367  0072    K72,      72
5370  0377    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  7700
5376  5161
5377  0077
      5400                  PAGE

5400  3162    INTSEV, DCA     SAVEAC          /SAVE AC AND LINK,
5401  7010            RAR
5402  3163            DCA     SAVEL
5403  6201            CDF     00              /GET FIELD 0 LOC 0
5404  1662            TAD I   P0              /AND PUT IN THIS FIELD'S
5405  3000            DCA     0               /LOC 0 FOR INT RETURN,
5406  4457            JMS I   SCDFP           /CHANGE TO THIS DF,
5407  6041    INTAGN, TSF                     /TTO INTERRUPT?
5410  5215            JMP     .+5
5411  6042            TCF                     /YES, CLEAR HARDWARE FLAG
5412  7240            CLA CMA                 /AND SET SOFTWARE FLAG,
5413  3173            DCA     TTOFLG
5414  5244            JMP     INTOK           /TRY TO EXIT,
5415  6031            KSF                     /NO, KEYBOARD INTERRUPT?
5416  5232            JMP     INTNGA
5417  6036            KRB                     /YES, GO TO MONITOR IF INPUT
5420  1257            TAD     M233
5421  7450            SNA
5422  5462            JMP I   MONITP
5423  1260            TAD     M142
5424  7450            SNA
5425  5462            JMP I   MONITP
5426  1261            TAD     M1
5427  7650            SNA CLA
5430  5462            JMP I   MONITP
5431  5244            JMP     INTOK           /OTHERWISE TRY TO EXIT,
5432  1164    INTNGA, TAD     INTEF           /EF INTERRUPT NULL?
5433  7640            SZA CLA
5434  5564            JMP I   INTEF           /NO, SERVICE,
5435  1166            TAD     INTMTF          /YES, MTTF INTERRUPT NULL?
5436  7640            SZA CLA
5437  5566            JMP I   INTMTF          /NO, SERVICE,
5440  1000            TAD     0
5441  3155            DCA     BAD
5442  4534            ERROR                   /ILLEGAL INTERRUPT
5443  6677            FE00A                   /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 0; ALL OTHERS ARE CURRENT VALUES AND
                    /MAY NOT APPLY,
5444  7300    INTOK,  CLA CLL                 /NO    STORE 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, 0                        /ENQUIRE ABOUT EXISTING
5510  4461          JMS I   AMG8EP           /EXTENDED MEMORY FIELDS
5511  7273          MSG10                    /FOR THE DA MEMORY REFERENCE
5512  6031          KSF                      /TEST.
5513  5312          JMP     .-1
5514  6036          KRB
5515  3157          DCA     TXXTM1
5516  1157          TAD     TXXTM1
5517  4470          JMS I   TYPEP
5520  1157          TAD     TXXTM1
5521  1044          TAD     M260
5522  7710          SPA CLA
5523  5310          JMP     EXECFX+1
5524  1157          TAD     TXXTM1
5525  1045          TAD     M270
5526  7700          SMA CLA
5527  5310          JMP     EXECFX+1
5530  1157          TAD     TXXTM1
5531  3022          AND     K7
5532  7001          IAC
5533  7041          CIA
5534  3167          DCA     MEMFLD
5535  4466          JMS I   CRLFP
5536  5707          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,   0
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, AMGBE, AMGBE1, AMGBE2,
                    /AMGBE3, EDIT, EDTEM, EDTEM1, EDTEM2, EDTEM3, ETC, MUST BE SAVED AND RESTORED
                    /SINCE MONITOR MAY INTERRUPT THESE ROUTINES
                    /AND USE THEM ITSELF.
        6000        PAGE
6000    0000    SAVEP,  0
6001    7200            CLA
6002    1242            TAD     M16
6003    3243            DCA     SAVEP1
6004    1266            TAD     SRPTS
6005    3244            DCA     SAVEP2
6006    1247            TAD     SRHOLD
6007    3245            DCA     SAVEP3
6010    2244            ISZ     SAVEP2
6011    2245            ISZ     SAVEP3
6012    1644            TAD I   SAVEP2
6013    3246            DCA     SAVEP4
6014    1646            TAD I   SAVEP4
6015    3645            DCA I   SAVEP3
6016    2243            ISZ     SAVEP1
6017    5210            JMP     .-7
6020    5600            JMP I   SAVEP
6021    0000    RESTP,  0
6022    7200            CLA
6023    1242            TAD     M16
6024    3243            DCA     SAVEP1
6025    1247            TAD     SRHOLD
6026    3244            DCA     SAVEP2
6027    1266            TAD     SRPTS
6030    3245            DCA     SAVEP3
6031    2244            ISZ     SAVEP2
6032    2245            ISZ     SAVEP3
6033    1645            TAD I   SAVEP3
6034    3246            DCA     SAVEP4
6035    1644            TAD I   SAVEP2
6036    3646            DCA I   SAVEP4
6037    2243            ISZ     SAVEP1
6040    5231            JMP     .-7
6041    5621            JMP I   RESTP
6042    7762    M16,    -16
6043    0000    SAVEP1, 0
6044    0000    SAVEP2, 0
6045    0000    SAVEP3, 0
6046    0000    SAVEP4, 0
6047    6047    SRHOLD, .
6050    0000            0
6051    0000            0
6052    0000            0
6053    0000            0
6054    0000            0
6055    0000            0
6056    0000            0
6057    0000            0
```

```
6060  3000            Ø
6061  3000            Ø
6062  3000            Ø
6063  3000            Ø
6064  3000            Ø
6065  3000            Ø
6066  6066    SRPTS,  .
6067  6270            CRLF
6070  6244            TYPE
6071  6261            TYPTEM
6072  6200            AMG8E
6073  6237            AMG8E1
6074  6214            AMG8E2
6075  6240            AMG8E3
6076  6301            EDIT
6077  6326            EDTEM
6100  6327            EDTEM1
6101  6330            EDTEM2
6102  6331            EDTEM3
6103  6547            SRBSW
6104  6561            SRBSWT
```

```
                        /LOOP SWITCH SENSORS.
6105  0000    LAS5,   0                       /ENTER BY "LOOP5".
6106  7604            LAS
6107  0027            AND     K127
6110  7650            SNA CLA
6111  2305            ISZ     LAS5
6112  5705            JMP I   LAS5
6113  0000    LAS6,   0                       /"LOOP6"
6114  7604            LAS
6115  0025            AND     K40
6116  7650            SNA CLA
6117  2313            ISZ     LAS6
6120  5713            JMP I   LAS6
6121  0000    LAS7,   0                       /"LOOP7"
6122  7604            LAS
6123  0024            AND     K20
6124  7650            SNA CLA
6125  2321            ISZ     LAS7
6126  5721            JMP I   LAS7
6127  0000    LAS8,   0                       /"LOOP8"
6130  7604            LAS
6131  0023            AND     K10
6132  7650            SNA CLA
6133  2327            ISZ     LAS8
6134  5727            JMP I   LAS8
6135  0000    LAS9,   0                       /"LOOP9"
6136  7604            LAS
6137  0343            AND     K4
6140  7650            SNA CLA
6141  2335            ISZ     LAS9
6142  5735            JMP I   LAS9
6143  0004    K4,     4
```

```
                    /PDP8-E PACKED ANSCII MESSAGE GENERATOR.
                    /ENTERED WITH:
                    /JMS    AMG8E
                    /MSGNP  (MESSAGE POINTER)
                    /
                    /00 IS TERMINATOR, 43 (#) IS CARRIAGE RETURN, LINEFEED,
                    /THE FOLLOWING CHARACTERS ARE NOT ALLOWED! @ WHICH GIVES
                    /TERMINATOR CODE, # WHICH RESULTS IN CRLF, AND 204-207,
                    /212, 215, 375, 377, 233.
                    /
                    /EXITS WITH AC CLEAR.
    6200            PAGE

6200  0000  AMG8E,  0
6201  7300          CLA CLL
6202  1600          TAD I   AMG8E
6203  3237          DCA     AMG8E1
6204  2200          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  5205          JMP     .-6
6214  0000  AMG8E2, 0
6215  0241          AND     K77
6216  7450          SNA
6217  5600          JMP I   AMG8E
6220  3240          DCA     AMG8E3
6221  1240          TAD     AMG8E3
6222  1243          TAD     M43
6223  7640          SZA CLA
6224  5227          JMP     .+3
6225  4466          JMS I   CRLFP
6226  5614          JMP I   AMG8E2
6227  1240          TAD     AMG8E3
6230  1242          TAD     M40
6231  7710          SPA CLA
6232  1027          TAD     K100
6233  1030          TAD     K200
6234  1240          TAD     AMG8E3
6235  4470          JMS I   TYPEP
6236  5614          JMP I   AMG8E2
6237  0000  AMG8E1, 0
6240  0000  AMG8E3, 0
6241  0077  K77,    77
6242  7740  M40,    -40
6243  7735  M43,    -43

                    /TYPE SUBROUTINE.
                    /ENTER WITH ANSCII IN AC.  EXIT WITH AC CLEAR
6244  0000  TYPE,   0
6245  3261          DCA     TYPTEM          /SAVE ANSCII.
```

```
6246  6041          TSF
6247  7410          SKP
6250  5254          JMP     .+4
6251  1173          TAD     TTOFLG
6252  7700          SMA CLA
6253  5246          JMP     .+5
6254  3173          DCA     TTOFLG
6255  1261          TAD     TYPTEM
6256  6046          TLS                     /TYPE CHARACTER IF REAL TTO FLAG
6257  7200          CLA                     /IS SET OR SOFTWARE FLAG SET,
6260  5644          JMP I   TYPE
6261  0000  TYPTEM, 0

                    /RING TTY BELL,

6262  0000  BELL,   0
6263  7200          CLA
6264  1267          TAD     K207
6265  4470          JMS I   TYPEP
6266  5662          JMP I   BELL
6267  0207  K207,   207

                    /CARRIAGE RETURN-LINE FEED,
6270  0000  CRLF,   0
6271  7200          CLA
6272  1031          TAD     K215
6273  4470          JMS I   TYPEP
6274  1300          TAD     K212
6275  4470          JMS I   TYPEP
6276  4470          JMS I   TYPEP
6277  5670          JMP I   CRLF
6300  0212  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",
6301  0000  EDIT,   0
6302  3327          DCA     EDTEM1
6303  1327          TAD     EDTEM1
6304  4522          BSW
6305  4312          JMS     EDIT1
6306  3326          DCA     EDTEM
6307  1327          TAD     EDTEM1
6310  4312          JMS     EDIT1
6311  5701          JMP I   EDIT
6312  0000  EDIT1,  0
6313  3330          DCA     EDTEM2
6314  1330          TAD     EDTEM2
6315  0022          AND     K7
6316  3331          DCA     EDTEM3
6317  1330          TAD     EDTEM2
6320  7006          RTL
6321  7004          RAL
6322  0333          AND     K700
```

```
6323  1331            TAD     EDTEM3
6324  1332            TAD     K6060
6325  5712            JMP I   EDIT1
6326  0000  EDTEM,    0
6327  0000  EDTEM1,   0
6330  0000  EDTEM2,   0
6331  0000  EDTEM3,   0
6332  6060  K6060,    6060
6333  0700  K700,     700
```

```
                    /ROUTINES TO CLEAR ALL FLAGS SAFELY.
          6400              PAGE
    6400  0000    CLR1,     0                       /"CLEAR1"
    6401  4460              JMS I     INSRSP        /NULL EF AND MTTF INTERRUPTS,
    6402  0200              0
    6403  0000              0
    6404  4206              JMS       CLR2          /DO "CLEAR2"
    6405  5600              JMP I     CLR1
    6406  0000    CLR2,     0                       /"CLEAR2"
    6407  4214              JMS       CLR3          /DO "CLEAR3"
    6410  7240              CLA CMA                 /SET INTERRUPT ACTIVE FLAG
    6411  3172              DCA       INTFLG
    6412  6001              ION                     /INTERRUPT ON,
    6413  5606              JMP I     CLR2
    6414  0000    CLR3,     0                       /"CLEAR3"
    6415  7300              CLA CLL                 /ENSURE TTO SOFTWARE FLAG
    6416  1173              TAD       TTOFLG        /IS SET PRIOR TO CAF
    6417  7640              SZA CLA                 /GENERATE "INITIALIZE"
    6420  5223              JMP       .+3
    6421  6041              TSF
    6422  5216              JMP       .-4
    6423  7340              CLA CMA CLL
    6424  3173              DCA       TTOFLG
    6425  6002              IOF
    6426  4501              LDBR                    /SINCE "INITIALIZE WILL NOT CLEAR DB,
    6427  4517              CLT
    6430  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  0000  OFFLNE, 0                    /OFF LINE, (OFFLIN)
6433  4311          JMS     FUNC
6434  0000          0
6435  5632          JMP I   OFFLNE
6436  3000  REWND,  0                    /REWIND, (REWIND)
6437  4311          JMS     FUNC
6440  1000          1000
6441  5636          JMP I   REWND
6442  0000  READD,  0                    /READ, (READ)
6443  4311          JMS     FUNC
6444  2000          2000
6445  5642          JMP I   READD

6446  0000  RDCMP,  0                    /READ COMPARE (RDCOMP)
6447  4311          JMS     FUNC
6450  3000          3000
6451  5646          JMP I   RDCMP
6452  0000  WRIT,   0                    /WRITE, (WRITE)
6453  4311          JMS     FUNC
6454  4000          4000
6455  5652          JMP I   WRIT
6456  0000  EOFW,   0                    /WRITE EOF, (WEOF)
6457  4311          JMS     FUNC
6460  5000          5000
6461  5656          JMP I   EOFW
6462  0000  SPFORD, 0                    /SPACE FORWARD (SPCFWD)
6463  4311          JMS     FUNC
6464  6000          6000
6465  5662          JMP I   SPFORD
6466  0000  SPREV,  0                    /SPACE REVERSE, (SPCREV)
6467  4311          JMS     FUNC
6470  7000          7000
6471  5666          JMP I   SPREV
6472  0000  XGAP,   0                    /SET EXTENDED GAP BIT IN CONSTANT
                                         /(EXTGAP)
6473  7200          CLA
6474  1032          TAD     K400
6475  3277          DCA     KXGAP
6476  5672          JMP I   XGAP
6477  0000  KXGAP,  0
6500  0000  GOE,    0                    /SET GO BIT NOT CHANGING FR
6501  7200          CLA                  /OTHERWISE, (GO)
6502  1324          TAD     FUNCT
6503  2310          AND     K7600A
6504  1027          TAD     K100
6505  4500          LFGR
6506  7200          CLA
6507  5700          JMP I   GOE
6510  7600  K7600A, 7600
```

```
6511  0000    FUNC,   0                       /LOAD FR WITH FUNCTION +
6512  7200            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  4500            LFGR
6520  7200            CLA
6521  3277            DCA     KXGAP
6522  2311            ISZ     FUNC
6523  5711            JMP I   FUNC
6524  0000    FUNCT,  0
```

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

6525  0000      CMD,    0                           /"COMAND"
6526  7200              CLA
6527  1725              TAD I   CMD                 /GET CONSTANT.
6530  4477              LCMR                        /LOAD.
6531  7200              CLA                         /ENSURE AC CLEAR.
6532  2325              ISZ     CMD                 /UPDATE RETURN.
6533  5725              JMP I   CMD                 /EXIT.

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

6534  0000      DATBRK, 0                           /"BREAK"
6535  4521              SBRM
6536  7000              NOP
6537  7000              NOP
6540  5734              JMP I   DATBRK

6541  0000      WCOVER, 0                           /GENERATE CONDITIONS NECESSARY
6542  4550              GO                          /FOR WCOV, SET GO BIT, WC TO
6543  7240              CLA CMA                     /7777 = LEAVE REMAINDER OF FR ALONE.
6544  4473              LWCR
6545  4535              BREAK
6546  5741              JMP I   WCOVER


                    /BYTE SWAP ROUTINE.

6547  0000      SRBSW,  0
6550  7106              CLL RTL
6551  7006              RTL
6552  7006              RTL
6553  3361              DCA     SRBSWT
6554  7004              RAL
6555  1361              TAD     SRBSWT
6556  2362              AND     SRBSWN
6557  1361              TAD     SRBSWT
6560  5747              JMP I   SRBSW
6561  0000      SRBSWT, 0
6562  0077      SRBSWN, 77
```

```
6563  0203    TST,    TEST01          /EXEC TEST POINTERS,
6564  0600            TEST02
6565  1000            TEST03
6566  1235            TEST04
6567  1400            TEST05
6570  1600            TEST06
6571  2000            TEST07
6572  2200            TEST10
6573  3000            TEST11
6574  3200            TEST12
6575  3400            TEST13
6576  3600            TEST14
6577  0000    TAL,    0               /TEST TALLIES FOR: T01
6600  7766            -12             /T02
6601  0000            0               /T03
6602  7766            -12             /T04
6603  7766            -12             /T05
6604  7766            -12             /T06
6605  0000            0               /T07
6606  0000            0               /T10
6607  0000            0               /T11
6610  0000            0               /T12
6611  0000            0               /T13
6612  7766            -12             /T14
6613  4473    T1LS1,  LWCR            /T1LS1-T1LS3 ARE UTILIZED BY
6614  4475            LCAR            /THE TEST 01 LOCAL EXECUTIVE
6615  4477            LCMR            /FOR IOT DETERMINATION,
6616  4501            LDBR
6617  4500            LFGR
6620  4474    T1LS2,  CWCR
6621  4476            CCAR
6622  0000            0
6623  0000            0
6624  0000            0
6625  4503    T1LS3,  RWCR
6626  7777            -1
6627  4504            RCAR
6630  7777            -1
6631  4506            RCMR
6632  7777            -1
6633  4510            RDBR
6634  7777            -1
6635  4507            RFSR
6636  7600            7600
6637  4503            RWCR
6640  7777            -1
6641  4504            RCAR
6642  7777            -1
6643  4506            RCMR
6644  7777            -1
6645  4510            RDBR
6646  7777            -1
                /
6647  4541    T3LS1,  READ            /FUN  ON LIST USED BY
```

```
6650  4542                RDCOMP          /TEST 03 ("T3EX1"),
6651  4543                WRITE
6652  4537                OFFLIN
6653  4540                REWIND
6654  4544                WEOF
6655  4545                SPCFWD
6656  4546                SPCREV
6657  4541                READ
6660  4542                RDCOMP
6661  4543                WRITE
6662  4544                WEOF
6663  5276    SYMPT,      I1              /SYMBOL ROUTINE POINTER TABLE,
6664  5304                I2
6665  5214                GD
6666  5221                BD
6667  5226                OD
6670  5233                WC
6671  5240                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 00X)

6677  4001      FE00A,  4001      /ILLEGAL INTERRUPT IN INTSEV,  CURRENT CONTENTS OF
6700  7776              7776      /LOCATION 0 IN BAD,
6701  0002      ER00B,  0002      /BITS 4 AND 5 OF CM DID NOT CLEAR AFTER LCMR0,
6702  1020              1020      /DETECTED IN INSRS,
6703  0003      ER00C,  0003
6704  7601              7601

                        /ERROR TABLE FOR ERRORS OCCURRING IN FORMAL TESTS (CODE 01X-14X),

6705  0101      ER01A,  0101
6706  1400              1400
6707  0102      ER01B,  0102
6710  1400              1400
6711  0103      ER01C,  0103
6712  1400              1400
6713  0104      ER01D,  0104
6714  5400              5400
6715  0105      ER01E,  0105
6716  6000              6000
6717  0106      ER01F,  0106
6720  7400              7400
6721  0107      ER01G,  0107
6722  6000              6000
6723  0110      ER01H,  0110
6724  7400              7400
6725  0111      ER01I,  0111
6726  7400              7400
6727  0112      ER01J,  0112
6730  3400              3400
6731  0113      ER01K,  0113
6732  6000              6000
6733  0114      ER01L,  0114
6734  7400              7400
6735  0115      ER01M,  0115
6736  5400              5400
6737  0116      ER01N,  0116
6740  7400              7400
6741  0117      ER01O,  0117
6742  7400              7400
6743  0120      ER01P,  0120
6744  7400              7400
6745  2201      ER02A,  2201
6746  1300              1300
6747  2202      ER02B,  2202
6750  1240              1240
6751  2203      ER02C,  2203
6752  1220              1220
6753  2204      ER02D,  2204
6754  1202              1202
6755  2205      ER02E,  2205
6756  1600              1600
6757  0301      ER03A,  0301
```

```
6760  1110                1110
6761  0302     ER03B,     0302
6762  1252                1252
6763  0303     ER03C,     0303
6764  1652                1652
6765  0304     ER03D,     0304
6766  1652                1652
6767  0305     ER03E,     0305
6770  1252                1252
6771  0306     ER03F,     0306
6772  1110                1110
6773  0307     ER03G,     0307
6774  1050                1050
6775  0310     ER03H,     0310
6776  1110                1110
6777  0311     ER03I,     0311
7000  1050                1050
7001  2401     ER04A,     2401
7002  1310                1310
7003  0402     ER04B,     0402
7004  1414                1414
7005  2403     ER04C,     2403
7006  1270                1270
7007  2404     ER04D,     2404
7010  1270                1270
7011  0405     ER04E,     0405
7012  1070                1070
7013  2501     ER05A,     2501
7014  1252                1252
7015  2502     ER05B,     2502
7016  1652                1652
7017  2601     ER06A,     2601
7020  1272                1272
7021  0701     ER07A,     0701
7022  1400                1400
7023  0702     ER07B,     0702
7024  1400                1400
7025  0703     ER07C,     0703
7026  1400                1400
7027  0704     ER07D,     0704
7030  1400                1400
7031  0705     ER07E,     0705
7032  1400                1400
7033  1001     ER10A,     1001
7034  0000                0000
7035  1002     ER10B,     1002
7036  0174                0174
7037  1003     ER10C,     1003
7040  0174                0174
7041  1004     ER10D,     1004
7042  4174                4174
7043  1005     ER10E,     1005
7044  3174                3174
7045  1006     ER10F,     1006
7046  0174                0174
```

```
7047  1007    ER10G,  1007
7050  0174            0174
7051  1010    ER10H,  1010
7052  1574            1574
7053  1011    ER10I,  1011
7054  1574            1574
7055  1012    ER10J,  1012
7056  1574            1574
7057  1013    ER10K,  1013
7060  1574            1574
7061  1017    ER10O,  1017
7062  1574            1574
7063  1020    ER10P,  1020
7064  1574            1574
7065  1021    ER10Q,  1021
7066  1574            1574
7067  1022    ER10R,  1022
7070  1574            1574
7071  1023    ER10S,  1023
7072  1576            1576
7073  1024    ER10T,  1024
7074  1576            1576
7075  1025    ER10U,  1025
7076  1574            1574
7077  1026    ER10V,  1026
7100  1574            1574
7101  1027    ER10W,  1027
7102  1574            1574
7103  1030    ER10X,  1030
7104  1574            1574
7105  1032    ER10Z,  1032
7106  1576            1576
7107  1061    ER101,  1061
7110  1574            1574
7111  1062    ER102,  1062
7112  1574            1574
7113  1101    ER11A,  1101
7114  0004            0004
7115  1102    ER11B,  1102
7116  4004            4004
7117  1103    ER11C,  1103
7120  4114            4114
7121  1104    ER11D,  1104
7122  4114            4114
7123  1105    ER11E,  1105
7124  4014            4014
7125  1106    ER11F,  1106
7126  4014            4014
7127  1201    ER12A,  1201
7130  0024            0024
7131  1202    ER12B,  1202
7132  0134            0134
7133  1203    ER12C,  1203
7134  0134            0134
7135  1204    ER12D,  1204
```

```
7136  0034              0034
7137  1205    ER12E,    1205
7140  0034              0034
7141  1301    ER13A,    1301
7142  0014              0014
7143  1302    ER13B,    1302
7144  4014              4014
7145  1303    ER13C,    1303
7146  1040              1040
7147  1304    ER13D,    1304
7150  1050              1050
7151  1305    ER13E,    1305
7152  5110              5110
7153  3401    ER14A,    3401
7154  1656              1656
7155  3402    ER14B,    3402
7156  1656              1656
7157  1403    ER14C,    1403
7160  1056              1056
7161  1404    ER14D,    1404
7162  1404              1404

                /MESSAGES!

7163  4324    MSG1,     TEXT    "#TM8E CONTROL TEST PART 1##MAINDEC-08-DHTMA-A#"
7164  1570
7165  0540
7166  0317
7167  1624
7170  2217
7171  1440
7172  2405
7173  2324
7174  4020
7175  0122
7176  2440
7177  6143
7200  4315
7201  0111
7202  1604
7203  0503
7204  5560
7205  7055
7206  0410
7207  2415
7210  0155
7211  0143
7212  0000

                /MSG2,    TEXT    "#TEST NN#"
7213  4324    MSG2,     4324                  /CR,T
7214  0523              0523                  /E,S
7215  2440              2440                  /T,SP
7216  0000    M2P1,     0000                  /TEST NUMBER FILL IN,
7217  4300              4300                  /+, END MSG
```

```
7220  4323    MSG3,   TEXT    "#SELECTED TESTS DONE#"
7221  0514
7222  0503
7223  2405
7224  0440
7225  2405
7226  2324
7227  2340
7230  0417
7231  1605
7232  4300

              /MSG4,  TEXT    "#PASS NNNN#"
7233  4320    MSG4,   4320                    /CR,P
7234  0123            0123                    /A,S
7235  2340            2340                    /S,SP
7236  0000    M4P1,   0                       /PASS NUMBER
7237  0000    M4P2,   0                       /FILL IN,
7240  0000            0000
7241  7743    MSG6,   TEXT    "?#"
7242  0000

7243  4343    MSG7,   TEXT    "##FATAL ERROR#"
7244  0601
7245  2401
7246  1440
7247  0522
7250  2217
7251  2252
7252  0000

7253  4333    MSG8,   TEXT    "#["
7254  0000

7255  4323    MSG9,   TEXT    "#SET SR0 FOR QUICK VERIFY#"
7256  0524
7257  4023
7260  2260
7261  4006
7262  1722
7263  4021
7264  2511
7265  2313
7266  4026
7267  0522
7270  1106
7271  3143
7272  0000

7273  4305    MSG10,  TEXT    "#EXTENDED MEMORY?(0-/) "
7274  3024
7275  0516
7276  3405
7277  3440
```

```
7300  1505
7301  1517
7302  2231
7303  7750
7304  6055
7305  6751
7306  4000

                  /MSG11, TEXT    "  NNNN"
7307  4040        MSG11,  4040                    /SP,SP
7310  0000        M11P1,  0                       /CONTENTS OF MEMORY
7311  0000        M11P2,  0                       /FILL IN,
7312  0000                0                       /END MSG,

                  /THE FULL ERROR MESSAGE CALLED BY STA 12=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  0000        ERMSG,  0                       /SPACING AND SYMBOL FILL IN,
7314  0000        ERMP1,  0                       /ERROR TYPE
7315  0000        ERMP2,  0                       /TEST TYPE
7316  0000        ERMP3,  0                       /SUBTEST LETTER, SPACE
7317  4020                4020                    /SP,P
7320  0372                0372                    /C,:
7321  0000        ERMP4,  0                       /CONTENTS OF PC,
7322  0000        ERMP5,  0                       /DITTO
                          DECIMAL                 /ALL OTHER INFORMATION INSERTED HERE,
7323  0000                ZBLOCK  65
7324  0000                ZBLOCK  65
7325  0000                ZBLOCK  65
7326  0000                ZBLOCK  65
7327  0000                ZBLOCK  65
7330  0000                ZBLOCK  65
7331  0000                ZBLOCK  65
7332  0000                ZBLOCK  65
7333  0000                ZBLOCK  65
7334  0000                ZBLOCK  65
7335  0000                ZBLOCK  65
7336  0000                ZBLOCK  65
7337  0000                ZBLOCK  65
7340  0000                ZBLOCK  65
7341  0000                ZBLOCK  65
7342  0000                ZBLOCK  65
7343  0000                ZBLOCK  65
7344  0000                ZBLOCK  65
7345  0000                ZBLOCK  65
7346  0000                ZBLOCK  65
7347  0000                ZBLOCK  65
7350  0000                ZBLOCK  65
7351  0000                ZBLOCK  65
7352  0000                ZBLOCK  65
7353  0000                ZBLOCK  65
7354  0000                ZBLOCK  65
7355  0000                ZBLOCK  65
7356  0000                ZBLOCK  65
7357  0000                ZBLOCK  65
```

```
7360  0000          .BLOCK  65
7361  0000          .BLOCK  65
7362  0000          .BLOCK  65
7363  0000          .BLOCK  65
7364  0000          .BLOCK  65
7365  0000          .BLOCK  65
7366  0000          .BLOCK  65
7367  0000          .BLOCK  65
7370  0000          .BLOCK  65
7371  0000          .BLOCK  65
7372  0000          .BLOCK  65
7373  0000          .BLOCK  65
7374  0000          .BLOCK  65
7375  0000          .BLOCK  65
7376  0000          .BLOCK  65
7377  0000          .BLOCK  65
7400  0000          .BLOCK  65
7401  0000          .BLOCK  65
7402  0000          .BLOCK  65
7403  0000          .BLOCK  65
7404  0000          .BLOCK  65
7405  0000          .BLOCK  65
7406  0000          .BLOCK  65
7407  0000          .BLOCK  65
7410  0000          .BLOCK  65
7411  0000          .BLOCK  65
7412  0000          .BLOCK  65
7413  0000          .BLOCK  65
7414  0000          .BLOCK  65
7415  0000          .BLOCK  65
7416  0000          .BLOCK  65
7417  0000          .BLOCK  65
7420  0000          .BLOCK  65
7421  0000          .BLOCK  65
7422  0000          .BLOCK  65
7423  0000          .BLOCK  65
                    .BLOCK  65
                                        /(DONE BY ASSEMBLER,)

                    OCTAL


      7500    *7500
7500  4472    T10LS1, I6700
7501  4473            LWCR
7502  4475            LCAR
7503  4477            LCMR
7504  4500            LFGR
7505  4474            CWCR
7506  4476            CCAR
7507  4503            RWCR
7510  4504            RCAR
7511  4506            RCMR
7512  4507            RFSR
7513  4510            RDBR
```

```
7514  4505          RMSR
7515  4502          I6710
7516  4511          I6720
7517  4515          SKTR
7520  4514          SKTD
7521  4513          SKCB
7522  4520          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    I6700=JMS I S6700P      /IOT 6700,
4473    LWCR=JMS I SLWCRP               /AC TO WC, 0 TO AC',
4474    CWCR=JMS I SCWCRP               /0 TO WC,
4475    LCAR=JMS I SLCARP               /AC TO CA, 0 TO AC',
4476    CCAR=JMS I SCCARP               /0 TO CA,
4477    LCMR=JMS I SLCMRP               /AC TO CM, 0 TO AC',
4500    LFGR=JMS I SLFGRP               /AC0-5, 5 TO FR, 0 TO AC,
4501    LDBR=JMS I SLDBRP               /AC TO DB, 0 TO AC

                        /READ IOT'S:

4502    I6710=JMS I S6710P      /IOT 6710,
4503    RWCR=JMS I SRWCRP               /0 TO AC, WC TO AC',
4504    RCAR=JMS I SRCARP               /0 TO AC, CA TO AC',
4505    RMSR=JMS I SRMSRP               /0 TO AC, MS TO AC',
4506    RCMR=JMS I SRCMRP               /0 TO AC, CM TO AC',
4507    RFSR=JMS I SRFSRP               /0 TO AC, FR TO AC0-4, GO BIT TO AC5, S1 TO AC6-11.
4510    RDBR=JMS I SRDBRP               /0 TO AC, DB TO AC',

                        /CONTROL IOT'S:

4511    I6720=JMS I S6720P      /IOT 6720,
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                 /0 TO REGISTERS AND FLAGS IF TUR; IF NOT 0 TO MTF, EF, SR
4517    CLT=JMS I SCLTP         /POWER CLEAR TRANSPORT,
4520    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 0-5 WITH BITS 6-11 IN THE AC.
4523    CLEAR1=JMS I CLR1P      /GENERATE "INITIALIZE" AND LCMR0, 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.
4530    LOOP7=JMS I LAS7P       /CHECK SR7,  IF CLEAR, SKIP NEXT INSTRUCTION.
4531    LOOP8=JMS I LAS8P       /CHECK SR8,     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     /0000+GAP CONSTANT TO FR,
4540    REWIND=JMS I REWNP      /1000+GAP CONSTANT FR,
4541    READ=JMS I READDP       /2000+GAP CONSTANT TO FR,
4542    RDCOMP=JMS I RDCMPP     /3000+GAP CONSTANT TO FR,
4543    WRITE=JMS I WRITP       /4000+GAP CONSTANT TO FR,
4544    WEOF=JMS I EOFP         /5000+GAP CONSTANT TO FR,
4545    SPCFWD=JMS I SPFORP     /6000+GAP CONSTANT TO FR,
4546    SPCREV=JMS I SPREVP     /7000+GAP CONSTANT TO FR,
4547    EXTGAP=JMS I XGAPP      /SET GAP CONSTANT TO 0400,  CONSTANT CLEARED AFTER NEXT FR LOAD,
4550    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 01,
4551    CXXR=JMS I SIOT1P
4552    RXXR=JMS I SIOT2P
        $$$$
```

```
0000  11111100  00000000  11111111  11111111  11111111  11111111  11111111  11111111
0100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

0200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

0400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111000

0600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0700  11111111  11111111  11111111  11111111  11111100  00000000  00000000  00000000


1000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

1200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

1400  11111111  11111111  11111111  11111111  11111111  11000000  00000000  00000000
1500  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

1600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  00000000


2000  11111111  11111111  11111111  11111111  11111111  11000000  00000000  00000000
2100  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

2200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2300  11111111  11111111  11111111  11111111  11111111  11111111  11110000  00000000

2400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2500  11111111  11111111  11111111  11111111  11111111  11100000  00000000  00000000

2600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2700  11111100  00000000  00000000  00000000  00000000  00000000  00000000  00000000


3000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111000

3200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3300  11111111  11111111  11111111  11111111  11111100  00000000  00000000  00000000

3400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3500  11111000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

3600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
```

```
4000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

4200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  00000000

4400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4500  11111111  11111111  11111111  11111111  11111000  00000000  00000000  00000000

4600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4700  11111111  11111111  11111111  11111111  11111111  00000000  00000000  00000000


5000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11110000

5200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  10000111

5400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5500  11111111  11111111  11111111  11111110  00000000  00000000  00000000  00000000

5600  11111111  11111111  11111111  11111111  11100000  00000000  00000000  00000000
5700  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000


6000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6100  11111111  11111111  11111111  11111111  11110000  00000000  00000000  00000000

6200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6300  11111111  11111111  11111111  11110000  00000000  00000000  00000000  00000000

6400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

6600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111


7000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
7100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

7200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
7300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

7400  11111111  11111111  11110000  00000000  00007000  00000000  00000000  00000000
7500  11111111  11111111  11111000  00007000  00000000  00000000  00000000  00000000

7600  11111111  11111111  11111111  11111111  10000000  00007000  00000000  00000000
7700  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
```

```
AC      5271    EDIT1   6312    ER07D   7027    ERMP4   7321
ACLOC   0170    EDITP   0063    ER07E   7031    ERMP4P  5152
ACTFLG  0175    EDTEM   6326    ER101   7107    ERMP5   7322
AMG8E   6200    EDTEM1  6327    ER102   7111    ERMP5P  5153
AMG8E1  6237    EDTEM2  6330    ER10A   7033    ERMPP   5363
AMG8E2  6214    EDTEM3  6331    ER10B   7035    ERMPTP  5156
AMG8E3  6240    EDTEMP  0064    ER10C   7037    ERMSG   7313
AMG8EP  0061    EOFP    0144    ER10D   7041    ERMSGP  0067
BAD     0155    EOFW    6456    ER10E   7043    ERMSYM  5364
BD      5221    ER00B   6701    ER10F   7045    ERR1P   5636
BELL    6262    ER00C   6703    ER10G   7047    ERROR   4534
BELLP   0065    ER01A   6705    ER10H   7051    ERRORP  0134
BREAK   4535    ER01B   6707    ER10I   7053    ERRORS  5000
BSW     4522    ER01C   6711    ER10J   7055    ERRPTR  5145
CA      5240    ER01D   6713    ER10K   7057    ES      4527
CCAR    4476    ER01E   6715    ER10O   7061    EXEC    4200
CDF     6201    ER01F   6717    ER10P   7063    EXECFP  0071
CLEAR1  4523    ER01G   6721    ER10Q   7065    EXECFX  5507
CLEAR2  4524    ER01H   6723    ER10R   7067    EXECL1  4256
CLEAR3  4525    ER01I   6725    ER10S   7071    EXECL2  4245
CLF     4516    ER01J   6727    ER10T   7073    EXECP   0055
CLR1    6400    ER01K   6731    ER10U   7075    EXITFL  0174
CLR1P   0123    ER01L   6733    ER10V   7077    EXOMT   4303
CLR2    6406    ER01M   6735    ER10W   7101    EXTAL   4354
CLR2P   0124    ER01N   6737    ER10X   7103    EXTEMP  4362
CLR3    6414    ER010   6741    ER10Z   7105    EXTGAP  4547
CLR3P   0125    ER01P   6743    ER11A   7113    FATERR  5031
CLT     4517    ER02A   6745    ER11B   7115    FE00A   6677
CM      5245    ER02B   6747    ER11C   7117    FS      5252
CMD     5525    ER02C   6751    ER11D   7121    FUNC    6511
CMDP    0136    ER02D   6753    ER11E   7123    FUNCP   0056
COMAND  4536    ER02E   6755    ER11F   7125    FUNCT   6524
COMP    5600    ER03A   6757    ER12A   7127    GD      5214
COMPAR  4533    ER03B   6761    ER12B   7131    GO      4550
COMPF   5631    ER03C   6763    ER12C   7133    GOE     6500
COMPP   0133    ER03D   6765    ER12D   7135    GOEP    0150
CRLF    6270    ER03E   6767    ER12E   7137    GOOD    0154
CRLFP   0066    ER03F   6771    ER13A   7141    HALTC   5211
CS      4420    ER03G   6773    ER13B   7143    HALTCP  5213
CWCR    4474    ER03H   6775    ER13C   7145    I1      5276
CXXR    4551    ER03I   6777    ER13D   7147    I1I2    5312
DATBRK  6534    ER04A   7001    ER13E   7151    I1P     3145
DATBRP  0135    ER04B   7003    ER14A   7153    I2      5304
DB      5264    ER04C   7005    ER14B   7155    I6700   4472
DCR     4665    ER04D   7007    ER14C   7157    I6710   4502
DLF     4636    ER04E   7011    ER14D   7161    I6720   4511
DLF1    4640    ER05A   7013    ERMP1   7314    INSRS   5463
DLF2    4621    ER05B   7015    ERMP1P  5146    INSRSF  5505
DS      4600    ER06A   7017    ERMP2   7315    INSRSP  0060
DS1     4667    ER07A   7021    ERMP2P  5147    INTAGN  5407
DS2     4670    ER07B   7023    ERMP3   7316    INTEF   0164
EDIT    6301    ER07C   7025    ERMP3P  5150    INTFLG  2172
```

```
INTMTF  0166
INTNGA  5432
INTOK   5444
INTOKP  4544
INTSEP  4345
INTSEV  5400
IOT1    0566
IOT1P   5303
IOT2    0571
IOT2P   5311
K1      0020
K10     0023
K100    0027
K1000   5366
K17     5165
K2      0021
K20     0024
K200    0030
K2000   0033
K207    6267
K212    6300
K215    0031
K2252   5641
K3      0005
K377    5370
K3774   2051
K4      6143
K40     0025
K400    0032
K4000   0034
K4040   0035
K4300   5166
K4324   5642
K4343   5167
K4352   5170
K522    5171
K605    5172
K6060   6332
K7      0022
K70     0026
K700    6333
K72     5367
K7400   0036
K7600   0574
K7600A  6510
K77     6241
K7700   5173
KCDF    0046
KJMP61  1757
KJMP62  1760
KJMPI3  4344
KNOP    1756
KRMF    4347
KXGAP   6477
LAS5    6105
LAS5P   0126
LAS6    6113
LAS6P   0127
LAS7    6121
LAS7P   0130
LAS8    6127
LAS8P   0131
LAS9    6135
LAS9P   0132
LCAR    4475
LCMR    4477
LDBR    4501
LFGR    4500
LISN    4677
LISN1   4712
LISN2   4724
LISN3   4740
LISNP   4543
LISNT1  4746
LISNUM  4727
LOOP5   4526
LOOP6   4527
LOOP7   4530
LOOP8   4531
LOOP9   4532
LWCR    4473
LXXR    4551
M1      5461
M11P1   7310
M11P1P  4671
M11P2   7311
M11P2P  4672
M14     0042
M142    5460
M16     6042
M2      0037
M20     0043
M233    5457
M240    4747
M260    0044
M270    0045
M2P1    7216
M2P1P   4360
M4      0040
M40     6242
M43     6243
M4P1    7236
M4P1P   4364
M4P2    7237
M4P2P   4363
M5      0041
MARK1   1622
MARK1P  1782
MARK2   1627
MARK2P  1763
MARK3   1624
MEMFLD  0167
MONIT   4400
MONITP  0062
MONP    4676
MS      5257
MSG1    7163
MSG10   7273
MSG11   7307
MSG2    7213
MSG3    7220
MSG4    7233
MSG6    7241
MSG7    7243
MSG8    7253
MSG9    7255
NOEXLP  4336
NOTSPK  4277
NOTSRN  4315
OD      5226
OFFLIN  4537
OFFLNE  6432
OFFLNP  0137
OLD     0136
P0      5462
P1      4351
P2      4352
P3      4353
PASCNT  4346
PRCNT1  5154
PRCNT2  5155
PRGFLD  0165
PRMPT   5160
PRNT    5053
PRNTCK  5135
PRNTK   5161
PRNTLP  5117
PRNTP   5637
QHALT   5200
QHALTP  5164
QPRNT   5040
QUES    4673
QUESP   4342
RCAR    4574
RCMR    4576
RDBR    4512
RDCMP   6446
RDCMPP  0142
RDCOMP  4542
READ    4541
READD   6442
READDP  0141
RESTP   6021
RESTPP  4541
REWIND  4540
REWND   6436
REWNP   0140
REX     4240
REXP    4536
RFSR    4507
RIF     6224
RMF     6244
RMSR    4505
RWCR    4503
RXXR    4552
S6700   4000
S6700P  0072
S6710   4040
S6710P  0102
S6720   4074
S6720P  0111
SAVEAC  0162
SAVEL   0163
SAVEP   6000
SAVEP1  6043
SAVEP2  6044
SAVEP3  6045
SAVEP4  6046
SAVEPP  4540
SBRM    4521
SCCAR   4020
SCCARP  0076
SCDF    4144
SCDFI   4145
SCDFIP  4350
SCDFP   0057
SCLF    4124
SCLFP   0116
SCLT    4130
SCLTP   0117
SCWCR   4010
SCWCRP  0074
SDLE    4520
SETEF   3157
SETEFP  3345
SIOT1   0565
SIOT1P  0151
SIOT2   0572
```

```
SIOI2P  0152
SKCB    4513
SKEF    4512
SKIPER  4147
SKTD    4514
SKTR    4515
SLCAR   4014
SLCARP  0075
SLCMR   4024
SLCMRP  0077
SLDBR   4034
SLDBRP  0101
SLFGR   4030
SLFGRP  0100
SLWCR   4004
SLWCRP  0073
SPCFWD  4545
SPCREV  4546
SPFORD  6462
SPFORP  0145
SPREV   6466
SPREVP  0146
SRBSW   6547
SRBSWN  6562
SRBSWP  0122
SRBSWT  6561
SRCAR   4050
SRCARP  0104
SRCMR   4060
SRCMRP  0106
SRDBR   4070
SRDBRP  0110
SRFSR   4064
SRFSRP  0107
SRHOLD  6047
SRMSR   4054
SRMSRP  0105
SRPTS   6066
SRWCR   4044
SRWCRP  0103
SSBRM   4140
SSBRMP  0121
SSDLE   4134
SSDLEP  0120
SSKCB   4105
SSKCBP  0113
SSKEF   4100
SSKEFP  0112
SSKTD   4112
SSKTDP  0114
SSKTR   4117
SSKTRP  0115
START   0200
SYMADR  5151
SYMBOL  5157
SYMPT   6663
SYMPTP  5162
SYMSEV  5330
SYMTEM  5365
T01A    0211
T01B    0216
T01C    0223
T01D    0235
T01E    0243
T01F    0255
T01G    0270
T01H    0304
T01I    0313
T01J    0323
T01K    0336
T01L    0347
T01M    0355
T01N    0362
T01O    0403
T01OL   0400
T01P    0427
T02A    0602
T02B    0617
T02C    0634
T02D    0654
T02E    0670
T02LDM  0745
T02LDR  0731
T02LDS  0707
T02LDT  0716
T02LDU  0714
T02LP9  0704
T03A    1004
T03B    1015
T03C    1027
T03D    1042
T03E    1062
T03F    1101
T03G    1120
T03H    1141
T03I    1156
T04A    1237
T04AL   1243
T04ASL  1252
T04B    1264
T04CD   1276
T04CL   1302
T04CSL  1324
T04E    1357
T04T1   1376
T04T2   1377
T05A    1402
T05ASL  1406
T05B    1424
T05BSL  1427
T05LDM  1451
T05LDS  1450
T06A    1602
T06ADF  1615
T06AMP  1623
T07A    2002
T07B    2011
T07C    2021
T07D    2032
T07E    2037
T101    2637
T102    2660
T10A    2204
T10B    2216
T10C    2231
T10D    2244
T10E    2264
T10F    2273
T10G    2302
T10H    2313
T10I    2324
T10J    2336
T10K    2350
T10KP   2352
T10LP8  2346
T10LPP  2705
T10LS1  7500
T10O    2400
T10P    2410
T10Q    2421
T10R    2430
T10S    2443
T10T    2454
T10U    2500
T10V    2511
T10W    2531
T10X    2602
T10X1P  2361
T10X2P  2362
T10X3P  2363
T10Z    2613
T11A    3003
T11B    3011
T11B1   3016
T11C    3027
T11C1   3037
T11D    3045
T11D1   3055
T11E    3066
T11E1   3074
T11EX1  3122
T11EX2  3141
T11EX3  3134
T11F    3102
T11F1   3110
T11LS1  3146
T12A    3237
T12A1   3215
T12B    3225
T12C    3240
T12D    3256
T12E    3267
T12EX1  3305
T12EX2  3322
T12EX3  3317
T12EX4  3326
T12INT  3335
T12LS1  3337
T13A    3402
T13B    3415
T13B1   3424
T13C    3434
T13D    3447
T13E    3461
T13E1   3471
T13EX1  3503
T13EX2  3504
T14A    3602
T14B    3612
T14BIT  3777
T14C    3623
T14C1   3637
T14D    3650
T14EX1  3671
T14EX2  3677
T14EX3  3701
T14EX4  3710
T14EX5  3725
T14EX6  3727
T14EX7  3747
T14EX8  3760
T14TM1  3775
T14TM2  3776
T1EX1   0450
T1EX1A  0461
T1EX1P  0375
T1EX2   0473
T1EX2P  0376
```

```
T1EX3   0501
T1EX3P  0377
T1EX4   0505
T1EX5   0540
T1EX6   0524
T1EX7   0516
T1EX8   0530
T1EXT1  0561
T1EXT2  0562
T1EXT3  0563
T1EXT4  0564
T1LP7   0371
T1LP7P  0573
T1LS1   6613
T1LS1P  0556
T1LS2   6620
T1LS2P  0557
T1LS3   6625
T1LS3P  0560
T3EX1   1213
T3EX1P  1176
T3EX2   1231
T3EX2P  1175
T3EX3   1225
T3EX32  1177
T3LOAD  1200
T3LODP  0004
T3LS1   6647
T4LP6   1262
T4LP7   1373
T5LP6   1422
T5LP7   1445
T6CNT1  1764
T6CNT2  1765
T6CNT3  1766
T6EX1   1642
T6EX2   1664
T6EX3   1727
T6EX4   1747
T6EX5   1737
T6EX6   1753
T6FLAG  1767
T6I1    1713
T6SAVE  1761
TA      4517
TAL     6577
TALP    4367
TALPP   4356
TCR     4513
TEST01  0203
TEST02  0600
TEST03  1000
TEST04  1235
TEST05  1400
TEST06  1600
TEST07  2000
TEST10  2207
TEST11  3000
TEST12  3200
TEST13  3400
TEST14  3600
TET1P   4366
TR      4501
TRACE   0171
TRPP    5640
TS      4435
TST     6563
TST10P  2704
TST1P   0447
TSTAEX  4357
TSTAT   0176
TSTATM  4537
TSTEM1  4535
TSTNUM  0177
TSTP    4365
TSTPP   4355
TSTTAL  4361
TTOFLG  0173
TXXTM1  0157
TXXTM2  0160
TXXTM3  0161
TYPE    6244
TYPEP   0070
TYPTEM  6261
WC      5233
WCOV    4553
WCOVEP  0153
WCOVER  6541
WEOF    4544
WHAT    5163
WRIT    6452
WRITE   4543
WRITP   0143
XBUFF   0050
XBUFFP  0047
XGAP    6472
XGAPP   0147
```

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 35 SECONDS

4K CORE USED