FP11-F

FP11F FLTG PNT C

CKFPCA0

AH-F638A-MC

FICHE 1 OF 2

JAN 1980

digital

MADE IN USA

FP11-F

FP11F FLTG PNT C
CKFPCA0

AH-F638A-MC

FICHE 2 OF 2

JAN 1980
digital
MADE IN USA

```
                .REM    &
```

IDENTIFICATION

PRODUCT CODE: AC-F636A-MC

PRODUCT NAME: CKFPCA0 FP11F FLTG PNT PRT C

PRODUCT DATE: OCTOBER, 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: ANTHONY VEZZA, DAN MILLEVILLE

HISTORY

NO CHANGES TO THE 11/34 FLOATING POINT DIAGNOSTIC PART 'A' WERE FOUND TO BE NEEDED TO ADAPT IT FOR USE ON THE 11/44.

THE FOLLOWING WAS ADDED TO THE 11/34 FLOATING POINT DIAGNOSTIC TO MAKE THE 'B' VERSION COVER THE 11/44:

1. TEST 22 - PROCESSOR LOOKS TO SEE IF APT IS CONTROLLING THE TEST, AND IF IT IS, CHECKS TO SEE IF THE USER HAS SELECTED THIS TEST BY CHECKING BIT 7 IN THE SWITCH REGISTER. IT HAS ALSO BEEN CHANGED SO THAT IF BIT 7 IS *ONE*, THE CODE WILL SELECT THE TEST.

THE FOLLOWING WAS ADDED TO THE 11/34 FLOATING POINT DIAGNOSTIC TO MAKE THE 'C' VERSION COVER THE 11/44:

1. TEST 76 - CHECKS THAT FP PROCESSOR DOESN'T ACCESS D-SPACE UNTIL CONDITIONS WARRANT.

2. TEST 77 - CHECKS THAT SR1 MATCHES WHAT ACTUALLY HAPPENED TO THE REGISTER OF THE INSTRUCTION, AND THAT THE VALUE OF AUTO INCREMENT/DECREMENT WAS PROPER.

CONTENTS

1. ABSTRACT

THE THREE PROGRAMS:

CKFPAA0 CKFPBA0 CKFPCA0

ARE DESIGN TO DETECT AND REPORT LOGIC FAULTS IN THE PDP 11/44 FP11-F FLOATING POINT PROCESSOR. THE DESIGN IS AN ATTEMPT TO REACH ALL ROM STATES, TAKE ALL BRANCH MICRO TESTS (BUT'S) AND VERIFY ALL THE LOGIC. THEY CONSIST OF 161 (OCT) INDIVIDUAL TESTS SEQUENCED TO DETECT AND ATTEMPT TO IDENTIFY FAULTS WITH A MINIMUM HARDWARE OR SOFTWARE LEVEL. THE TESTS ARE PARTIONED INTO THREE STAND-ALONE PROGRAMS DESCRIBED BELOW.

NOTE THAT ERROR REPORTS IN THESE PROGRAMS ARE BASED UPON THE KNOWLEDGE THAT ALL PREVIOUS TESTS HAVE BEEN RUN AND IN MOST CASE THAT THERE IS ONLY A SINGLE POINT FAULT IN THE FP11-F. IF THE PROGRAMS OR TESTS ARE NOT RUN IN ORDER THEN ERROR MESSAGES MAY NOT BE ACCURATE.

A. CKFPAA0

CKFPAA0 TESTS:

- LDFPS
- STFPS
- CFCC
- SETF, SETD, SETI AND SETL
- STST
- LDF AND LDD (ALL SOURCE MODES)
- STD (MODE 0 AND 1)
- ADDF, ADDD AND SUBD (MOST CONDITIONS)

B. CKFPBA0

CKFPBA0 TESTS:

- ADDF, ADDD AND SUBD (ALL CONDITIONS NOT TESTED IN CKFPAA0)
- CMPD AND CMPF
- DIVD AND DIVF
- MULD AND MULF
- MODD AND MODF

C. CKFPCA0

CKFPCA0 TESTS:

- STF AND STD (ALL MODES)
- STCFD AND STCDF
- CLRD AND CLRF
- NEGF AND NEGD

```
                                ABSF AND ABS
                                TSTF AND TSTD
                                NEGF, ABSF AND TSTF (ALL SOURCE MODES)
                                NEGF, ABSF AND TSTF (ALL SOURCE MODES)
                                LDFPS (ALL SOURCE MODES)
                                LDCIF AND LDCLF
                                LDCID AND LDCLD
                                LDEXP
                                STFPS (ALL DESTINATION MODES)
                                STCFL AND STCFI
                                STCDL AND STCDI
                                STEXP
                                STST
                                I AND D SPACE TESTS (ALL MODES AND REGS 0 AND 7)
                                AUTO INCREMENT/DECREMENT CHECK - SR1 (ALL MODES AND REGS 1 AND 7)


                    2.                  REQUIREMENTS
                                        ------------

                    2.1         EQUIPMENT
                                A PDP 11/44 WITH CONSOLE AND AN FP11-F FLOATING
                                POINT PROCESSOR. NOTE THAT A SPECIAL INTERRUPTS TEST
                                MODULE IS BEING DESIGNED FOR USE IN THE MANUFAC-
                                TURING ENVIRONMENT. WHEN THIS DEVICE IS PRESENT THE
                                PROGRAM CKFPBA0 WILL MAKE USE OF IT TO TEST THE FPP
                                INTERRUPT ON BUS REQUEST FUNCTIONS.

                    2.2         STORAGE

                                ALL THREE PROGRAM REQUIRE A MEMORY SYSTEM OF AT
                                LEAST 16K TO LOAD AND RUN.

                    2.3         PRELIMINARY PROGRAMS

                                THESE THREE DIAGNOSTICS WILL ASSUME THAT THE PDP
                                11/44 CENTRAL PROCESSOR IS FAULTLESS, THEREFORE WHEN
                                IN DOUBT RUN THE PDP 11/44 PROCESSOR DIAGNOSTICS
                                BEFORE THESE FP11-F DIAGNOSTICS.


                    3.                  LOADING PROCEDURE
                                        -----------------

                                THE PROGRAMS WILL BE SUPPLIED ON THE USUAL
                                DIAGNOSTIC MEDIA. REFER TO THE XXDP OPERATING
                                MANUAL FOR FURTHER INFORMATION.

                    4.                  STARTING PROCEDURE
                                        ------------------

                    4.1         CONTROL SWITCH SETTINGS

                                SEE SECTION 5.1

                    4.2         PROGRAM AND OPERATOR ACTION
```

1. LOAD PROGRAM INTO MEMORY
2. LOAD ADDRESS 200
3. SET CONSOLE SWITCHES (IF CONSOLE IS PRESENT)
4. PRESS START.
   ON FIRST PASS, THE PROGRAM WILL IDENTIFY ITSELF. NOTE THAT IF THERE IS NO PHYSICAL CONSOLE THE PROGRAM WILL REQUEST THE OPERATOR FOR INITIAL VALUE FOR THE SOFTWARE SWITCH REGISTER (SEE SECTION 8.5). OF RUNNING UNDER ACT, APT OR CHAIN THIS DOES NOT APPLY.
5. THE PROGRAM WILL LOOP AND AN END OF PASS AND ERROR SUMMARY WILL BE TYPED AT THE END OF EVERY PASS.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

THE SWITCH SETTING ARE:

	OCTAL	
SW<15>=1...	100000	HALT ON ERROR
SW<14>=1...	40000	LOOP ON CURRENT TEST
SW<13>=1...	20000	INHIBIT ERROR TYPE OUTS
SW<12>=1...	10000	INHIBIT T-BIT TRAPPING
SW<11>=1...	4000	INHIBIT ITERATIONS
SW<10>=1...	2000	RING TTY BELL ON ERROR
SW<9>=1....	1000	LOOP ON ERROR
SW<8>=1....	400	LOOP ON TEST SPECIFIED IN SW<6> THROUGH SW<0>
SW<7>=1....	200	PRINT ERROR SUMMARY EVEN IF SW<13>=1, THIS APPLIES ONLY TO PROGRAM CKFPAA0.
SW<7>=1....	200	SELECT CORRECT INTERRUPT TEST IN PROGRAM CKFPBA0.

6. ERRORS

6.1 SUMMARIES

IN PROGRAM CKFPAA0 TESTS 1 AND 11 HAVE A SPECIAL ERROR SUMMARY FEATURE. THESE TWO TEST RUN MANY TEST PATTERNS THROUGH THE LOGIC. AFTER AN ERROR IS ENCOUNTERED, ONLY THE FIRST FIVE ERRORS ARE REPORTED (TYPED ON THE TTY). EVERY ERROR THOUGH IS LOGGED AND AN ERROR SUMMARY IS PRINTED WHEN THE TEST IS COMPLETE. NOTE THAT IS SW<13>=1 THIS SUMMARY WILL NOT BE TYPED UNLESS SW<7>=1. IN OTHER WORDS TO GET JUST AN ERROR SUMMARY FROM EITHER OF THESE TWO TESTS 1 AND 11 IN PROGRAM CKFPAA0 BOTH SWITCHES 13 AND 7 MUST = 1.

6. ERROR RECOVERY

```
                                SW<15:9>=0...   MOST ERRORS WILL CAUSE EXECUTION  TO
                                                GO  TO  THE  START  OF THE NEXT TEST
                                                AFTER THE MESSAGE IS TYPED.   A  FEW
                                                TESTS  ARE  IN  SECTIONS.   IN THESE
                                                TESTS AN ERROR WILL CAUSE  EXECUTION
                                                TO  GO TO THE NEXT SECTION AFTER THE
                                                MESSAGE IS TYPED.
                                SW<15>=1..      THE PROGRAM WILL HALT  AFTER  TYPING
                                                THE  ERROR  MESSAGE.   PRESSING  THE
                                                CONSOLE  CONTINUE  WILL  CAUSE   THE
                                                PROGRAM TO CONTINUE AS IF SW<15>=0.

                        7.                      RESTRICTIONS
                                                ------------

                                NONE


                        8.                      MISCELLANEOUS
                                                -------------

                        8.1     EXECUTION TIMES

                                LESS THAN 10 SECONDS FOR EACH PROGRAM ON ANY PASS.

                        8.2     STACK POINTER

                                THE STACK POINTER IS INITIALIZED TO 1100 IN EACH  OF
                                THE THREE PROGRAMS.

                        8.3     PASS COUNT
                                THE PROGRAM MAKES ONE PASS  FOR  EACH  END  OF  PASS
                                MESSAGE  TYPED.   THE  END OF PASS MESSAGE DESCRIBES
                                THE TOTAL NUMBER OF PASSES COMPLETED AND  THE  TOTAL
                                NUMBER OF ERRORS SINCE THE LAST END OF PASS MESSAGE.

                        8.4     T-BIT TRAPPING

                                IF SW<12>=0 EACH PROGRAM WILL RUN WITH  TRACE  TRAPS
                                ON  EVERY  OTHER  PASS.   FIRST PASS WILL NOT ENABLE
                                TRACE TRAPS.  NOTE SW<12>=1 DISABLES T-BIT TRAPS.

                        8.5     SOFTWARE SWITCH REGISTER

                                IF THE USER DESIRES, A  SOFTWARE SWITCH REGISTER CAN
                                BE EXAMINED OR MODIFIED  AT ANY  TIME BY THE USER IF
                                HE TYPES CNTRL/G WHILE THE PROGRAM IS RUNNING.  THIS
                                CNTRL/G  WILL CAUSE  THE  CONTENTS OF  THE  SOFTWARE
                                SWITCH REGISTER TO BE TYPED  ON  THE TTY AND ASK THE
                                USER FOR A NEW VALUE.  WHEN THE  USER TYPES  A VALUE
                                AND  CARRIAGE RETURN  THEN  THE PROGRAM  WILL RESUME
                                TESTING AT THE SAME POINT AT  WHICH IT LEFT OFF WHEN
                                THE USER TYPED CNTRL/G.  NOTE THAT  WHEN NOT RUNNING
                                UNDER ACT, APT  OR CHAIN  THE  USER  WILL  BE  ASKED
                                FOR A SOFTWARE SWITCH REGISTER VALUE  AFTER  LOADING
                                ADDRESS  200 AND STARTING THE PROGRAM THE FIRST TIME
```

THE PROGRAM IS RUN AFTER LOADING ONLY IF THE CONSOLE SWITCH REGISTER CONTAINS 177777.

8.6 INTERRUPTS TEST

IN PROGRAM CKFPBA0 THERE IS A SPECIAL TEST FOR CHECKING THE CORRECT FLOWS OF THE FPP. THIS TEST CAN BE RUN ONLY IF A SPECIAL TEST MODULE IS IN THE SYSTEM. THIS MODULE WILL PROBABLY ONLY BE USED IN MANUFACTURING. IF THIS MODULE IS NOT IN THE SYSTEM THIS TEST WILL AUTOMATICALLY BE DESELECTED. IF THIS TEST MODULE IS ON THE SYSTEM AND SW<7>=1 THIS TEST WILL BE RUN. IF SW<7>=0, THIS TEST WILL BE DESELECTED.

8.7 ACT, APT AND XXDP COMPATIBILITY

THESE PROGRAMS ARE FULLY COMPATIBLE WITH:

- APT
- ACT
- XXDP MONITOR AND CHAIN PROGRAMS.

9. PROGRAM DESCRIPTION

TEST 1 — STF WITH ILLEGAL ACCUMULATOR TEST

THIS IS A TEST OF THE ST INSTRUCTION USING ILLEGAL ACCUMULATOR 7, MODE 0.

TEST 2 — FDST MODE 1, FLOATING MODE, TEST

THIS IS A TEST OF THE STF INSTRUCTION USING FDST MODE 1.

TEST 3 — FDST MODE 2 TEST

THIS IS A TEST OF BOTH STF AND STD WITH FDST MODE 2.

TEST 4 — FDST MODE 2, WITH GR7, TEST

THIS IS A TEST OF STF WITH GR7 MODE 2 OR IMMEDIATE MODE.

```
                        TEST 5          FDST MODE 4 TEST
                        ------          ----------------

                                THIS IS A TEST OF STD WITH FDST MODE 4.

                        TEST 6          FDST MODE 3 TEST
                        ------          ----------------

                                THIS IS A TEST OF FDST MODE 3 USING STD.

                        TEST 7          FDST MODE 5 TEST
                        ------          ----------------

                                THIS IS A TEST OF FDST MODE 5 USING STD.

                        TEST 10         FDST MODE 6, INDEX MODE, TEST
                        -------         -----------------------------

                                THIS IS A TEST OF FDST MODE  6,  INDEX  MODE,  USING
                                STD.

                        TEST 11         FDST MODE 7, INDEX DEFERRED MODE, TEST
                        -------         --------------------------------------

                                THIS IS A TEST OF FDST MODE 7, INDEX DEFERRED  MODE,
                                USING STD.

                        TEST 12         STCFD TEST
                        -------         ----------

                                THIS IS A TEST OF THE STCFD INSTRUCTION.

                        TEST 13         STCDF TEST
                        -------         ----------

                                THIS IS A TEST OF THE STCDF INSTRUCTION.

                        TEST 14         STCFD WITH ILLEGAL ACCUMULATOR TEST
                        -------         -----------------------------------

                                THIS TEST STCFD WITH ILLEGAL AC 6.

                        TEST 15         CLRD TEST
                        -------         ---------

                                THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS.

                        TEST 16         CLRD WITH ILLEGAL ACCUMULATOR TEST
                        -------         ----------------------------------

                                THIS IS A TEST OF CLRD WITH ILLEGAL AC7.

                        TEST 17         NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
                        -------         -------------------------------------------------------

                                THIS IS A TEST OF THE SPECIAL DEST FLOWS  USING  THE
```

```
                                NEGD INST WITH MODE ZERO AND ILLEGAL AC7.

TEST 20         NEGF, ABSF AND TSTF SOURCE MODE 0 TEST
-------         --------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE NEGD INSTRUCTION IS USED TO TEST MODE 0

TEST 21         NEGF, ABSF AND TSTF SOURCE MODE 1 TEST
-------         --------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE NEGD INSTRUCTION IS USED TO TEST MODE 1

TEST 22         NEGF, ABSF AND TSTF SOURCE MODE 2 TEST
-------         --------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE ABSD INSTRUCTION IS USED TO TEST MODE 2

TEST 23         NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
-------         --------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE ABSD INSTRUCTION IS USED TO TEST MODE 4

TEST 24         NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
-------         --------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE ABSD INSTRUCTION IS USED TO TEST MODE 3

TEST 25         NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
-------         --------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE NEGD INSTRUCTION IS USED TO TEST MODE 5

TEST 26         NEGF, ABSF AND TSTF SOURCE MODE 6 TEST
-------         --------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE ABSD INSTRUCTION IS USED TO TEST MODE 6

TEST 27         NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
-------         --------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE ABSD INSTRUCTION IS USED TO TEST MODE 6

TEST 30         NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST
-------         --------------------------------------------

                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                THE NEGD INSTRUCTION IS USED TO TEST MODE 6

TEST 31         NEGF, ABSF AND TSTF SOURCE MODE 7, GR7, TEST
```

```
                        ------          ----------------------------------------

                                THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
                                THE ABSD INSTRUCTION IS USED TO TEST MODE 7

                        TEST 32         SPECIAL DEST, MODE 0, TEST
                        ------          --------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 0 USING THE NEGD INSTR.

                        TEST 33         SPECIAL DEST, MODE 1, TEST
                        ------          --------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 1 USING THE NEGD INSTR.

                        TEST 34         SPECIAL DEST, MODE 2, TEST
                        ------          --------------------------
                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 2 USING THE NEGD INSTR.

                        TEST 35         SPECIAL DEST, MODE 4, TEST
                        ------          --------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 4 USING THE NEGD INSTR.

                        TEST 36         SPECIAL DEST, MODE 3, TEST
                        ------          --------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 3 USING THE NEGD INSTR.

                        TEST 37         SPECIAL DEST, MODE 5, TEST
                        ------          --------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 5 USING THE NEGD INSTR.

                        TEST 40         SPECIAL DEST, FLOATING MODE 2, TEST
                        ------          ----------------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 2 USING THE NEGF INSTR.

                        TEST 41         SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST
                        ------          ------------------------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 2(IMMEDIATE) USING THE NEGD INSTR.

                        TEST 42         SPECIAL DEST, MODE 6, TEST
                        ------          --------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 6 USING THE NEGD INSTR.
```

```
                        TEST 43         SPECIAL DEST, MODE 7, TEST
                        -------         --------------------------

                                THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
                                FLOWS MODE 7 USING THE NEGD INSTR.

                        TEST 44         NEGD, ABSD AND TSTD TEST
                        -------         ------------------------

                                THIS IS A TEST OF THE NEGD ABSD AND TSTD
                                INSTRUCTIONS.

                        TEST 45         SOURCE MODES, MODE 1 (FL=0), TEST
                        -------         ---------------------------------

                                THIS IS A TEST OF SOURCE MODE 1 USING THE LDFPS
                                INSTRUCTION.

                        TEST 46         SOURCE MODES, MODE 2 (FL=0), TEST
                        -------         ---------------------------------

                                THIS IS A TEST OF SOURCE MODE 2 USING THE LDFPS
                                INSTRUCTION.

                        TEST 47         SOURCE MODES, MODE 4 (FL=0), TEST
                        -------         ---------------------------------

                                THIS IS A TEST OF SOURCE MODE 4 USING THE LDFPS
                                INSTRUCTION.

                        TEST 50         SOURCE MODES, MODE 3 (FL=0), TEST
                        -------         ---------------------------------

                                THIS IS A TEST OF SOURCE MODE 3 USING THE LDFPS
                                INSTRUCTION.

                        TEST 51         SOURCE MODES, MODE 5 (FL=0), TEST
                        -------         ---------------------------------

                                THIS IS A TEST OF SOURCE MODE 5 USING THE LDFPS
                                INSTRUCTION.

                        TEST 52         SOURCE MODES, MODE 6 (FL=0), TEST
                        -------         ---------------------------------

                                THIS IS A TEST OF SOURCE MODE 6 USING THE LDFPS
                                INSTRUCTION.

                        TEST 53         SOURCE MODES, MODE 7 (FL=0), TEST
                        -------         ---------------------------------

                                THIS IS A TEST OF SOURCE MODE 7 USING THE LDFPS
                                INSTRUCTION

                        TEST 54         SOURCE MODES, MODE 2 GR7 (FL=1), TEST
                        -------         -------------------------------------
```

```
                        THIS  IS  A  TEST  OF  THE  LDCLD  WITH  IMMEDIATE
                        ADDRESSING MODE

TEST 55                 SOURCE MODES, MODE 2 (FL=1), TEST
-------                 ---------------------------------
                        THIS IS A TEST OF THE LDCLD INSTRUCTION WITH MODE 2.

TEST 56                 LDCIF AND LDCLF TEST
-------                 --------------------

                        THIS  IS  A  TEST  OF  THE  LDCIF  AND  THE  LDCLF
                        INSTRUCTIONS.

TEST 57                 LDCID AND LDCLD TEST
-------                 --------------------

                        THIS IS A TEST OF LDCID AND LDCLD

TEST 60                 LDEXP TEST
-------                 ----------

                        THIS IS A TEST OF THE LDEXP  INST  A  SUBROUTINE  IS
                        USED  TO SET UP OPERANDS, EXECUTE THE LDEXP INST AND
                        CHECK THE RESULTS.

TEST 61                 DESTINATION MODES, MODE 1 (FL=0), TEST
-------                 --------------------------------------

                        THIS IS A TEST OF DESTINATION MODE 1 USING THE STFPS
                        INSTRUCTION

TEST 62                 DESTINATION MODES, MODE 2 (FL=0), TEST
-------                 --------------------------------------

                        THIS IS A TEST OF DESTINATION MODE 2 USING THE STFPS
                        INSTRUCTION

TEST 63                 DESTINATION MODES, MODE 4 (FL=0), TEST
-------                 --------------------------------------

                        THIS IS A TEST OF DESTINATION MODE 4 USING THE STFPS
                        INSTRUCTION

TEST 64                 DESTINATION MODES, MODE 3 (FL=0), TEST
-------                 --------------------------------------

                        THIS IS A TEST OF DESTINATION MODE 3 USING THE STFPS
                        INSTRUCTION

TEST 65                 DESTINATION MODES, MODE 5 (FL=0), TEST
-------                 --------------------------------------

                        THIS IS A TEST OF DESTINATION MODE 5 USING THE STFPS
                        INSTRUCTION

TEST 66                 DESTINATION MODES, MODE 6 (FL=0), TEST
```

```
                        -------         ---------------------------------------

                                THIS IS A TEST OF DESTINATION MODE 6 USING THE STFPS
                                INSTRUCTION

                        TEST 67         DESTINATION MODES, MODE 7 (FL=0), TEST
                        -------         ---------------------------------------
```

```
                        THIS IS A TEST OF DESTINATION MODE 7 USING THE STFPS
                        INSTRUCTION

                TEST 70         DESTINATION MODES, MODE 2 (FL=1), TEST
                -------         --------------------------------------

                        THIS IS A TEST OF DESTINATION  MODE  2  USING  STCOL
                        WITH REGISTER 0

                TEST 71         DESTINATION MODES, MODE 4 (FL=1), TEST
                -------         --------------------------------------

                        THIS IS A TEST OF DESTINATION  MODE  4  USING  STCDL
                        WITH REGISTER 0

                TEST 72         STCDI AND STCDL TEST
                -------         --------------------

                        THIS IS A TEST OF THE STCDI AND STCDL  INSTRUCTIONS.
                        NOTE  THAT  A  SUBROUTINE, STCSUB, IS USED TO SET UP
                        THE OPERANDS, EXECUTE THE STC INSTRUCTION AND  CHECK
                        THE RESULT.

                TEST 73         STCFL AND STCFI TEST
                -------         --------------------

                        THIS IS A TEST OF STCFL AND STCFI.  IT MAKES USE  OF
                        THE  SAME SUBROUTINE, STCSUB, WHICH WAS USED TO TEST
                        STCDL AND STCDI.

                TEST 74         STEXP TEST
                -------         ----------

                        THIS IS A TEST OF THE STEXP INSTRUCTION

                TEST 75         STST TEST
                -------         ---------

                        THIS IS A TEST OF THE STST  INSTRUCTION.   FIRST  AN
                        ILLEGAL  FPS  OP CODE (INSTRUCTION) IS USED TO ENTER
                        AN ERROR CONDITION IN THE FEC AND FEA.  THE STST  IS
                        EXECUTED AND THE FEC AND FEA ARE CHECKED

                TEST 76         D-SPACE NON-ACCESS TEST
                -------         -----------------------

                        THIS IS A TEST THAT ENABLES D-SPACE, BUT   MAKES  IT
                        NON-RESIDENT,  CAUSING  A  MEMORY  MANAGEMENT   TRAP
                        SHOULD IT BE  ACCESSED  DURING  AN  INSTRUCTION THAT
                        WILL NOT NORMALLY ACCESS D-SPACE.

                TEST 77         AUTO INCREMENT/DECREMENT TEST
                -------         -----------------------------

                        THIS IS A TEST THAT ENABLES D-SPACE, BUT   MAKES  IT
                        NON-RESIDENT IN THE AREA  OF  THE  TEST,  FORCING  A
                        MEMORY MANAGEMENT TRAP FOR EVERY FPP INSTRUCTION  IN
```

```
                        THE TEST. SR1 IS THEN EXAMINED FOR PROPER  CONTENTS.
                        SHOULD  THE FPP INSTRUCTION FAIL TO ABORT,  THE NEXT
                        INSTRUCTION IS  AN  IOT  TRAP, AND CALLS AN ERROR TO
                        ANOUNCE  THE FPP INSTRUCTION'S  FAILING  TO CAUSE AN
                        ABORT, NOT  ALLOWING PROPER EXAMINATION OF SR1.


                10.                     LISTING
                                        -------


                &

000443          MNUMBER=443
000003          PROGNUM=3


                        .LIST   ME
                        .NLIST  MD,MC,CND
```

```
1018 000000                             .ENABL  ABS
1025                                    .MCALL  .HEADER,.SWRHI,.EQUAT,.SETUP,.$CATCH,.$ACT11,.$CMTAG
1026                                    .MCALL  .$EOP,.$SCOPE,.$ERROR,.$SAVE,.$TYPE,.$TYPOCT
1027                                    .MCALL  .$TYPDEC,.$TRAP,.$POWER,.$APTHDR,.$APTBLS
1028                                    .MCALL  .$APTYPE,.$READ
1029                                    .MCALL  .EQUIV                          ;REMOVE FOR PDP-10
1030                            .TITLE  CKFPCA0 FP11F FLTG PNT PRT C
                                ;*COPYRIGHT (C) 1979
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
1031
1032
1033 000244                     FPVECT=244
1034 000250                     MMVECT=250
1035 177400                     $SWR=177400
1036 000200                     $SWRMSK=200
1037 000011                     TAB=11
1038 000015                     CRLF=15
1039
1040                            .SBTTL  BASIC DEFINITIONS
                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
104000                                  ERROR=EMT
000004                                  SCOPE=IOT
                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
177776                                  PSW=PS
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
000006                          SP=     %6              ;;STACK POINTER
000007                          PC=     %7              ;;PROGRAM COUNTER
                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
```

```
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7
                        ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
001000                          SW9=SW09
000400                          SW8=SW08
000200                          SW7=SW07
000100                          SW6=SW06
000040                          SW5=SW05
000020                          SW4=SW04
000010                          SW3=SW03
000004                          SW2=SW02
000002                          SW1=SW01
000001                          SW0=SW00
                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
001000                          BIT9=BIT09
000400                          BIT8=BIT08
000200                          BIT7=BIT07
000100                          BIT6=BIT06
000040                          BIT5=BIT05
000020                          BIT4=BIT04
000010                          BIT3=BIT03
000004                          BIT2=BIT02
```

```
                000002                  BIT1=BIT01
                000001                  BIT0=BIT00
                                ;*BASIC ''CPU'' TRAP VECTOR ADDRESSES
                000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                000014          TBITVEC=14              ;;''T'' BIT
                000014          TRTVEC= 14              ;;TRACE TRAP
                000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                000024          PWRVEC= 24              ;;POWER FAIL
                000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                000034          TRAPVEC=34              ;;''TRAP'' TRAP
                000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
                000064          TPVEC=  64              ;;TTY PRINTER VECTOR
                000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
   1041                         .SBTTL  FPP REGISTER DEFINITIONS
   1042         000000          AC0     =%0
   1043         000001          AC1     =%1
   1044         000002          AC2     =%2
   1045         000003          AC3     =%3
   1046         000004          AC4     =%4
   1047         000005          AC5     =%5
   1048         000006          AC6     =%6
   1049         000007          AC7     =%7
   1050         172300          KIPDR0  =172300
   1051         172302          KIPDR1  =172302
   1052         172304          KIPDR2  =172304
   1053         172306          KIPDR3  =172306
   1054         172310          KIPDR4  =172310
   1055         172316          KIPDR7  =172316
   1056         172340          KIPAR0  =172340
   1057         172342          KIPAR1  =172342
   1058         172344          KIPAR2  =172344
   1059         172346          KIPAR3  =172346
   1060         172350          KIPAR4  =172350
   1061         172356          KIPAR7  =172356
   1062         172320          KDPDR0  =172320
   1063         172322          KDPDR1  =172322
   1064         172324          KDPDR2  =172324
   1065         172326          KDPDR3  =172326
   1066         172330          KDPDR4  =172330
   1067         172336          KDPDR7  =172336
   1068         172360          KDPAR0  =172360
   1069         172362          KDPAR1  =172362
   1070         172364          KDPAR2  =172364
   1071         172366          KDPAR3  =172366
   1072         172370          KDPAR4  =172370
   1073         172376          KDPAR7  =172376
   1074         177572          MMR0    =177572
   1075         177574          SR1     =177574
   1076         177576          MMR2    =177576
   1077         172516          MMR3    =172516
   1078         117760          DATA    =117760
   1079         000020          IOTRAP  =000020
   1080
   1082
   1083                         .SBTTL  TRAP CATCHER
```

```
                000000                          .=0
                                        ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ''.+2,HALT''
                                        ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                        ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                000174                          .=174
000174  000000                         DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                         SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                        .SBTTL  STARTING ADDRESS(ES)
000200  000137  006106                          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
```

```
1084                                    .SBTTL  COMMON TAGS
                                        ;;**********************************************************************
                                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                        ;*USED IN THE PROGRAM.
            001100                              .=1100
001100                                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                                  .WORD   0
001102     000                          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000                          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000                          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000                          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000                          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001                          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000                          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000                          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000                          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                                  .WORD   0
001134     000                          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000                          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                                  .WORD   0
001140  177570                          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570                          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560                          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562                          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564                          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566                          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000                          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002                          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012                          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
001157     000                          $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
001160  000000                          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                        ;;WHICH  ($REG0) WAS OBTAINED
            000024                              .REPT   $CM3
001162  000000                          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000                          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000                          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000                          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000                          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000                          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000                          $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001200  000000                          $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001202  000000                          $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
001204  000000                          $REG11: .WORD   0               ;;CONTAINS (($REGAD)+22)
001206  000000                          $REG12: .WORD   0               ;;CONTAINS (($REGAD)+24)
001210  000000                          $REG13: .WORD   0               ;;CONTAINS (($REGAD)+26)
001212  000000                          $REG14: .WORD   0               ;;CONTAINS (($REGAD)+30)
001214  000000                          $REG15: .WORD   0               ;;CONTAINS (($REGAD)+32)
001216  000000                          $REG16: .WORD   0               ;;CONTAINS (($REGAD)+34)
001220  000000                          $REG17: .WORD   0               ;;CONTAINS (($REGAD)+36)
001222  000000                          $REG20: .WORD   0               ;;CONTAINS (($REGAD)+40)
001224  000000                          $REG21: .WORD   0               ;;CONTAINS (($REGAD)+42)
001226  000000                          $REG22: .WORD   0               ;;CONTAINS (($REGAD)+44)
```

```
001230  000000                  $REG23: .WORD   0               ;;CONTAINS (($REGAD)+46)
        000024                  .REPT   24
001232  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
001234  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
001236  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
001240  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
001242  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
001244  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
001246  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
001250  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
001252  000000                  $TMP10: .WORD   0               ;;USER DEFINED
001254  000000                  $TMP11: .WORD   0               ;;USER DEFINED
001256  000000                  $TMP12: .WORD   0               ;;USER DEFINED
001260  000000                  $TMP13: .WORD   0               ;;USER DEFINED
001262  000000                  $TMP14: .WORD   0               ;;USER DEFINED
001264  000000                  $TMP15: .WORD   0               ;;USER DEFINED
001266  000000                  $TMP16: .WORD   0               ;;USER DEFINED
001270  000000                  $TMP17: .WORD   0               ;;USER DEFINED
001272  000000                  $TMP20: .WORD   0               ;;USER DEFINED
001274  000000                  $TMP21: .WORD   0               ;;USER DEFINED
001276  000000                  $TMP22: .WORD   0               ;;USER DEFINED
001300  000000                  $TMP23: .WORD   0               ;;USER DEFINED
001302  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001304  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001306     207      377     377 $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001311     000
001312     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
001313     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001314     012      000         $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE
                                ;;*****************************************************************
                                .EVEN
001316                          $MAIL:                          ;;APT MAILBOX
001316  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001320  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001322  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001324  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001326  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001330  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001332  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001334  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001336                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
001336     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001337     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001340  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001342  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
001344  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
                                ;*                              BITS 15-11=CPU TYPE
                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                                      11/70=06,PDQ=07,Q=10
                                ;*                              BIT 10=REAL TIME CLOCK
                                ;*                              BIT  9=FLOATING POINT PROCESSOR
                                ;*                              BIT  8=MEMORY MANAGEMENT
001346     000                  $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
001347     000                  $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
```

```
                                ;*                                  900 NSEC CORE=001
                                ;*                                  300 NSEC BIPOLAR=002
                                ;*                                  500 NSEC MOS=003
001350  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
001352     000                  $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001353     000                  $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001354  000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001356     000                  $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001357     000                  $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001360  000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001362     000                  $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001363     000                  $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001364  000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001366  000000                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001370  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001372  000000                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001374  000000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
001376  000000                  $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
001400  000000                  $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
001402  000000                  $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
001404  000000                  $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
001406  000000                  $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
001410  000000                  $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
001412  000000                  $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
001414  000000                  $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
001416  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
001420  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
001422  000000                  $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
001424  000000                  $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
001426  000000                  $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
001430  000000                  $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
001432  000000                  $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
001434  000000                  $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
001436  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
001440  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
001442                          $ETEND:
```

```
                                        .SBTTL  ERROR POINTER TABLE
                                        ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                        ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                        ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                        ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                        ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
                                        ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                        ;*      DH              ;;POINTS TO THE DATA HEADER
                                        ;*      DT              ;;POINTS TO THE DATA
                                        ;*      DF              ;;POINTS TO THE DATA FORMAT
     001442                             $ERRTB:
1088          000443                            .REPT   MNUMBER
1090                                    ;ITEM 1
     001442   046614   073547   076012          .WORD   EM1,DH1,DT1,DF1
     001450   075354
                                        ;ITEM 2
     001452   046653   073622   076032          .WORD   EM2,DH2,DT2,DF2
     001460   075363
                                        ;ITEM 3
     001462   046706   073712   076054          .WORD   EM3,DH3,DT3,DF3
     001470   075363
                                        ;ITEM 4
     001472   046741   074002   076076          .WORD   EM4,DH4,DT4,DF4
     001500   075363
                                        ;ITEM 5
     001502   047001   074071   076120          .WORD   EM5,DH5,DT5,DF5
     001510   075373
                                        ;ITEM 6
     001512   047023   074071   076146          .WORD   EM6,DH6,DT6,DF6
     001520   075405
                                        ;ITEM 7
     001522   047127   074002   076076          .WORD   EM7,DH7,DT7,DF7
     001530   075363
                                        ;ITEM 10
     001532   047170   074071   076120          .WORD   EM10,DH10,DT10,DF10
     001540   075373
                                        ;ITEM 11
     001542   047213   074002   076076          .WORD   EM11,DH11,DT11,DF11
     001550   075363
                                        ;ITEM 12
     001552   047254   074071   076120          .WORD   EM12,DH12,DT12,DF12
     001560   075411
                                        ;ITEM 13
     001562   047277   074132   076146          .WORD   EM13,DH13,DT13,DF13
     001570   075405
                                        ;ITEM 14
     001572   047277   074132   076146          .WORD   EM14,DH14,DT14,DF14
     001600   075405
                                        ;ITEM 15
     001602   047333   074071   076120          .WORD   EM15,DH15,DT15,DF15
     001610   075411
                                        ;ITEM 16
     001612   047354   074172   076160          .WORD   EM16,DH16,DT16,DF16
     001620   075363
                                        ;ITEM 17
     001622   047403   074132   076146          .WORD   EM17,DH17,DT17,DF17
     001630   075405
```

```
                                          ;ITEM 20
001632  047441  074002  076160                  .WORD   EM20,DH20,DT20,DF20
001640  075363
                                          ;ITEM 21
001642  047502  074071  076120                  .WORD   EM21,DH21,DT21,DF21
001650  075411
                                          ;ITEM 22
001652  047502  074071  076120                  .WORD   EM22,DH22,DT22,DF22
001660  075411
                                          ;ITEM 23
001662  047525  074132  076146                  .WORD   EM23,DH23,DT23,DF23
001670  075405
                                          ;ITEM 24
001672  047564  074002  076160                  .WORD   EM24,DH24,DT24,DF24
001700  075363
                                          ;ITEM 25
001702  047626  074071  076120                  .WORD   EM25,DH25,DT25,DF25
001710  075411
                                          ;ITEM 26
001712  047652  074132  076146                  .WORD   EM26,DH26,DT26,DF26
001720  075405
                                          ;ITEM 27
001722  047711  074002  076160                  .WORD   EM27,DH27,DT27,DF27
001730  075363
                                          ;ITEM 30
001732  047753  074071  076120                  .WORD   EM30,DH30,DT30,DF30
001740  075411
                                          ;ITEM 31
001742  047777  074132  076146                  .WORD   EM31,DH31,DT31,DF31
001750  075405
                                          ;ITEM 32
001752  050035  074002  076160                  .WORD   EM32,DH32,DT32,DF32
001760  075363
                                          ;ITEM 33
001762  050076  074071  076120                  .WORD   EM33,DH33,DT33,DF33
001770  075411
                                          ;ITEM 34
001772  050121  074132  076146                  .WORD   EM34,DH34,DT34,DF34
002000  075405
                                          ;ITEM 35
002002  050160  074002  076160                  .WORD   EM35,DH35,DT35,DF35
002010  075363
                                          ;ITEM 36
002012  050222  074071  076120                  .WORD   EM36,DH36,DT36,DF36
002020  075411
                                          ;ITEM 37
002022  050246  074261  076202                  .WORD   EM37,DH37,DT37,DF37
002030  075423
                                          ;ITEM 40
002032  050272  074261  076202                  .WORD   EM40,DH40,DT40,DF40
002040  075423
                                          ;ITEM 41
002042  050320  074351  076246                  .WORD   EM41,DH41,DT41,DF41
002050  075444
                                          ;ITEM 42
002052  050346  074261  076202                  .WORD   EM42,DH42,DT42,DF42
002060  075423
```

```
                                        ;ITEM 43
002062  050425  074261  076202                .WORD   EM43,DH43,DT43,DF43
002070  075423
                                        ;ITEM 44
002072  050531  074261  076202                .WORD   EM44,DH44,DT44,DF44
002100  075423
                                        ;ITEM 45
002102  050631  074261  076202                .WORD   EM45,DH45,DT45,DF45
002110  075423
                                        ;ITEM 46
002112  050707  074261  076202                .WORD   EM46,DH46,DT46,DF46
002120  075423
                                        ;ITEM 47
002122  051013  074261  076202                .WORD   EM47,DH47,DT47,DF47
002130  075423
                                        ;ITEM 50
002132  051113  074261  076202                .WORD   EM50,DH50,DT50,DF50
002140  075423
                                        ;ITEM 51
002142  051227  074261  076202                .WORD   EM51,DH51,DT51,DF51
002150  075423
                                        ;ITEM 52
002152  051253  074261  076202                .WORD   EM52,DH52,DT52,DF52
002160  075423
                                        ;ITEM 53
002162  051277  074351  076246                .WORD   EM53,DH53,DT53,DF53
002170  075423
                                        ;ITEM 54
002172  051323  074261  076202                .WORD   EM54,DH54,DT54,DF54
002200  075423
                                        ;ITEM 55
002202  051402  074261  076202                .WORD   EM55,DH55,DT55,DF55
002210  075423
                                        ;ITEM 56
002212  051530  074261  076202                .WORD   EM56,DH56,DT56,DF56
002220  075423
                                        ;ITEM 57
002222  051632  074261  076202                .WORD   EM57,DH57,DT57,DF57
002230  075423
                                        ;ITEM 60
002232  051742  074261  076202                .WORD   EM60,DH60,DT60,DF60
002240  075423
                                        ;ITEM 61
002242  052052  074261  076202                .WORD   EM61,DH61,DT61,DF61
002250  075423
                                        ;ITEM 62
002252  052154  073622  076160                .WORD   EM62,DH62,DT62,DF62
002260  075363
                                        ;ITEM 63
002262  052260  073712  076160                .WORD   EM63,DH63,DT63,DF63
002270  075363
                                        ;ITEM 64
002272  052306  074071  076120                .WORD   EM64,DH64,DT64,DF64
002300  075373
                                        ;ITEM 65
002302  052362  073622  076160                .WORD   EM65,DH65,DT65,DF65
002310  075363
```

```
                                        ;ITEM 66
002312  052405  074002  076076                .WORD   EM66,DH66,DT66,DF66
002320  075363
                                        ;ITEM 67
002322  052444  073622  076076                .WORD   EM67,DH67,DT67,DF67
002330  075363
                                        ;ITEM 70
002332  052545  073712  076076                .WORD   EM70,DH70,DT70,DF70
002340  075363
                                        ;ITEM 71
002342  052636  074071  076312                .WORD   EM71,DH71,DT71,DF71
002350  075465
                                        ;ITEM 72
002352  052655  073622  076076                .WORD   EM72,DH72,DT72,DF72
002360  075363
                                        ;ITEM 73
002362  052736  074071  076346                .WORD   EM73,DH73,DT73,DF73
002370  075465
                                        ;ITEM 74
002372  052757  074002  076076                .WORD   EM74,DH74,DT74,DF74
002400  075363
                                        ;ITEM 75
002402  053001  073622  076032                .WORD   EM75,DH75,DT75,DF75
002410  075363
                                        ;ITEM 76
002412  053024  074132  076146                .WORD   EM76,DH76,DT76,DF76
002420  075405
                                        ;ITEM 77
002422  053065  074071  076346                .WORD   EM77,DH77,DT77,DF77
002430  075465
                                        ;ITEM 100
002432  053107  074002  076076                .WORD   EM100,DH100,DT100,DF100
002440  075363
                                        ;ITEM 101
002442  053132  073622  076032                .WORD   EM101,DH101,DT101,DF101
002450  075363
                                        ;ITEM 102
002452  053156  074132  076146                .WORD   EM102,DH102,DT102,DF102
002460  075405
                                        ;ITEM 103
002462  053217  074071  076346                .WORD   EM103,DH103,DT103,DF103
002470  075465
                                        ;ITEM 104
002472  053241  074002  076076                .WORD   EM104,DH104,DT104,DF104
002500  075363
                                        ;ITEM 105
002502  053264  073622  076032                .WORD   EM105,DH105,DT105,DF105
002510  075363
                                        ;ITEM 106
002512  053310  074132  076146                .WORD   EM106,DH106,DT106,DF106
002520  075405
                                        ;ITEM 107
002522  052676  074132  076146                .WORD   EM107,DH107,DT107,DF107
002530  075405
                                        ;ITEM 110
002532  053352  074071  076346                .WORD   EM110,DH110,DT110,DF110
002540  075465
```

```
                                        ;ITEM 111
002542  053375  074002  076076                .WORD   EM111,DH111,DT111,DF111
002550  075363
                                        ;ITEM 112
002552  053421  073622  076032                .WORD   EM112,DH112,DT112,DF112
002560  075363
                                        ;ITEM 113
002562  053446  074132  076146                .WORD   EM113,DH113,DT113,DF113
002570  075405
                                        ;ITEM 114
002572  053510  074071  076346                .WORD   EM114,DH114,DT114,DF114
002600  075465
                                        ;ITEM 115
002602  053533  074002  076076                .WORD   EM115,DH115,DT115,DF115
002610  075363
                                        ;ITEM 116
002612  053557  073622  076032                .WORD   EM116,DH116,DT116,DF116
002620  075363
                                        ;ITEM 117
002622  053604  074132  076146                .WORD   EM117,DH117,DT117,DF117
002630  075405
                                        ;ITEM 120
002632  053645  074071  076346                .WORD   EM120,DH120,DT120,DF120
002640  075465
                                        ;ITEM 121
002642  053667  074002  076076                .WORD   EM121,DH121,DT121,DF121
002650  075363
                                        ;ITEM 122
002652  053712  073622  076032                .WORD   EM122,DH122,DT122,DF122
002660  075363
                                        ;ITEM 123
002662  053736  074132  076146                .WORD   EM123,DH123,DT123,DF123
002670  075405
                                        ;ITEM 124
002672  054000  074071  076346                .WORD   EM124,DH124,DT124,DF124
002700  075465
                                        ;ITEM 125
002702  054023  074002  076076                .WORD   EM125,DH125,DT125,DF125
002710  075363
                                        ;ITEM 126
002712  054047  073622  076032                .WORD   EM126,DH126,DT126,DF126
002720  075363
                                        ;ITEM 127
002722  054074  074132  076146                .WORD   EM127,DH127,DT127,DF127
002730  075405
                                        ;ITEM 130
002732  054136  074071  076346                .WORD   EM130,DH130,DT130,DF130
002740  075465
                                        ;ITEM 131
002742  054161  073622  076032                .WORD   EM131,DH131,DT131,DF131
002750  075363
                                        ;ITEM 132
002752  054206  074132  076146                .WORD   EM132,DH132,DT132,DF132
002760  075405
                                        ;ITEM 133
002762  054251  074071  076346                .WORD   EM133,DH133,DT133,DF133
002770  075465
```

```
                                        ;ITEM 134
002772  054275  073622  076032                .WORD   EM134,DH134,DT134,DF134
003000  075363
                                        ;ITEM 135
003002  054323  074071  076120                .WORD   EM135,DH135,DT135,DF135
003010  075411
                                        ;ITEM 136
003012  054376  074071  076120                .WORD   EM136,DH136,DT136,DF136
003020  075411
                                        ;ITEM 137
003022  054415  073622  076160                .WORD   EM137,DH137,DT137,DF137
003030  075363
                                        ;ITEM 140
003032  054436  074071  076120                .WORD   EM140,DH140,DT140,DF140
003040  075411
                                        ;ITEM 141
003042  054457  074002  076076                .WORD   EM141,DH141,DT141,DF141
003050  075363
                                        ;ITEM 142
003052  054526  073622  076076                .WORD   EM142,DH142,DT142,DF142
003060  075363
                                        ;ITEM 143
003062  054551  074071  076120                .WORD   EM143,DH143,DT143,DF143
003070  075411
                                        ;ITEM 144
003072  054573  074002  076076                .WORD   EM144,DH144,DT144,DF144
003100  075363
                                        ;ITEM 145
003102  054643  073622  076076                .WORD   EM145,DH145,DT145,DF145
003110  075363
                                        ;ITEM 146
003112  054667  074071  076120                .WORD   EM146,DH146,DT146,DF146
003120  075411
                                        ;ITEM 147
003122  054711  074002  076076                .WORD   EM147,DH147,DT147,DF147
003130  075363
                                        ;ITEM 150
003132  054761  073622  076076                .WORD   EM150,DH150,DT150,DF150
003140  075363
                                        ;ITEM 151
003142  055005  074071  076120                .WORD   EM151,DH151,DT151,DF151
003150  075411
                                        ;ITEM 152
003152  055030  074002  076076                .WORD   EM152,DH152,DT152,DF152
003160  075363
                                        ;ITEM 153
003162  055101  073622  076076                .WORD   EM153,DH153,DT153,DF153
003170  075363
                                        ;ITEM 154
003172  055126  074071  076120                .WORD   EM154,DH154,DT154,DF154
003200  075411
                                        ;ITEM 155
003202  055151  074002  076076                .WORD   EM155,DH155,DT155,DF155
003210  075363
                                        ;ITEM 156
003212  055222  073622  076076                .WORD   EM156,DH156,DT156,DF156
003220  075363
```

```
                                        ;ITEM 157
003222  055247  074071  076120                .WORD   EM157,DH157,DT157,DF157
003230  075411
                                        ;ITEM 160
003232  055271  074002  076076                .WORD   EM160,DH160,DT160,DF160
003240  075363
                                        ;ITEM 161
003242  055363  073622  076076                .WORD   EM161,DH161,DT161,DF161
003250  075363
                                        ;ITEM 162
003252  055407  074071  076120                .WORD   EM162,DH162,DT162,DF162
003260  075411
                                        ;ITEM 163
003262  055432  073622  076076                .WORD   EM163,DH163,DT163,DF163
003270  075363
                                        ;ITEM 164
003272  055457  074172  076076                .WORD   EM164,DH164,DT164,DF164
003300  075363
                                        ;ITEM 165
003302  056255  074261  076202                .WORD   EM165,DH165,DT165,DF165
003310  075423
                                        ;ITEM 166
003312  056276  074261  076202                .WORD   EM166,DH166,DT166,DF166
003320  075423
                                        ;ITEM 167
003322  056317  074261  076202                .WORD   EM167,DH167,DT167,DF167
003330  075423
                                        ;ITEM 170
003332  056340  074261  076202                .WORD   EM170,DH170,DT170,DF170
003340  075423
                                        ;ITEM 171
003342  056363  074261  076202                .WORD   EM171,DH171,DT171,DF171
003350  075423
                                        ;ITEM 172
003352  056406  074261  076202                .WORD   EM172,DH172,DT172,DF172
003360  075423
                                        ;ITEM 173
003362  056431  074351  076246                .WORD   EM173,DH173,DT173,DF173
003370  075444
                                        ;ITEM 174
003372  056454  074351  076246                .WORD   EM174,DH174,DT174,DF174
003400  075444
                                        ;ITEM 175
003402  056477  074351  076246                .WORD   EM175,DH175,DT175,DF175
003410  075444
                                        ;ITEM 176
003412  052570  073622  076076                .WORD   EM176,DH176,DT176,DF176
003420  075363
                                        ;ITEM 177
003422  052613  073712  076076                .WORD   EM177,DH177,DT177,DF177
003430  075363
                                        ;ITEM 200
003432  056522  074261  076202                .WORD   EM200,DH200,DT200,DF200
003440  075423
                                        ;ITEM 201
003442  056577  074261  076202                .WORD   EM201,DH201,DT201,DF201
003450  075423
```

```
                                        ;ITEM 202
003452  056700  074261  076202                .WORD   EM202,DH202,DT202,DF202
003460  075423
                                        ;ITEM 203
003462  057001  074261  076202                .WORD   EM203,DH203,DT203,DF203
003470  075423
                                        ;ITEM 204
003472  057161  074261  076202                .WORD   EM204,DH204,DT204,DF204
003500  075423
                                        ;ITEM 205
003502  057236  074261  076202                .WORD   EM205,DH205,DT205,DF205
003510  075423
                                        ;ITEM 206
003512  057335  074261  076202                .WORD   EM206,DH206,DT206,DF206
003520  075423
                                        ;ITEM 207
003522  057436  074261  076202                .WORD   EM207,DH207,DT207,DF207
003530  075423
                                        ;ITEM 210
003532  057535  074261  076202                .WORD   EM210,DH210,DT210,DF210
003540  075423
                                        ;ITEM 211
003542  057634  074261  076202                .WORD   EM211,DH211,DT211,DF211
003550  075423
                                        ;ITEM 212
003552  057742  074261  076202                .WORD   EM212,DH212,DT212,DF212
003560  075423
                                        ;ITEM 213
003562  060043  074261  076202                .WORD   EM213,DH213,DT213,DF213
003570  075423
                                        ;ITEM 214
003572  060170  074261  076202                .WORD   EM214,DH214,DT214,DF214
003600  075423
                                        ;ITEM 215
003602  055533  074172  076076                .WORD   EM215,DH215,DT215,DF215
003610  075363
                                        ;ITEM 216
003612  055664  074071  076120                .WORD   EM216,DH216,DT216,DF216
003620  075411
                                        ;ITEM 217
003622  055706  074002  076076                .WORD   EM217,DH217,DT217,DF217
003630  075363
                                        ;ITEM 220
003632  055756  073622  076076                .WORD   EM220,DH220,DT220,DF220
003640  075363
                                        ;ITEM 221
003642  056002  074172  076076                .WORD   EM221,DH221,DT221,DF221
003650  075363
                                        ;ITEM 222
003652  056134  074071  076120                .WORD   EM222,DH222,DT222,DF222
003660  075411
                                        ;ITEM 223
003662  056157  074002  076076                .WORD   EM223,DH223,DT223,DF223
003670  075363
                                        ;ITEM 224
003672  056230  073622  076076                .WORD   EM224,DH224,DT224,DF224
003700  075363
```

```
                                        ;ITEM 225
003702  060315  074002  076076                .WORD   EM225,DH225,DT225,DF225
003710  075502
                                        ;ITEM 226
003712  060340  073622  076076                .WORD   EM226,DH226,DT226,DF226
003720  075502
                                        ;ITEM 227
003722  060364  074446  076146                .WORD   EM227,DH227,DT227,DF227
003730  075512
                                        ;ITEM 230
003732  060414  074002  076076                .WORD   EM230,DH230,DT230,DF230
003740  075502
                                        ;ITEM 231
003742  060440  073622  076076                .WORD   EM231,DH231,DT231,DF231
003750  075502
                                        ;ITEM 232
003752  060465  074446  076146                .WORD   EM232,DH232,DT232,DF232
003760  075512
                                        ;ITEM 233
003762  060516  074002  076076                .WORD   EM233,DH233,DT233,DF233
003770  075502
                                        ;ITEM 234
003772  060542  073622  076076                .WORD   EM234,DH234,DT234,DF234
004000  075502
                                        ;ITEM 235
004002  060567  074446  076146                .WORD   EM235,DH235,DT235,DF235
004010  075512
                                        ;ITEM 236
004012  060620  074002  076076                .WORD   EM236,DH236,DT236,DF236
004020  075502
                                        ;ITEM 237
004022  060645  073622  076076                .WORD   EM237,DH237,DT237,DF237
004030  075502
                                        ;ITEM 240
004032  060673  074446  076146                .WORD   EM240,DH240,DT240,DF240
004040  075512
                                        ;ITEM 241
004042  060725  074002  076076                .WORD   EM241,DH241,DT241,DF241
004050  075502
                                        ;ITEM 242
004052  060752  073622  076076                .WORD   EM242,DH242,DT242,DF242
004060  075502
                                        ;ITEM 243
004062  061000  074446  076146                .WORD   EM243,DH243,DT243,DF243
004070  075512
                                        ;ITEM 244
004072  061032  074002  076076                .WORD   EM244,DH244,DT244,DF244
004100  075502
                                        ;ITEM 245
004102  061056  073622  076076                .WORD   EM245,DH245,DT245,DF245
004110  075502
                                        ;ITEM 246
004112  061103  074172  076076                .WORD   EM246,DH246,DT246,DF246
004120  075502
                                        ;ITEM 247
004122  061134  074446  076146                .WORD   EM247,DH247,DT247,DF247
004130  075512
```

```
                                        ;ITEM 250
004132  061165  074002  076076                .WORD   EM250,DH250,DT250,DF250
004140  075502
                                        ;ITEM 251
004142  061212  073622  076076                .WORD   EM251,DH251,DT251,DF251
004150  075502
                                        ;ITEM 252
004152  061240  074172  076076                .WORD   EM252,DH252,DT252,DF252
004160  075502
                                        ;ITEM 253
004162  061272  074446  076146                .WORD   EM253,DH253,DT253,DF253
004170  075512
                                        ;ITEM 254
004172  061324  074172  076076                .WORD   EM254,DH254,DT254,DF254
004200  075502
                                        ;ITEM 255
004202  061360  074446  076146                .WORD   EM255,DH255,DT255,DF255
004210  075512
                                        ;ITEM 256
004212  061414  074002  076076                .WORD   EM256,DH256,DT256,DF256
004220  075502
                                        ;ITEM 257
004222  061442  073622  076076                .WORD   EM257,DH257,DT257,DF257
004230  075502
                                        ;ITEM 260
004232  061471  074261  076202                .WORD   EM260,DH260,DT260,DF260
004240  075516
                                        ;ITEM 261
004242  061526  074261  076202                .WORD   EM261,DH261,DT261,DF261
004250  075516
                                        ;ITEM 262
004252  061565  074261  076202                .WORD   EM262,DH262,DT262,DF262
004260  075516
                                        ;ITEM 263
004262  061665  074261  076202                .WORD   EM263,DH263,DT263,DF263
004270  075516
                                        ;ITEM 264
004272  061713  074261  076202                .WORD   EM264,DH264,DT264,DF264
004300  075516
                                        ;ITEM 265
004302  062010  074261  076202                .WORD   EM265,DH265,DT265,DF265
004310  075516
                                        ;ITEM 266
004312  062101  074261  076202                .WORD   EM266,DH266,DT266,DF266
004320  075516
                                        ;ITEM 267
004322  062214  074261  076202                .WORD   EM267,DH267,DT267,DF267
004330  075516
                                        ;ITEM 270
004332  062311  074261  076202                .WORD   EM270,DH270,DT270,DF270
004340  075516
                                        ;ITEM 271
004342  062352  074261  076202                .WORD   EM271,DH271,DT271,DF271
004350  075516
                                        ;ITEM 272
004352  062420  074261  076202                .WORD   EM272,DH272,DT272,DF272
004360  075516
```

```
                                        ;ITEM 273
004362  062511  074261  076202                .WORD   EM273,DH273,DT273,DF273
004370  075537
                                        ;ITEM 274
004372  062546  074261  076202                .WORD   EM274,DH274,DT274,DF274
004400  075537
                                        ;ITEM 275
004402  062605  074261  076202                .WORD   EM275,DH275,DT275,DF275
004410  075537
                                        ;ITEM 276
004412  062705  074261  076202                .WORD   EM276,DH276,DT276,DF276
004420  075537
                                        ;ITEM 277
004422  063002  074261  076202                .WORD   EM277,DH277,DT277,DF277
004430  075537
                                        ;ITEM 300
004432  063056  074261  076202                .WORD   EM300,DH300,DT300,DF300
004440  075537
                                        ;ITEM 301
004442  063153  074261  076402                .WORD   EM301,DH301,DT301,DF301
004450  075560
                                        ;ITEM 302
004452  063177  074261  076402                .WORD   EM302,DH302,DT302,DF302
004460  075560
                                        ;ITEM 303
004462  063225  074351  076454                .WORD   EM303,DH303,DT303,DF303
004470  075604
                                        ;ITEM 304
004472  063253  074261  076402                .WORD   EM304,DH304,DT304,DF304
004500  075560
                                        ;ITEM 305
004502  063342  074261  076402                .WORD   EM305,DH305,DT305,DF305
004510  075560
                                        ;ITEM 306
004512  063445  074261  076402                .WORD   EM306,DH306,DT306,DF306
004520  075560
                                        ;ITEM 307
004522  063632  074261  076402                .WORD   EM307,DH307,DT307,DF307
004530  075560
                                        ;ITEM 310
004532  063734  074261  076402                .WORD   EM310,DH310,DT310,DF310
004540  075560
                                        ;ITEM 311
004542  064037  074261  076402                .WORD   EM311,DH311,DT311,DF311
004550  075560
                                        ;ITEM 312
004552  064140  074261  076402                .WORD   EM312,DH312,DT312,DF312
004560  075560
                                        ;ITEM 313
004562  064242  074261  076402                .WORD   EM313,DH313,DT313,DF313
004570  075560
                                        ;ITEM 314
004572  064343  074261  076402                .WORD   EM314,DH314,DT314,DF314
004600  075560
                                        ;ITEM 315
004602  064444  074261  076402                .WORD   EM315,DH315,DT315,DF315
004610  075560
```

```
                                        ;ITEM 316
004612  064545  074261  076402                .WORD   EM316,DH316,DT316,DF316
004620  075560
                                        ;ITEM 317
004622  064646  074261  076402                .WORD   EM317,DH317,DT317,DF317
004630  075560
                                        ;ITEM 320
004632  064747  074261  076402                .WORD   EM320,DH320,DT320,DF320
004640  075560
                                        ;ITEM 321
004642  065050  074261  076402                .WORD   EM321,DH321,DT321,DF321
004650  075560
                                        ;ITEM 322
004652  065151  074261  076526                .WORD   EM322,DH322,DT322,DF322
004660  075630
                                        ;ITEM 323
004662  065206  074261  076526                .WORD   EM323,DH323,DT323,DF323
004670  075630
                                        ;ITEM 324
004672  065245  074351  076572                .WORD   EM324,DH324,DT324,DF324
004700  075651
                                        ;ITEM 325
004702  065304  074261  076526                .WORD   EM325,DH325,DT325,DF325
004710  075630
                                        ;ITEM 326
004712  065304  074261  076526                .WORD   EM326,DH326,DT326,DF326
004720  075630
                                        ;ITEM 327
004722  065445  074261  076526                .WORD   EM327,DH327,DT327,DF327
004730  075630
                                        ;ITEM 330
004732  065547  074261  076526                .WORD   EM330,DH330,DT330,DF330
004740  075630
                                        ;ITEM 331
004742  065652  074261  076526                .WORD   EM331,DH331,DT331,DF331
004750  075630
                                        ;ITEM 332
004752  067126  074261  076526                .WORD   EM332,DH332,DT332,DF332
004760  075630
                                        ;ITEM 333
004762  065206  074261  076526                .WORD   EM333,DH333,DT333,DF333
004770  075630
                                        ;ITEM 334
004772  065755  074261  076526                .WORD   EM334,DH334,DT334,DF334
005000  075630
                                        ;ITEM 335
005002  066051  074261  076526                .WORD   EM335,DH335,DT335,DF335
005010  075630
                                        ;ITEM 336
005012  066153  074261  076526                .WORD   EM336,DH336,DT336,DF336
005020  075630
                                        ;ITEM 337
005022  066227  074261  076526                .WORD   EM337,DH337,DT337,DF337
005030  075630
                                        ;ITEM 340
005032  066331  074261  076526                .WORD   EM340,DH340,DT340,DF340
005040  075630
```

```
                                        ;ITEM 341
005042  066433  074261  076526                .WORD   EM341,DH341,DT341,DF341
005050  075630
                                        ;ITEM 342
005052  066537  074261  076526                .WORD   EM342,DH342,DT342,DF342
005060  075630
                                        ;ITEM 343
005062  066641  074261  076526                .WORD   EM343,DH343,DT343,DF343
005070  075630
                                        ;ITEM 344
005072  066743  074261  076526                .WORD   EM344,DH344,DT344,DF344
005100  075630
                                        ;ITEM 345
005102  067220  074261  076526                .WORD   EM345,DH345,DT345,DF345
005110  075630
                                        ;ITEM 346
005112  067320  074261  076526                .WORD   EM346,DH346,DT346,DF346
005120  075630
                                        ;ITEM 347
005122  067416  074261  076526                .WORD   EM347,DH347,DT347,DF347
005130  075672
                                        ;ITEM 350
005132  067442  074261  076526                .WORD   EM350,DH350,DT350,DF350
005140  075672
                                        ;ITEM 351
005142  067470  074132  076146                .WORD   EM351,DH351,DT351,DF351
005150  075512
                                        ;ITEM 352
005152  067574  074261  076526                .WORD   EM352,DH352,DT352,DF352
005160  075672
                                        ;ITEM 353
005162  067700  074261  076526                .WORD   EM353,DH353,DT353,DF353
005170  075672
                                        ;ITEM 354
005172  070004  074261  076526                .WORD   EM354,DH354,DT354,DF354
005200  075672
                                        ;ITEM 355
005202  070110  074261  076526                .WORD   EM355,DH355,DT355,DF355
005210  075672
                                        ;ITEM 356
005212  070214  074002  076032                .WORD   EM356,DH356,DT356,DF356
005220  075502
                                        ;ITEM 357
005222  070312  074506  076054                .WORD   EM357,DH357,DT357,DF357
005230  075502
                                        ;ITEM 360
005232  070410  074132  076146                .WORD   EM360,DH360,DT360,DF360
005240  075512
                                        ;ITEM 361
005242  072640  073622  076402                .WORD   EM361,DH361,DT361,DF361
005250  075502
                                        ;ITEM 362
005252  072663  074576  076636                .WORD   EM362,DH362,DT362,DF362
005260  075713
                                        ;ITEM 363
005262  072773  074641  076654                .WORD   EM363,DH363,DT363,DF363
005270  075721
```

```
                                        ;ITEM 364
005272  073041  074716  076674                .WORD   EM364,DH364,DT364,DF364
005300  075405
                                        ;ITEM 365
005302  073141  075003  076146                .WORD   EM365,DH365,DT365,DF365
005310  075405
                                        ;ITEM 366
005312  073224  075041  076706                .WORD   EM366,DH366,DT366,DF366
005320  075730
                                        ;ITEM 367
005322  073307  075114  076754                .WORD   EM367,DH367,DT367,DF367
005330  075752
                                        ;ITEM 370
005332  073372  075202  076776                .WORD   EM370,DH370,DT370,DF370
005340  075762
                                        ;ITEM 371
005342  000000  000000  000000                .WORD   EM371,DH371,DT371,DF371
005350  000000
                                        ;ITEM 372
005352  000000  000000  000000                .WORD   EM372,DH372,DT372,DF372
005360  000000
                                        ;ITEM 373
005362  000000  000000  000000                .WORD   EM373,DH373,DT373,DF373
005370  000000
                                        ;ITEM 374
005372  000000  000000  000000                .WORD   EM374,DH374,DT374,DF374
005400  000000
                                        ;ITEM 375
005402  000000  000000  000000                .WORD   EM375,DH375,DT375,DF375
005410  000000
                                        ;ITEM 376
005412  000000  000000  000000                .WORD   EM376,DH376,DT376,DF376
005420  000000
                                        ;ITEM 377
005422  000000  000000  000000                .WORD   EM377,DH377,DT377,DF377
005430  000000
                                        ;ITEM 400
005432  000000  000000  000000                .WORD   EM400,DH400,DT400,DF400
005440  000000
                                        ;ITEM 401
005442  070503  074002  076076                .WORD   EM401,DH401,DT401,DF401
005450  075502
                                        ;ITEM 402
005452  070526  073622  076076                .WORD   EM402,DH402,DT402,DF402
005460  075502
                                        ;ITEM 403
005462  070550  074132  076146                .WORD   EM403,DH403,DT403,DF403
005470  075512
                                        ;ITEM 404
005472  070702  074446  076146                .WORD   EM404,DH404,DT404,DF404
005500  075512
                                        ;ITEM 405
005502  070732  074002  076076                .WORD   EM405,DH405,DT405,DF405
005510  075502
                                        ;ITEM 406
005512  070756  073622  076076                .WORD   EM406,DH406,DT406,DF406
005520  075502
```

```
                                        ;ITEM 407
005522  071001  074132  076146                .WORD   EM407,DH407,DT407,DF407
005530  075512
                                        ;ITEM 410
005532  071134  074446  076146                .WORD   EM410,DH410,DT410,DF410
005540  075512
                                        ;ITEM 411
005542  071165  074002  076076                .WORD   EM411,DH411,DT411,DF411
005550  075502
                                        ;ITEM 412
005552  071211  073622  076076                .WORD   EM412,DH412,DT412,DF412
005560  075502
                                        ;ITEM 413
005562  071234  074132  076146                .WORD   EM413,DH413,DT413,DF413
005570  075512
                                        ;ITEM 414
005572  071367  074446  076146                .WORD   EM414,DH414,DT414,DF414
005600  075512
                                        ;ITEM 415
005602  071420  074002  076076                .WORD   EM415,DH415,DT415,DF415
005610  075502
                                        ;ITEM 416
005612  071445  073622  076076                .WORD   EM416,DH416,DT416,DF416
005620  075502
                                        ;ITEM 417
005622  071471  074132  076146                .WORD   EM417,DH417,DT417,DF417
005630  075512
                                        ;ITEM 420
005632  071537  074446  076146                .WORD   EM420,DH420,DT420,DF420
005640  075512
                                        ;ITEM 421
005642  071571  074002  076076                .WORD   EM421,DH421,DT421,DF421
005650  075502
                                        ;ITEM 422
005652  071616  073622  076076                .WORD   EM422,DH422,DT422,DF422
005660  075502
                                        ;ITEM 423
005662  071642  074132  076146                .WORD   EM423,DH423,DT423,DF423
005670  075512
                                        ;ITEM 424
005672  071710  074446  076146                .WORD   EM424,DH424,DT424,DF424
005700  075512
                                        ;ITEM 425
005702  071742  074002  076076                .WORD   EM425,DH425,DT425,DF425
005710  075502
                                        ;ITEM 426
005712  071766  073622  076076                .WORD   EM426,DH426,DT426,DF426
005720  075502
                                        ;ITEM 427
005722  072011  074132  076146                .WORD   EM427,DH427,DT427,DF427
005730  075512
                                        ;ITEM 430
005732  072144  074446  076146                .WORD   EM430,DH430,DT430,DF430
005740  075512
                                        ;ITEM 431
005742  072175  074132  076146                .WORD   EM431,DH431,DT431,DF431
005750  075512
```

```
                                        ;ITEM 432
005752  072250  074002  076076                .WORD   EM432,DH432,DT432,DF432
005760  075502
                                        ;ITEM 433
005762  072275  073622  076076                .WORD   EM433,DH433,DT433,DF433
005770  075502
                                        ;ITEM 434
005772  072321  074132  076146                .WORD   EM434,DH434,DT434,DF434
006000  075512
                                        ;ITEM 435
006002  072455  074446  076146                .WORD   EM435,DH435,DT435,DF435
006010  075512
                                        ;ITEM 436
006012  072507  074132  076146                .WORD   EM436,DH436,DT436,DF436
006020  075512
                                        ;ITEM 437
006022  072564  074002  076076                .WORD   EM437,DH437,DT437,DF437
006030  075502
                                        ;ITEM 440
006032  072612  074002  076076                .WORD   EM440,DH440,DT440,DF440
006040  075502
                                        ;ITEM 441
006042  073426  075245  077042                .WORD   EM441,DH441,DT441,DF441
006050  076003
                                        ;ITEM 442
006052  073462  075313  077060                .WORD   EM442,DH442,DT442,DF442
006060  076003
                                        ;ITEM 443
006062  073514  075313  077060                .WORD   EM443,DH443,DT443,DF443
006070  076003
1091
1092
1093                                    .SBTTL  ACT11 HOOKS
                                        ;;******************************************************************
                                        ;HOOKS REQUIRED BY ACT11
        006072                                  $SVPC=.                 ;SAVE PC
        000046                                  .=46
000046  042736                                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
        000052                                  .=52
000052  000000                                  .WORD   0               ;;2)SET LOC.52 TO ZERO
        006072                                  .=$SVPC                 ;; RESTORE PC
1094                                    .SBTTL  APT PARAMETER BLOCK
                                        ;;******************************************************************
                                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                        ;;******************************************************************
        006072                                  .$X=.   ;;SAVE CURRENT LOCATION
        000024                                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                                  200     ;;FOR APT START UP
        000044                                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  006072                                  $APTHDR ;;POINT TO APT HEADER BLOCK
        006072                                  .=.$X   ;;RESET LOCATION COUNTER
                                        ;;******************************************************************
                                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                        ;INTERFACE SPEC.
006072                                  $APTHD:
006072  000000                          $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
006074  001316                          $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
```

```
006076  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
006100  000040                  $PASTM: .WORD   40      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
006102  000000                  $UNITM: .WORD   0       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
006104  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1095
1096
1097 006106                     START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
006106  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
006112  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
006114  022706  001140                  CMP     #SWR,R6 ;;DONE?
006120  001374                          BNE     .-6             ;;LOOP BACK IF NO
006122  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
006126  012737  043016  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
006134  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
006142  012737  043276  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
006150  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
006156  012737  045314  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
006164  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
006172  012737  045400  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
006200  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
006206  016767  034346  034336          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
006214  005067  173062                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
006220  005067  173060                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
006224  112767  000001  172663          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
                                ;;INITIALIZE THE ''T-BIT'' TRAP VECTOR. THEN LOAD LOCATION ''$RTRN'', IN
                                ;;THE ''END-OF-PASS'' ($EOP) ROUTINE, WITH A ''RTI'' OR ''RTT''.
006232  012737  043002  000014          MOV     #$RTRN,@#TBITVEC ;;SET ''T'' BIT VECTOR TO $RTRN
006240  012737  000340  000016          MOV     #340,@#TBITVEC+2 ;;LEVEL 7
006246  012767  000002  034526          MOV     #RTI,$RTRN      ;;SET $RTRN TO A RTI
006254  012737  006302  000010          MOV     #65$,@#RESVEC   ;;TRY TO DO A RTT
006262  005046                          CLR     -(SP)           ;;DUMMY PS
006264  012746  006272                  MOV     #64$,-(SP)      ;;AND PC
006270  000006                          RTT                     ;;TRY THE RTT
006272  012767  000006  034502  64$:    MOV     #RTT,$RTRN      ;;RTT IS LEGAL--SET $RTRN TO A RTT
006300  000402                          BR      66$
006302  062706  000010          65$:    ADD     #10,SP          ;;RTT ILLEGAL--CLEAN OFF THE STACK
006306  012737  000012  000010  66$:    MOV     #RESVEC+2,@#RESVEC ;;RESTORE TRAP CATCHER
006314  005067  034470                  CLR     $TBIT           ;;CLEAR ''T'' BIT SWITCH
006320  012767  006320  172560          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
006326  012767  006326  172554          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A ''-1'', SETUP FOR A SOFTWARE SWITCH REGISTER.
006334  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
006340  012737  006374  000004          MOV     #67$,@#ERRVEC   ;;SET UP ERROR VECTOR
006346  012767  177570  172564          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
006354  012767  177570  172560          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
006362  022777  177777  172550          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
006370  001012                          BNE     69$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
006372  000403                          BR      68$             ;;BRANCH IF NO TIMEOUT
006374  012716  006402          67$:    MOV     #68$,(SP)       ;;SET UP FOR TRAP RETURN
006400  000002                          RTI
006402  012767  000176  172530  68$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
006410  012767  000174  172524          MOV     #DISPREG,DISPLAY
```

```
006416  012637  000004          69$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
006422  005067  172676                  CLR     $PASS           ;;CLEAR PASS COUNT
006426  132767  000200  172703        BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
006434  001403                        BEQ     70$             ;;YES,USE NON-APT SWITCH
006436  012767  001340  172474        MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
006444                          70$:
1098                            .SBTTL  TYPE PROGRAM NAME
                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
006444  005227  177777                  INC     #-1             ;;FIRST TIME?
006450  001047                        BNE     71$             ;;BRANCH IF NO
006452  022737  042736  000042        CMP     #$ENDAD,@#42    ;;ACT-11?
006460  001443                        BEQ     71$             ;;BRANCH IF YES
006462  104401  006530                TYPE    ,72$            ;;TYPE ASCIZ STRING
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
006466  005737  000042                TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
006472  001012                        BNE     73$             ;;BRANCH IF YES
006474  126727  172636  000001        CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
006502  001406                        BEQ     73$             ;;BRANCH IF YES
006504  026727  172430  000176        CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
006512  001005                        BNE     74$             ;;BRANCH IF NO
006514  104405                        GTSWR                   ;;GET SOFT-SWR SETTINGS
006516  000403                        BR      74$
006520  112767  000001  172406  73$:   MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
006526                          74$:
006526  000420                        BR      71$             ;;GET OVER THE ASCIZ
                                ;;72$:  .ASCIZ  <CRLF>*CKFPCA0 FP11F FLTG PNT PRT C*<CRLF>
006570                          71$:
1099
1100 006570                     LOOP:
1101
1102
1103
1104
1110
1111                            ;;*****************************************************************
                                ;*TEST 1        STF WITH ILLEGAL ACCUMULATOR TEST
                                ;*
                                ;*THIS IS A TEST OF THE ST INSTRUCTION USING ILLEGAL ACCUMULATOR 7, MODE 0.
                                ;*
                                ;;*****************************************************************
006570  000004                  TST1:   SCOPE
1112
1113 006572                     0001:
006572  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1114 006574  005000                     CLR     R0              ;SET THE FPS.
1115 006576  170100                     LDFPS   R0
1116
1117 006600  012737  006636  000244      MOV     #000T,@#FPVECT  ;SET UP FOR FP TRAPS.
1118 006606  012737  006614  001236      MOV     #1$,@#$TMP2
1119
1120 006614  174007              1$:     STF     AC0,AC7         ;THIS TEST INSTRUCTION SHOULD
1121                                                             ;CAUSE A TRAP.
1122
1123                            ;REPORT FAILURE OF USE OF ILLEGAL ACCUMULATOR 7 TO CAUSE AN FPP TRAP.
1124 006616                     0002:
1125 006616  170200                     STFPS   R0              ;GET FPS.
1126 006620  010037  001240              MOV     R0,@#$TMP3
```

```
1127 006624  170300                    STST    R0              ;GET FEC.
1128 006626  010037  001242            MOV     R0,@#$TMP4
1129 006632  104001            3$:     ERROR   +1              ;STF WITH ILLEGAL ACCUMULATOR, MODE
1130                                                           ;0, DIDN'T TRAP. ST 765 TO ST 537.
1131 006634  000434                    BR      OODONE
1132
1133                           ;TRAP TO OOOT, HERE, WHEN THE EXPECTED ERROR OCCURS.
1134 006636  011600            OOOT:   MOV     (SP),R0         ;MAKE SURE THE ERROR OCCURRED
1135 006640  022700  006616            CMP     #0002,R0        ;AT THE CORRECT ADDRESS.
1136 006644  001402                    BEQ     0003            ;BRANCH IF TRAP ADDRESS CORRECT.
1137 006646  000137  046214            JMP     @#FPSPUR        ;IF INCORRECT GO REPORT SPURIOUS
1138                                                           ;FP TRAP.
1139
1140 006652  170204            0003:   STFPS   R4              ;GET FPS.
1141 006654  170305                    STST    R5              ;GET FEC.
1142 006656  010437  001240            MOV     R4,@#$TMP3      ;SAVE DATA INCASE OF ERROR.
1143 006662  010537  001242            MOV     R5,@#$TMP4
1144 006666  012702  100000            MOV     #100000,R2      ;EXPECTED FPS
1145 006672  012703  000002            MOV     #2,R3           ;EXPECTED FEC
1146 006676  010237  001244            MOV     R2,@#$TMP5
1147 006702  010337  001246            MOV     R3,@#$TMP6
1148 006706  022626                    CMP     (SP)+,(SP)+     ;RESET THE STACK.
1149
1150 006710  020204                    CMP     R2,R4           ;WAS FPS CORRECT?
1151 006712  001402                    BEQ     0004            ;BRANCH IF YES.
1152                                                           ;OTHERWISE REPORT FPS INCORRECTLY
1153 006714  104002            1$:     ERROR   +2              ;SET AFTER USE OF ILLEGAL ACC.
1154 006716  000403                    BR      OODONE
1155
1156 006720  020305            0004:   CMP     R3,R5           ;WAS THE FEC CORRECT?
1157 006722  001401                    BEQ     OODONE          ;BRANCH IF CORRECT.
1158                                                           ;OTHERWISE REPORT INCORRECT FEC
1159 006724  104003            1$:     ERROR   +3              ;AFTER USE OF ILLEGAL ACC.
1160
1161 006726                    OODONE:
     006726  104412                    RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                               ;SEE IF THE USER HAS EXPRESSED
                                                               ;THE DESIRE TO CHANGE THE SOFTWARE
                                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                               ;THE USER TYPED CONTROL G?).
1162
1163
1164
1170
1171                           ;;*****************************************************************
                               ;*TEST 2        FDST MODE 1, FLOATING MODE, TEST
                               ;*
                               ;*THIS IS A TEST OF THE STF INSTRUCTION USING FDST MODE 1.
                               ;*
                               ;;*****************************************************************
     006730  000004            TST2:   SCOPE
1172
1173 006732                    PPP1:
     006732  104413                    LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1174
1175 006734  012700  177777            MOV     #-1,R0          ;SET UP A BACKROUND PATTERN IN THE
1176 006740  012701  007070            MOV     #PPPBF0,R1      ;INPUT BUFFER.
```

```
1177 006744 012702 000014            MOV   #14,R2
1178 006750 010021            PPP2:  MOV   R0,(R1)+
1179 006752 077202                   SOB   R2,PPP2
1180
1181 006754 012700 000200            MOV   #200,R0          ;SET FD MODE.
1182 006760 170100                   LDFPS R0
1183 006762 012700 007120            MOV   #PPPTP1,R0       ;PUT TEST DATA INTO AC0.
1184 006766 172410                   LDD   (R0),AC0
1185
1186 006770 012700 007104            MOV   #PPPBF1,R0       ;FDST ADDRESS.
1187 006774 005002                   CLR   R2               ;CLEAR THE FPS.
1188 006776 170102                   LDFPS R2
1189 007000 012737 007012 001236     MOV   #PPP3,@#$TMP2
1190 007006 010037 001240            MOV   R0,@#$TMP3
1191
1192 007012 174010            PPP3:  STF   AC0,(R0)         ;TEST INSTRUCTION.
1193
1194 007014 022700 007104            CMP   #PPPBF1,R0       ;WAS R0 MODIFIED DURING EXECUTION?
1195 007020 001404                   BEQ   PPP4             ;BRANCH IF R0 NOT MODIFIED, CORRECT.
1196
1197 007022 010037 001242            MOV   R0,@#$TMP4       ;OTHERWISE REPORT ERROR, R0 MODIFIED.
1198 007026 104004            1$:    ERROR +4
1199 007030 000456                   BR    PPPDONE          ;GO TO NEXT TEST.
1200
1201 007032 012700 007104     PPP4:  MOV   #PPPBF1,R0       ;CHECK THE DATA IN THE OUTPUT BUFFER.
1202 007036 012701 007120            MOV   #PPPTP1,R1
1203 007042 022021                   CMP   (R0)+,(R1)+
1204 007044 001031                   BNE   PPP10            ;BRANCH IF INCORRECT.
1205 007046 022011                   CMP   (R0)+,(R1)
1206 007050 001027                   BNE   PPP10            ;BRANCH IF INCORRECT.
1207 007052 022720 177777            CMP   #-1,(R0)+        ;WAS FLOATING MODE USED?
1208 007056 001034                   BNE   PPP15            ;BRANCH IF NOT.
1209 007060 022710 177777            CMP   #-1,(R0)
1210 007064 001031                   BNE   PPP15
1211 007066 000437                   BR    PPPDONE ;GO TO NEXT TEST.
1212
1213 007070 177777 177777 177777 PPPBF0: .WORD  -1,-1,-1,-1,-1,-1
     007076 177777 177777 177777
1214
1215 007104 177777 177777 177777 PPPBF1: .WORD  -1,-1,-1,-1,-1,-1
     007112 177777 177777 177777
1216
1217 007120 123456 023456        PPPTP1: .WORD  123456,23456
1218 007124 034567 045671                .WORD  34567,45671
1219
1220                             ;REPORT DATA IN OUT PUT BUFFER INCORRECT.
1221 007130 012737 007120 001242 PPP10:  MOV   #PPPTP1,@#$TMP4
1222 007136 012737 007104 001240         MOV   #PPPBF1,@#$TMP3
1223 007144 104005               1$:     ERROR +5               ;BAD DATA.
1224 007146 000407                       BR    PPPDONE
1225
1226                             ;REPORT FLOATING MODE NOT USED, BUT FD FAILED.
1227 007150 012737 007120 001242 PPP15:  MOV   #PPPTP1,@#$TMP4
1228 007156 012737 007104 001240         MOV   #PPPBF1,@#$TMP3
1229 007164 104006               1$:     ERROR +6               ;ST 707 TO 245 INTO 244 (BUT FD).
1230
1231 007166                      PPPDONE:
```

```
        007166  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
   1232
   1233
   1234
   1240
   1241                                 ;;*****************************************************************
                                        ;*TEST 3        FDST MODE 2 TEST
                                        ;*
                                        ;*THIS IS A TEST OF BOTH STF AND STD WITH FDST MODE 2.
                                        ;*
                                        ;;*****************************************************************
        007170  000004                  TST3:   SCOPE
   1242
   1243                                 ;FIRST TEST STF.
   1244 007172                          QQQ1:
        007172  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
   1245
   1246 007174  012700  177777                  MOV     #-1,R0  ;SET UP THE OUTPUT BUFFER.
   1247 007200  012701  007332                  MOV     #QQQBF0,R1
   1248 007204  012702  000014                  MOV     #14,R2
   1249 007210  010021                  QQQ2:   MOV     R0,(R1)+
   1250 007212  077202                          SOB     R2,QQQ2
   1251
   1252 007214  012700  000200                  MOV     #200,R0         ;SET FD MODE.
   1253 007220  170100                          LDFPS   R0
   1254 007222  012700  007362                  MOV     #QQQTP1,R0      ;SETUP AC0.
   1255 007226  172410                          LDD     (R0),AC0
   1256
   1257 007230  012700  007346                  MOV     #QQQBF1,R0      ;FDST ADDRESS.
   1258 007234  005002                          CLR     R2
   1259 007236  170102                          LDFPS   R2              ;SET FPS.
   1260 007240  012737  007246  001236          MOV     #QQQ3,@#$TMP2
   1261
   1262 007246  174020                  QQQ3:   STF     AC0,(R0)+       ;TEST INSTRUCTION.
   1263
   1264 007250  022700  007352                  CMP     #QQQBF1+4,R0    ;WAS R0 INCREMENTED BY 4 PROPERLY?
   1265
   1266 007254  001407                          BEQ     QQQ4            ;BRANCH IF R0 CORRECT.
   1267 007256  010037  001242                  MOV     R0,@#$TMP4      ;REPORT R0 INCORRECT AFTER FDST MODE 2.
   1268 007262  012737  007352  001240          MOV     #QQQBF1+4,@#$TMP3
   1269 007270  104007                  1$:     ERROR   +7              ;BAD CONSTANT USED OR DIDN'T GO 527 TO 642
   1270 007272  000526                          BR      QQQDONE
   1271 007274  012700  007346          QQQ4:   MOV     #QQQBF1,R0      ;WAS THE OUTPUT DATA CORRECT?
   1272 007300  012701  007362                  MOV     #QQQTP1,R1
   1273 007304  022021                          CMP     (R0)+,(R1)+
   1274 007306  001031                          BNE     QQQ10           ;BRANCH IF INCORRECT.
   1275 007310  022021                          CMP     (R0)+,(R1)+
   1276 007312  001027                          BNE     QQQ10           ;BRANCH IF INCORRECT.
   1277 007314  022027  177777                  CMP     (R0)+,#-1       ;SEE IF ANY OTHER DATA BUFFER WORDS WERE MODIFIED.
   1278 007320  001024                          BNE     QQQ10           ;BRANCH IF INCORRECT.
   1279 007322  022027  177777                  CMP     (R0)+,#-1
   1280 007326  001021                          BNE     QQQ10           ;BRANCH IF INCORRECT.
   1281 007330  000430                          BR      QQQ20
```

```
1282 007332  177777  177777  177777  QQQBF0: .WORD   -1,-1,-1,-1,-1,-1
     007340  177777  177777  177777
1283 007346  177777  177777  177777  QQQBF1: .WORD   -1,-1,-1,-1,-1,-1
     007354  177777  177777  177777
1284 007362  076543                  QQQTP1: 76543
1285 007364  065432                          65432
1286 007366  054321                          54321
1287 007370  043210                          43210
1288                                 ;REPORT OUTPUT DATA INCORRECT:
1289 007372  012737  007362  001240  QQQ10:  MOV     #QQQTP1,@#$TMP3
1290 007400  012737  007346  001242          MOV     #QQQBF1,@#$TMP4
1291 007406  104010                  1$:     ERROR   +10             ;BAD DATA
1292 007410  000457                          BR      QQQDONE
1293
1294                                 ;NOW TEST STD MODE 2.
1295
1296 007412                          QQQ20:
     007412  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1297 007414  012700  007332                  MOV     #QQQBF0,R0      ;SET UP DEFAULT INPUT DATA BUFFER.
1298 007420  010001                          MOV     R0,R1
1299 007422  012702  000014                  MOV     #14,R2
1300 007426  010021                  QQQ22:  MOV     R0,(R1)+
1301 007430  077202                          SOB     R2,QQQ22
1302 007432  012700  000200                  MOV     #200,R0         ;ENTER FLOATING DOUBLE MODE.
1303 007436  170100                          LDFPS   R0
1304 007440  012700  007362                  MOV     #QQQTP1,R0      ;LOAD AC0.
1305 007444  172410                          LDD     (R0),AC0
1306 007446  012700  007346                  MOV     #QQQBF1,R0      ;SET DESTINATION ADDRESS.
1307 007452  012737  007460  001236          MOV     #QQQ23,@#$TMP2
1308 007460  174020                  QQQ23:  STD     AC0,(R0)+       ;TEST INSTRUCTION.
1309 007462  022700  007356                  CMP     #QQQBF1+10,R0   ;WAS R0 INCREMENTED BY 10 CORRECTLY?
1310 007466  001407                          BEQ     QQQ24           ;BRANCH IF CORRECT.
1311 007470  010037  001242                  MOV     R0,@#$TMP4      ;REPORT R0 INCORRECTLY INCREMENTED.
1312 007474  012737  007356  001240          MOV     #QQQBF1+10,@#$TMP3
1313 007502  104011                  1$:     ERROR   +11             ;DO NOT INCREM BY 10 BAD CONSTANT
1314 007504  000421                          BR      QQQDONE
1315 007506  012700  007346          QQQ24:  MOV     #QQQBF1,R0      ;DID THE DATA REACH THE OUTPUT BUFFER CORRECTLY?
1316 007512  012701  007362                  MOV     #QQQTP1,R1
1317 007516  012702  000004                  MOV     #4,R2
1318 007522  022021                  1$:     CMP     (R0)+,(R1)+
1319 007524  001002                          BNE     QQQ25           ;BRANCH IF INCORRECT.
1320 007526  077203                          SOB     R2,1$
1321 007530  000407                          BR      QQQDONE
1322                                 ;REPORT DATA INCORRECT.
1323 007532  012737  007362  001240  QQQ25:  MOV     #QQQTP1,@#$TMP3
1324 007540  012737  007346  001242          MOV     #QQQBF1,@#$TMP4
1325 007546  104012                  1$:     ERROR   +12             ;BAD DATA
1326 007550                          QQQDONE:
     007550  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                             ;THE USER TYPED CONTROL G?).
1327
1333                                 ;;*****************************************************************
                                     ;*TEST 4        FDST MODE 2, WITH GR7, TEST
                                     ;*
```

```
                                        ;*THIS IS A TEST OF STF WITH GR7 MODE 2 OR IMMEDIATE MODE.
                                        ;*
                                        ;;*****************************************************************
     007552  000004                     TST4:   SCOPE
1334
1335 007554                             RRR1:
     007554  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1336 007556  012700  007634                      MOV     #RRR3,R0        ;SET UP THE DATA BUFFER FOLLOWING THE TEST INSTRUCTION.
1337 007562  012701  007702                      MOV     #RRRTP1,R1
1338 007566  012702  000004                      MOV     #4,R2
1339 007572  012021                     1$:     MOV     (R0)+,(R1)+
1340 007574  077202                             SOB     R2,1$
1341 007576  012700  000200                      MOV     #200,R0         ;ENTER FLOATING DOUBLE MODE.
1342 007602  170100                             LDFPS   R0
1343 007604  012700  007712                      MOV     #RRRTP2,R0      ;SET UP AC0.
1344 007610  172410                             LDD     (R0),AC0
1345 007612  012737  007732  000004              MOV     #RRR10,@#ERRVECT ;SET UP FOR AN ODD ADDRESS.
1346 007620  012737  007632  001236              MOV     #RRR2,@#$TMP2
1347 007626  005001                             CLR     R1
1348 007630  005004                             CLR     R4
1349                                    ;THIS IS THE TEST INSTRUCTION. IT SHOULD MODIFY THE FIRST LOCATION
1350                                    ;AFTER IT TO BE AN INCREMENT R4, INC R4, INSTRUCTION INSTEAD
1351                                    ;OF AN INCREMENT R1 INSTRUCTION. THE INCREMENT R4 SHOULD NOT BE
1352                                    ;EXECUTED SINCE THE PC SHOULD BE INCREMENTED BY TWO DURING IMMEDIATE
1353                                    ;MODE ADDRESSING. THUS AFTER THE EXECUTION OF THE NEXT 5 INSTRUCTIONS
1354                                    ;R1 SHOULD CONTAIN 3 AND R4 SHOULD CONTAIN 0.
1355 007632  174027                     RRR2:   STD     AC0,(R7)+       ;TEST INSTRUCTION.
1356 007634  005201                     RRR3:   INC     R1              ;THE STD INSTRUCTION SHOULD CHANGE THIS TO INC R4.
1357 007636  005201                             INC     R1
1358 007640  005201                             INC     R1
1359 007642  005201                             INC     R1
1360 007644  012700  007722                      MOV     #RRREXP,R0      ;SEE IF THE DATA WAS OUTPUT CORRECTLY.
1361 007650  012702  007634                      MOV     #RRR3,R2
1362 007654  012703  000004                      MOV     #4,R3
1363 007660  022022                     RRR4:   CMP     (R0)+,(R2)+
1364 007662  001051                             BNE     RRR25           ;BRANCH IF INCORRECT.
1365 007664  077303                             SOB     R3,RRR4
1366 007666  005704                             TST     R4              ;MAKE SURE R4 IS 0.
1367 007670  001056                             BNE     RRR15           ;BRANCH IF R4 IS INCORRECT.
1368 007672  022701  000003                      CMP     #3,R1           ;SEE IF R1 IS CORRECT.
1369 007676  001053                             BNE     RRR15           ;BRANCH IF R1 IS INCORRECT.
1370 007700  000474                             BR      RRRDONE
1371                                    ;THESE ARE TEST DATA PATTERNS USED TO SET UP THE OUTPUT BUFFER AT RRR3.
1372 007702  005201                     RRRTP1: INC     R1
1373 007704  005201                             INC     R1
1374 007706  005201                             INC     R1
1375 007710  005201                             INC     R1
1376
1377 007712  005204                     RRRTP2: INC     R4
1378 007714  005204                             INC     R4
1379 007716  005204                             INC     R4
1380 007720  005204                             INC     R4
1381
1382 007722  005204                     RRREXP: INC     R4
1383 007724  005201                             INC     R1
1384 007726  005201                             INC     R1
1385 007730  005201                             INC     R1
```

```
                                        ;IF A FAILURE IN THE FDST FLOWS RESULTS IN AN ODD ADDRESS TRAP THROUGH
                                        ;4 TO HERE:
007732  011602                  RRR10:  MOV     (SP),R2         ;SEE IF THE TRAP WAS BECAUSE OF AN ODD ADDRESS.
007734  032702  000001                  BIT     #1,R2
007740  001005                          BNE     RRR11           ;BRANCH IF YES.
007742  020227  007636                  CMP     R2,#RRR3+2      ;SEE IF THE TRAP OCCURRED AT THE TEST INSTRUCTION.
007746  001412                          BEQ     RRR12           ;BRANCH IF YES.
007750  000137  046250                  JMP     @#CPSPUR        ;OTHERWISE REPORT A SPURIOUS TRAP THROUGH VECTOR 4.
                                        ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP.
007754  010237  001236          RRR11:  MOV     R2,@#$TMP2
007760  012737  007636  001240          MOV     #RRR3+2,@#$TMP3
007766  022626                          CMP     (SP)+,(SP)+
007770  104013                  1$:     ERROR   +13             ;BAD CONSTANT #2 + PC ODD ADDR.
007772  000437                          BR      RRRDONE
007774  010237  001236          RRR12:  MOV     R2,@#$TMP2
010000  022626                          CMP     (SP)+,(SP)+
010002  104014                  1$:     ERROR   +14             ;ODD ADDRESS TRAP
010004  000432                          BR      RRRDONE         ;WRONG MODE USED.

                                        ;REPORT DATA INCORRECT:
010006  012737  007634  001240  RRR25:  MOV     #RRR3,@#$TMP3
010014  012737  007722  001242          MOV     #RRREXP,@#$TMP4
010022  104015                  1$:     ERROR   +15             ;BAD DATA BUT GR7 FAIL
010024  000422                          BR      RRRDONE

                                        ;REPORT PC INCORRECT MODIFIED DURING THE EXECUTION OF FDST IMMEDIATE
                                        ;MODE. THE PC SHOULD HAV  BEEN INCREMENTED BY 2 BUT IT WASN'T.
                                        ;USE R1 AND R4 TO COMPU.  THE ACTUAL ACTION THAT WAS TAKEN ON THE PC.
010026  012737  007636  001240  RRR15:  MOV     #RRR3+2,@#$TMP3
010034  005704                          TST     R4              ;IS R4 CLEAR.
010036  001404                          BEQ     1$
010040  012737  007634  001242          MOV     #RRR3,@#$TMP4
010046  000410                          BR      2$
010050  012702  007636          1$:     MOV     #RRR3+2,R2
010054  062701  177775                  ADD     #-3,R1
010060  006301                          ASL     R1
010062  160102                          SUB     R1,R2
010064  010237  001242                  MOV     R2,@#$TMP4
010070                          2$:
010070  104016                  3$:     ERROR   +16             ;BAD CONSTANT PC+
010072                  RRRDONE:
010072  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).

;;*****************************************************************
;*TEST 5         FDST MODE 4 TEST
;*
;*THIS IS A TEST OF STD WITH FDST MODE 4.
;*
;;*****************************************************************
010074  000004                  TST5:   SCOPE

010076                          SSS1:
010076  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
1436 010100  012700  177777                MOV     #-1,R0          ;SET UP THE OUTPUT BUFFER.
1437 010104  012701  010234                MOV     #SSSBF0,R1
1438 010110  012702  000010                MOV     #10,R2
1439 010114  010021                1$:     MOV     R0,(R1)+
1440 010116  077202                        SOB     R2,1$
1441 010120  012700  000200                MOV     #200,R0         ;ENTER FLOATING DOUBLE MODE.
1442 010124  170100                        LDFPS   R0
1443 010126  012700  010254                MOV     #SSSTP1,R0      ;SET UP AC0.
1444 010132  172410                        LDD     (R0),AC0
1445 010134  012737  010274  000004        MOV     #SSS10,@#ERRVECT ;SET UP FOR A TRAP TO 4.
1446 010142  012737  010154  001236        MOV     #SSS2,@#$TMP2
1447 010150  012700  010244                MOV     #SSSA1,R0       ;SET UP THE DESTINATION ADDRESS.
1448
1449 010154  174040                SSS2:   STD     AC0,-(R0)       ;TEST INSTRUCTION.
1450 010156  005201                        INC     R1
1451 010160  020027  010234                CMP     R0,#SSSBF0      ;SEE IF R0 WAS DECREMENTED PROPERLY.
1452 010164  001060                        BNE     SSS15           ;BRANCH IF R0 IS INCORRECT.
1453 010166  012700  010234                MOV     #SSSBF0,R0      ;WAS THE OUTPUT DATA CORRECT?
1454 010172  012701  010254                MOV     #SSSTP1,R1
1455 010176  012702  000004                MOV     #4,R2
1456 010202  022021                1$:     CMP     (R0)+,(R1)+
1457 010204  001057                        BNE     SSS20           ;BRANCH IF INCORRECT.
1458 010206  077203                        SOB     R2,1$
1459 010210  012700  177777                MOV     #-1,R0          ;IS THE REST OF THE OUTPUT BUFFER CORRECT, -1?
1460 010214  012701  010244                MOV     #SSSA1,R1
1461 010220  012702  000004                MOV     #4,R2
1462 010224  020021                2$:     CMP     R0,(R1)+
1463 010226  001056                        BNE     SSS25           ;BRANCH IF INCORRECT.
1464 010230  077203                        SOB     R2,2$
1465 010232  000463                        BR      SSSDONE
1466
1467                               ;THIS IS THE OUTPUT DATA BUFFER.
1468 010234  177777                SSSBF0: -1
1469 010236  177777                        -1
1470 010240  177777                        -1
1471 010242  177777                        -1
1472 010244  177777                SSSA1:  -1
1473 010246  177777                        -1
1474 010250  177777                        -1
1475 010252  177777                        -1
1476
1477                               ;THIS IS THE TEST DATA LOADED INTO AC0:
1478 010254  147250                SSSTP1: 147250
1479 010256  036147                        36147
1480 010260  025036                        25036
1481 010262  147250                        147250
1482 010264  177777                SSSTP2: -1
1483 010266  177777                        -1
1484 010270  177777                        -1
1485 010272  177777                        -1
1486
1487                               ;IF AN ODD ADDRESS TRAP OCCURS  COME HERE:
1488 010274  011600                SSS10:  MOV     (SP),R0         ;SEE IF THE TRAP ACCURRED ON THE TEST INSTRUCTION.
1489 010276  020027  010156                CMP     R0,#SSS2+2
1490 010302  001405                        BEQ     SSS11           ;BRANCH IF YES.
1491 010304  020027  010160                CMP     R0,#SSS2+4
1492 010310  001402                        BEQ     SSS11           ;BRANCH IF YES.
```

```
1493 010312  000137  046250                 JMP     @#CPSPUR          ;OTHERWISE GO REPORT A SPURIOUS TRAP THROUGH 4.
1494                                ;REPORT FAILURE IN FDST FLOWS RESULTED IN AN ODD ADDRESS.
1495 010316  010037  001236         SSS11:  MOV     R0,@#$TMP2
1496 010322  104017                 2$:     ERROR   +17               ;FDST FORK X ODD AD RES.
1497 010324  000426                         BR      SSSDONE
1498
1499                                ;REPORT R0 INCORRECTLY DECREMENTED.
1500 010326  010037  001242         SSS15:  MOV     R0,@#$TMP4
1501 010332  012737  010234  001240         MOV     #SSSBF0,@#$TMP3
1502 010340  104020                 1$:     ERROR   +20               ;R0 NOT DECRE PROP
1503 010342  000417                         BR      SSSDONE
1504
1505                                ;REPORT OUTPUT DATA INCORRECT:
1506 010344  012737  010234  001240 SSS20:  MOV     #SSSBF0,@#$TMP3
1507 010352  012737  010254  001242         MOV     #SSSTP1,@#$TMP4
1508 010360  104021                 1$:     ERROR   +21               ;BAD DATA
1509 010362  000407                         BR      SSSDONE
1510 010364  012737  010244  001242 SSS25:  MOV     #SSSA1,@#$TMP4
1511 010372  012737  010264  001240         MOV     #SSSTP2,@#$TMP3
1512 010400  104022                 1$:     ERROR   +22               ;DATA BAD OUTSIDE TARGET AREA
1513 010402                         SSSDONE:
     010402  104412                         RSETUP                    ;GO INITIALIZE THE FPS AND STACK; AND
                                                                      ;SEE IF THE USER HAS EXPRESSED
                                                                      ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                      ;THE USER TYPED CONTROL G?).
1514
1520                                ;;***************************************************************
                                    ;*TEST 6          FDST MODE 3 TEST
                                    ;*
                                    ;*THIS IS A TEST OF FDST MODE 3 USING STD.
                                    ;*
                                    ;;***************************************************************
     010404  000004                 TST6:   SCOPE
1521
1522 010406                         TTT1:
     010406  104413                         LPERR                     ;SET UP THE LOOP ON ERROR ADDRESS.
1523 010410  012701  010526                 MOV     #TTTBF0,R1        ;SET UP THE OUTPUT DATA BUFFER.
1524 010414  012700  177777                 MOV     #-1,R0
1525 010420  012702  000012                 MOV     #12,R2
1526 010424  010021                 1$:     MOV     R0,(R1)+
1527 010426  077202                         SOB     R2,1$
1528 010430  012737  010526  010542         MOV     #TTTBF0,@#TTTA2
1529 010436  012700  000200                 MOV     #200,R0           ;ENTER DOUBLE FLOATING MODE.
1530 010442  170100                         LDFPS   R0
1531 010444  012700  010552                 MOV     #TTTTP1,R0        ;SET UP AC0.
1532 010450  172410                         LDD     (R0),AC0
1533 010452  012737  010562  000004         MOV     #TTT10,@#ERRVECT  ;SET UP FOR TRAPS TO 4.
1534 010460  016737  000006  001236         MOV     TTT2,@#$TMP2
1535 010466  012700  010542                 MOV     #TTTA2,R0         ;SET UP THE DESTINATION ADDRESS.
1536
1537 010472  174030                 TTT2:   STD     AC0,@(R0)+        ;TEST INSTRUCTION.
1538
1539 010474  020027  010544                 CMP     R0,#TTTA2+2       ;SEE IF R0 WAS INCREMENTED CORRECTLY.
1540 010500  001046                         BNE     TTT15             ;BRANCH IF INCORRECT.
1541 010502  012701  010526                 MOV     #TTTBF0,R1        ;CHECK THE OUTPUT DATA BUFFER.
1542 010506  012702  010552                 MOV     #TTTTP1,R2
```

```
1543 010512  012703  000004              MOV     #4,R3
1544 010516  022122              TTT3:   CMP     (R1)+,(R2)+
1545 010520  001045                      BNE     TTT20           ;BRANCH IF NOT CORRECT.
1546 010522  077303                      SOB     R3,TTT3
1547 010524  000452                      BR      TTTDONE
1548
1549                             ;THIS IS THHE OUTPUT DATA BUFFER:
1550 010526  177777              TTTBF0: -1
1551 010530  177777                      -1
1552 010532  177777                      -1
1553 010534  177777                      -1
1554 010536  177777                      -1
1555 010540  177777              TTTA1:  -1
1556 010542  010526              TTTA2:  TTTBF0
1557 010544  177777              TTTA3:  -1
1558 010546  177777                      -1
1559 010550  177777                      -1
1560 010552  101213              TTTTP1: 101213
1561 010554  141516                      141516
1562 010556  071727                      71727
1563 010560  037475                      37475
1564
1565                             ;TRAP THROUGH VECTOR 4 TO HERE.
1566 010562  011602              TTT10:  MOV     (SP),R2         ;SEE IF THE TRAP ADDRESS IS THAT OF THE TEST INSTRUCTION.
1567 010564  020227  010474              CMP     R2,#TTT2+2
1568 010570  001405                      BEQ     TTT11           ;BRANCH IF YES.
1569 010572  020227  010476              CMP     R2,#TTT2+4
1570 010576  001402                      BEQ     TTT11           ;BRANCH IF YES.
1571 010600  000137  046250              JMP     @#CPSPUR        ;OTHERWISE GO REPORT A SPURIOUS TRAP TO 4.
1572
1573                             ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP.
1574 010604  010237  001236      TTT11:  MOV     R2,@#$TMP2
1575 010610  022626                      CMP     (SP)+,(SP)+
1576 010612  104023              1$:     ERROR   +23             ;BET FDST X ODD ADR
1577 010614  000416                      BR      TTTDONE
1578
1579                             ;REPORT R0 INCORRECT:
1580 010616  010037  001242      TTT15:  MOV     R0,@#$TMP4
1581 010622  012737  010544  001240      MOV     #TTTA2+2,@#$TMP3
1582 010630  104024              1$:     ERROR   +24             ;R0 NOT INCREMENT PROPERLY
1583 010632  000407                      BR      TTTDONE
1584
1585                             ;REPORT INCORRECT OUTPUT DATA:
1586 010634  012737  010526  001240 TTT20:  MOV     #TTTBF0,@#$TMP3
1587 010642  012737  010552  001242      MOV     #TTTTP1,@#$TMP4
1588 010650  104025              1$:     ERROR   +25             ;BAD DATA
1589 010652                      TTTDONE:
     010652  104412                      RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                 ;SEE IF THE USER HAS EXPRESSED
                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                 ;THE USER TYPED CONTROL G?).
1590
1596                             ;;*****************************************************************
                                 ;*TEST 7         FDST MODE 5 TEST
                                 ;*
                                 ;*THIS IS A TEST OF FDST MODE 5 USING STD.
```

```
                                        ;*
                                        ;**********************************************************************
     010654  000004                     TST7:   SCOPE
1597
1598 010656                             UUU1:
     010656  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1599 010660  012701  010776                     MOV     #UUUBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
1600 010664  012700  177777                     MOV     #-1,R0
1601 010670  012702  000012                     MOV     #12,R2
1602 010674  010021                     1$:     MOV     R0,(R1)+
1603 010676  077202                             SOB     R2,1$
1604 010700  012737  010776  011010             MOV     #UUUBF0,@#UUUA1
1605 010706  012700  000200                     MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
1606 010712  170100                             LDFPS   R0
1607 010714  012700  011022                     MOV     #UUUTP1,R0      ;SET UP AC0.
1608 010720  172410                             LDD     (R0),AC0
1609 010722  012737  011032  000004             MOV     #UUU10,@#ERRVECT ;GET READY FOR ANY TRAPS TO 4.
1610 010730  016737  000006  001236             MOV     UUU2,@#$TMP2
1611 010736  012700  011012                     MOV     #UUUA2,R0       ;SET UP THE DESTINATION ADDRESS.
1612 010742  174050                     UUU2:   STD     AC0,@-(R0)      ;TEST INSTRUCTION.
1613 010744  020027  011010                     CMP     R0,#UUUA2-2     ;WAS R0 DECRIMENTED PROPERLY?
1614 010750  001046                             BNE     UUU15           ;BRANCH IF R0 IS INCORRECT.
1615 010752  012701  010776                     MOV     #UUUBF0,R1      ;WAS THE DATA OUTPUT CORRECTLY?
1616 010756  012702  011022                     MOV     #UUUTP1,R2
1617 010762  012703  000004                     MOV     #4,R3
1618 010766  022122                     UUU3:   CMP     (R1)+,(R2)+
1619 010770  001045                             BNE     UUU20           ;BRANCH IF DATA IS INCORRECT.
1620 010772  077303                             SOB     R3,UUU3
1621 010774  000452                             BR      UUUDONE
1622
1623                                    ;THIS IS THE OUTPUT DATA BUFFER
1624 010776  177777                     UUUBF0: -1
1625 011000  177777                             -1
1626 011002  177777                             -1
1627 011004  177777                             -1
1628 011006  177777                             -1
1629 011010  010776                     UUUA1:  UUUBF0
```

```
1631 011012  177777                UUUA2:  -1
1632 011014  177777                UUUA3:  -1
1633 011016  177777                        -1
1634 011020  177777                        -1
1635 011022  020212                UUUTP1: 20212
1636 011024  023242                        23242
1637 011026  026273                        26273
1638 011030  031323                        031323
1639
1640                               ;IF A TRAP TO 4 OCCURS COME HERE.
1641 011032  011602                UUU10:  MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
1642 011034  020227  010744                CMP     R2,#UUU2+2
1643 011040  001405                        BEQ     UUU11           ;BRANCH IF YES.
1644 011042  020227  010746                CMP     R2,#UUU2+4
1645 011046  001402                        BEQ     UUU11           ;BRANCH IF YES.
1646 011050  000137  046250                JMP     @#CPSPUR        ;OTHERWISE REPORT A SPURIOUS TRAP TO 4.
1647                               .REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4.
1648 011054  010237  001236        UUU11:  MOV     R2,@#$TMP2
1649 011060  022626                        CMP     (SP)+,(SP)+
1650 011062  104026                1$:     ERROR   +26             ;BET FDST X ODD ADR
1651 011064  000416                        BR      UUUDONE
1652
1653                               ;REPORT R0 INCORRECT.
1654 011066  010037  001242        UUU15:  MOV     R0,@#$TMP4
1655 011072  012737  011014  001240        MOV     #UUUA2+2,@#$TMP3
1656 011100  104027                1$:     ERROR   +27             ;R0 NOT INCREMENT PROPERLY
1657 011102  000407                        BR      UUUDONE
1658
1659                               ;REPORT BAD DATA.
1660 011104  012737  010776  001242 UUU20:  MOV     #UUUBF0,@#$TMP4
1661 011112  012737  011022  001240        MOV     #UUUTP1,@#$TMP3
1662 011120  104030                1$:     ERROR   +30             ;BAD DATA
```

```
1664 011122                          UUUDONE:
     011122  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
1665
```

```
1673                                ;;*****************************************************************
                                    ;*TEST 10       FDST MODE 6, INDEX MODE, TEST
                                    ;*
                                    ;*THIS IS A TEST OF FDST MODE 6, INDEX MODE, USING STD.
                                    ;*
                                    ;*****************************************************************
011124  000004                      TST10:  SCOPE
```

```

011126                                 VVV1:
011126  104413                                 LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
011130  012700  000200                         MOV     #200,R0             ;ENTER DOUBLE FLOATING MODE.
011134  170100                                 LDFPS   R0
011136  012701  011246                         MOV     #VVVBF0,R1          ;SET UP THE OUT PUT DATA BUFFER.
011142  012700  177777                         MOV     #-1,R0
011146  012702  000004                         MOV     #4,R2
011152  010021                         1$:     MOV     R0,(R1)+
011154  077202                                 SOB     R2,1$
011156  012737  011266  000004                 MOV     #VVV10,@#ERRVECT ;SET UP VECTOR 4 INCASE OF ERROR.
011164  012700  011256                         MOV     #VVVTP1,R0          ;SET UP AC0.
011170  172410                                 LDD     (R0),AC0
011172  012737  011210  001236                 MOV     #VVV2,@#$TMP2
011200  012700  003345                         MOV     #VVVBF0-5701,R0 ;SET UP THE DESTINATION ADDRESS.
011204  012701  000001                         MOV     #1,R1
011210  174060  005701                 VVV2:   STD     AC0,5701(R0)        ;TEST INSTRUCTION.

011214  020027  003345                         CMP     R0,#VVVBF0-5701 ;SEE IF R0 WAS MODIFIED.
011220  001040                                 BNE     VVV15               ;BRANCH IF INCORRECT.
011222  012702  011246                         MOV     #VVVBF0,R2          ;WAS THE OUTPUT DATA CORRECT.
011226  012703  011256                         MOV     #VVVTP1,R3
011232  012704  000004                         MOV     #4,R4
011236  022223                         1$:     CMP     (R2)+,(R3)+
011240  001037                                 BNE     VVV20               ;BRANCH IF INCORRECT DATA.
011242  077403                                 SOB     R4,1$
011244  000444                                 BR      VVVDONE
011246  177777                         VVVBF0: -1
011250  177777                                 -1
011252  177777                                 -1
011254  177777                                 -1
011256  030313                         VVVTP1: 30313
011260  023334                                 23334
011262  035363                                 35363
011264  074041                                 74041


                                       ;COME HERE AFTER A TRAP THROUGH VECTOR 4.
011266  011602                         VVV10:  MOV     (SP),R2             ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
011270  020227  011212                         CMP     R2,#VVV2+2
011274  001405                                 BEQ     VVV11               ;BRANCH IF YES.
011276  020227  011214                         CMP     R2,#VVV2+4
011302  001402                                 BEQ     VVV11               ;BRANCH IF YES.
011304  000137  046214                         JMP     @#FPSPUR            ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
                                       ;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4.
011310  010237  001236                 VVV11:  MOV     R2,@#$TMP2
011314  022626                                 CMP     (SP)+,(SP)+
011316  104031                         1$:     ERROR   +31                 ;FDST FORK X ODD ADD
011320  000416                                 BR      VVVDONE


                                       ;REPORT R0 MODIFIED.
011322  010037  001242                 VVV15:  MOV     R0,@#$TMP4
011326  012737  003345  001240                 MOV     #VVVBF0-5701,@#$TMP3
011334  104032                         1$:     ERROR   +32                 ;R0 MODIFIED!
011336  000407                                 BR      VVVDONE


                                       ;REPORT INCORRECT DATA.
011340  012737  011246  001240         VVV20:  MOV     #VVVBF0,@#$TMP3
```

```
1731 011346  012737  011256  001242          MOV     #VVVTP1,@#$TMP4
1732 011354  104033                  1$:     ERROR   +33             ;BAD DATA
1733 011356                          VVVDONE:
     011356  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                             ;THE USER TYPED CONTROL G?).
1734
1740                                 ;;*****************************************************************
                                     ;*TEST 11       FDST MODE 7, INDEX DEFERRED MODE, TEST
                                     ;*
                                     ;*THIS IS A TEST OF FDST MODE 7, INDEX DEFERRED MODE, USING STD.
                                     ;*
                                     ;;*****************************************************************
     011360  000004                  TST11:  SCOPE
1741
1742 011362                          WWW1:
     011362  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1743 011364  012700  000200                  MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
1744 011370  170100                          LDFPS   R0
1745 011372  012701  011510                  MOV     #WWWBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
1746 011376  012700  177777                  MOV     #-1,R0
1747 011402  012702  000004                  MOV     #4,R2
1748 011406  010021                  1$:     MOV     R0,(R1)+
1749 011410  077202                          SOB     R2,1$
1750 011412  012737  011540  000004          MOV     #WWW10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
1751 011420  012700  011520                  MOV     #WWWTP1,R0      ;SET UP AC0.
1752 011424  172410                          LDD     (R0),AC0
1753 011426  012737  011452  001236          MOV     #WWW2,@#$TMP2
1754 011434  012700  003627                  MOV     #WWWBF1-5701,R0 ;SET UP THE DESTINATION ADDRESS.
1755 011440  012701  000001                  MOV     #1,R1
1756 011444  012737  011510  011530          MOV     #WWWBF0,@#WWWBF1
1757 011452  174070  005701          WWW2:   STD     AC0,@5701(R0)   ;TEST INSTRUCTION.
1758
1759 011456  020027  003627                  CMP     R0,#WWWBF1-5701 ;IS R0 CORRECT?
1760 011462  001044                          BNE     WWW15           ;BRANCH IF INCORRECT.
1761 011464  012702  011510                  MOV     #WWWBF0,R2      ;WAS THE DATA OUTPUT CORRECTLY?
1762 011470  012703  011520                  MOV     #WWWTP1,R3
1763 011474  012704  000004                  MOV     #4,R4
1764 011500  022223                  1$:     CMP     (R2)+,(R3)+
1765 011502  001043                          BNE     WWW20           ;BRANCH IF DATA IS INCORRECT.
1766 011504  077403                          SOB     R4,1$
1767 011506  000450                          BR      WWWDONE
1768 011510  177777                  WWWBF0: -1
1769 011512  177777                          -1
1770 011514  177777                          -1
1771 011516  177777                          -1
1772 011520  041424                  WWWTP1: 41424
1773 011522  034445                          34445
1774 011524  046475                          46475
1775 011526  051525                          051525
1776 011530  177777                  WWWBF1: -1
1777 011532  177777                          -1
1778 011534  177777                          -1
1779 011536  177777                          -1
1780
```

```
1781                                    ;TRAP THROUGH 4 TO HERE.
1782 011540  011602                     WWW10:  MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
1783 011542  020227  011454                     CMP     R2,#WWW2+2
1784 011546  001405                             BEQ     WWW11           ;BRANCH IF YES.
1785 011550  020227  011456                     CMP     R2,#WWW2+4
1786 011554  001402                             BEQ     WWW11           ;BRANCH IF YES.
1787 011556  000137  046214                     JMP     @#FPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
1788                                    ;REPORT FAILURE OF FDST FORK RESULTED IN AN ODD ADDRESS TRAP TO 4.
1789 011562  010237  001236             WWW11:  MOV     R2,@#$TMP2
1790 011566  022626                             CMP     (SP)+,(SP)+
1791 011570  104034                     1$:     ERROR   +34             ;FDST FORK X ODD ADD
1792 011572  000416                             BR      WWWDONE
1793
1794                                    ;REPORT R0 MODIFIED.
1795 011574  010037  001242             WWW15:  MOV     R0,@#$TMP4
1796 011600  012737  003607  001240             MOV     #WWWBF0-5701,@#$TMP3
1797 011606  104035                     1$:     ERROR   +35             ;R0 MODIFIED!
1798 011610  000407                             BR      WWWDONE
1799
1800                                    ;REPORT DATA INCORRECT
1801 011612  012737  011510  001240     WWW20:  MOV     #WWWBF0,@#$TMP3
1802 011620  012737  011520  001242             MOV     #WWWTP1,@#$TMP4
1803 011626  104036                     1$:     ERROR   +36             ;BAD DATA
1804 011630                             WWWDONE:
     011630  104412                             RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
1805
1811                                    ;;**************************************************************
                                        ;*TEST 12       STCFD TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE STCFD INSTRUCTION.
                                        ;*
                                        ;;**************************************************************
     011632  000004                     TST12:  SCOPE
1812
1813                                    ;AC=0
1814 011634                             XXX1:
     011634  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1815 011636  004767  000330                     JSR     PC,STCFDS
1816 011642  000000                     1$:     0               ;AC
1817 011644  000000                             0
1818 011646  000000                             0
1819 011650  000000                             0
1820 011652  000000                     2$:     0               ;RES
1821 011654  000000                             0
1822 011656  000000                             0
1823 011660  000000                             0
1824 011662  000000                     3$:     0               ;ERROR RES.
1825 011664  000000                             0
1826 011666  177777                             -1
1827 011670  177777                             -1
1828 011672  047000                     4$:     47000           ;FPS BEFORE EXECUTION.
1829 011674  047004                             47004           ;FPS AFTER EXECUTION.
1830 011676  177777                             -1              ;FEC
```

```
1831 011700  147004                        147004          ;ERROR FPS.
1832 011702  104042                5$:     ERROR   +42             ;FDFL<---FDFLXST 767
1833 011704  000401                        BR      6$
1834 011706  104043                        ERROR   +43             ;BUT EZBT X ST560 TO 061 INTO 261
1835 011710                        6$:
1836                               ;
1837 011710                        XXX2:
     011710  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1838 011712  004767  000254                JSR     PC,STCFDS
1839 011716  017203                1$:     17203           ;AC
1840 011720  142536                        142536
1841 011722  047506                        47506
1842 011724  172031                        172031
1843 011726  017203                2$:     17203           ;RES
1844 011730  142536                        142536
1845 011732  000000                        0
1846 011734  000000                        0
1847 011736  017203                3$:     17203           ;ERROR RES.
1848 011740  142536                        142536
1849 011742  047506                        47506
1850 011744  172031                        172031
1851 011746  040000                4$:     40000           ;FPS BEFORE EXECUTION.
1852 011750  040000                        40000           ;FPS AFTER EXECUTION.
1853 011752  177777                        -1              ;FEC
1854 011754  177777                        -1              ;ERROR FPS.
1855 011756  104044                5$:     ERROR   +44             ;X11(1,0)<---0 X ST766
1856 011760  000401                        BR      6$
1857 011762  104040                        ERROR   +40
1858 011764                        6$:
1859                               ;
1860 011764                        XXX3:
     011764  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1861 011766  004767  000200                JSR     PC,STCFDS
1862 011772  050717                1$:     50717           ;AC
1863 011774  027374                        27374
1864 011776  075767                        75767
1865 012000  077071                        77071
1866 012002  050717                2$:     50717           ;RES
1867 012004  027374                        27374
1868 012006  000000                        0
1869 012010  000000                        0
1870 012012  000000                3$:     0               ;ERROR RES.
1871 012014  000000                        0
1872 012016  000000                        0
1873 012020  000000                        0
1874 012022  047000                4$:     47000           ;FPS BEFORE EXECUTION.
1875 012024  047000                        47000           ;FPS AFTER EXECUTION.
1876 012026  177777                        -1              ;FEC
1877 012030  174002                        174002          ;ERROR FPS.
1878 012032  104045                5$:     ERROR   +45             ;BUT OPIC X ST251
1879 012034  000401                        BR      6$
1880 012036  104046                        ERROR   +46             ;BUT EZBT X ST421
1881 012040                        6$:
1882                               ;
1883 012040                        XXX4:
     012040  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1884 012042  004767  000124                JSR     PC,STCFDS
```

```
1885 012046  020212                1$:     20212           ;AC
1886 012050  032425                        32425
1887 012052  026272                        26272
1888 012054  002123                        02123
1889 012056  020212                2$:     20212           ;RES
1890 012060  032425                        32425
1891 012062  000000                        0
1892 012064  000000                        0
1893 012066  020212                3$:     20212           ;ERROR RES.
1894 012070  032425                        32425
1895 012072  100000                        100000
1896 012074  000000                        0
1897 012076  040000                4$:     40000           ;FPS BEFORE EXECUTION.
1898 012100  040000                        40000           ;FPS AFTER EXECUTION.
1899 012102  177777                        -1              ;FEC
1900 012104  177777                        -1              ;ERROR FPS.
1901 012106  104047                5$:     ERROR   +47             ;BUT FD IN ROUND X ST113
1902 012110  000401                        BR      6$
1903 012112  104040                        ERROR   +40
1904 012114                        6$:
1905                               ;
1906 012114                        XXX5:
     012114  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1907 012116  004767  000050               JSR     PC,STCFDS
1908 012122  121314                1$:     121314          ;AC
1909 012124  151617                        151617
1910 012126  101112                        101112
1911 012130  131415                        131415
1912 012132  121314                2$:     121314          ;RES
1913 012134  151617                        151617
1914 012136  000000                        0
1915 012140  000000                        0
1916 012142  021314                3$:     21314           ;ERROR RES.
1917 012144  151617                        151617
1918 012146  000000                        0
1919 012150  000000                        0
1920 012152  040000                4$:     40000           ;FPS BEFORE EXECUTION.
1921 012154  040010                        40010           ;FPS AFTER EXECUTION.
1922 012156  177777                        -1              ;FEC
1923 012160  177777                        -1              ;ERROR FPS.
1924 012162  104050                5$:     ERROR   +50             ;BUT ENBT X ST567 OR BAD SIGN ST460
1925 012164  000401                        BR      6$
1926 012166  104040                        ERROR   +40
1927 012170  000535                6$:     BR      XXXDONE
1928
1929                               ;
1930
1931
1932                               ;THIS SUBROUTINE, STCFDS, IS USED TO SET UP THE OPERANDS, EXECUTE
1933                               ;THE STCFD INSTRUCTION AND CHECK THE RESULTS. A CALL
1934                               ;TO IT IS MADE THUS:
1935                               ;
1936                               ;               JSR     PC,@#STCFDS
1937                               ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
1938                               ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
1939                               ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
1940                               ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
```

```
1941                                    ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
1942                                    ;               FEC:    .WORD   X               ;EXPECTED FEC
1943                                    ;               ERFPS:  .WORD   X               ;ERROR FPS.
1944                                    ;               ERR1:   ERROR   +X              ;DATA ERROR.
1945                                    ;                       BR      CONT
1946                                    ;               ERR2:   ERROR   +X              ;FPS ERROR.
1947                                    ;               CONT:                           ;RETURN ADDRESS
1948                                    ;
1949                                    ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
1950                                    ;THE STCFD INSTRUCTION IS EXECUTED.
1951                                    ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
1952                                    ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
1953                                    ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
1954                                    ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
1955                                    ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
1956                                    ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
1957                                    ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
1958                                    ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
1959                                    ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
1960                                    ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
1961                                    ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
1962                                    ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
1963
1964 012172  012601                    STCFDS: MOV     (SP)+,R1                ;PICK UP THE POINTER TO THE OPERANDS.
1965 012174  012700  000200                    MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
1966 012200  170100                            LDFPS   R0
1967 012202  010100                            MOV     R1,R0           ;LOAD AC0.
1968 012204  172410                            LDD     (R0),AC0
1969 012206  012700  177777                    MOV     #-1,R0          ;FILL THE OUTPUT BUFFER WITH -1'S.
1970 012212  012702  012454                    MOV     #STCFT,R2
1971 012216  012703  000004                    MOV     #4,R3
1972 012222  010022                    1$:     MOV     R0,(R2)+
1973 012224  077302                            SOB     R3,1$
1974 012226  016100  000030                    MOV     30(R1),R0       ;LOAD THE FPS.
1975 012232  170100                            LDFPS   R0
1976 012234  012737  012246  001236            MOV     #2$,@#$TMP2
1977 012242  012700  012454                    MOV     #STCFT,R0       ;SET UP THE DESTINATION ADDRESS.
1978 012246  176010                    2$:     STCFD   AC0,(R0)        ;TEST INSTRUCTION.
1979
1980 012250  170204                            STFPS   R4              ;GET THE FPS.
1981 012252  170305                            STST    R5              ;GET THE FEC.
1982 012254  010102                            MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
1983 012256  010237  001240                    MOV     R2,@#$TMP3
1984 012262  062702  000010                    ADD     #10,R2
1985 012266  010237  001244                    MOV     R2,@#$TMP5
1986 012272  012737  012454  001242            MOV     #STCFT,@#$TMP4
1987 012300  010437  001250                    MOV     R4,@#$TMP7
1988 012304  016137  000032  001252            MOV     32(R1),@#$TMP10
1989
1990 012312  010102                            MOV     R1,R2           ;CHECK THE RESULT.
1991 012314  062702  000010                    ADD     #10,R2
1992 012320  012703  012454                    MOV     #STCFT,R3
1993 012324  012700  000004                    MOV     #4,R0
1994 012330  022223                    3$:     CMP     (R2)+,(R3)+
1995 012332  001014                            BNE     15$             ;BRANCH IF INCORRECT.
1996 012334  077003                            SOB     R0,3$
1997
```

```
1998 012336  016102  000032                MOV     32(R1),R2
1999 012342  020204                        CMP     R2,R4           ;IS THE FPS CORRECT?
2000 012344  001025                        BNE     20$             ;BRANCH IF FPS INCORRECT.
2001 012346  005702                        TST     R2              ;IF EXPECTED FPS IS NEGATIVE, THEN
2002 012350  100003                        BPL     4$              ;GO AHEAD AND CHECK THE FEC.
2003 012352  026105  000036                CMP     36(R1),R5
2004 012356  001027                        BNE     25$             ;BRANCH IF FEC IS INCORRECT.
2005 012360  000161  000046          4$:   JMP     46(R1)          ;RETURN.
2006
2007                                 ;RESULT INCORRECT:
2008 012364  010102                  15$:  MOV     R1,R2           ;SEE IF ERROR WAS ANTICIPATED.
2009 012366  062702  000020                ADD     #20,R2
2010 012372  012703  012454                MOV     #STCFT,R3
2011 012376  012700  000004                MOV     #4,R0
2012 012402  022223                  16$:  CMP     (R2)+,(R3)+
2013 012404  001003                        BNE     17$             ;BRANCH IF NOT ANTICIPATED.
2014 012406  077003                        SOB     R0,16$
2015 012410  000161  000040                JMP     40(R1)          ;IF ERROR WAS ANTICIPATED RETURN.
2016                                 ;OTHERWISE REPORT RESULT INCORRECT HERE.
2017 012414                          17$:
2018 012414  104037                  18$:  ERROR   +37             ;DATA ERROR
```

```
012416  000760                                  BR      4$
                                                
                                        ;FPS INCORRECT:
012420  020461  000034                  20$:    CMP     R4,34(R1)               ;WAS THE ERROR ANTICIPATED.
012424  001002                                  BNE     21$                     ;BRANCH IF NOT ANTICIPATED.
012426  000161  000044                          JMP     44(R1)                  ;IF IT WAS ANTICIPATED RETURN.

                                        ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE.
012432                                  21$:
012432  104040                          22$:    ERROR   +40                     ;FPS X
012434  000751                                  BR      4$

                                        ;REPORT FEC INCORRECT:
012436  016137  000036  001256          25$:    MOV     36(R1),@#$TMP12
012444  010537  001254                          MOV     R5,@#$TMP11
012450  104041                          26$:    ERROR   +41                     ;FEC X
012452  000742                                  BR      4$
012454  177777  177777  177777          STCFT:  -1,-1,-1,-1
012462  177777
012464                                  XXXDONE:
012464  104412                                  RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                                ;THE USER TYPED CONTROL G?).

                                        ;;*****************************************************************
                                        ;*TEST 13               STCDF TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE STCDF INSTRUCTION.
                                        ;*
                                        ;;*****************************************************************
012466  000004                          TST13:  SCOPE

                                        ;AC=0
012470                                  YYY1:
012470  104413                                  LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
012472  004767  000330                          JSR     PC,STCDFS
012476  000000                          1$:     0                               ;AC
012500  000000                                  0
012502  000000                                  0
012504  000000                                  0
012506  000000                          2$:     0                               ;RES
012510  000000                                  0
012512  177777                                  -1
012514  177777                                  -1
012516  000000                          3$:     0                               ;ERROR RES.
012520  000000                                  0
012522  000000                                  0
012524  000000                                  0
012526  047200                          4$:     47200                           ;FPS BEFORE EXECUTION.
012530  047204                                  47204                           ;FPS AFTER EXECUTION.
012532  177777                                  -1                              ;FEC
012534  177777                                  -1                              ;ERROR FPS.
012536  104054                          5$:     ERROR   +54                     ;FDFL<---FDFL X ST767
012540  000401                                  BR      6$
012542  104052                                  ERROR   +52                     ;FPS INCORRECT.
```

```
                                        6$:
                                        ;
                                        YYY2:
012544 104413                                   LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
012546 004767 000254                            JSR     PC,STCDFS
012552 067574                           1$:     67574                   ;AC0
012554 073727                                   73727
012556 170777                                   170777
012560 067574                                   67574
012562 067574                           2$:     67574                   ;RES
012564 073730                                   73730
012566 177777                                   -1
012570 177777                                   -1
012572 067574                           3$:     67574                   ;ERROR RES.
012574 073727                                   73727
012576 177777                                   -1
012600 177777                                   -1
012602 040200                           4$:     40200                   ;FPS BEFORE EXECUTION.
012604 040200                                   40200                   ;FPS AFTER EXECUTION.
012606 177777                                   -1                      ;FEC
012610 177777                                   -1                      ;ERROR FPS.
012612 104055                           5$:     ERROR   +55             ;EITHER ROUND FAILED OR WENT TO 766 X1(1,0)<---0 INTO 767
012614 000401                                   BR      6$
012616 104052                                   ERROR   +52
012620                                  6$:
                                        ;
                                        YYY3:
012620 104413                                   LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
012622 004767 000200                            JSR     PC,STCDFS
012626 077777                           1$:     77777                   ;AC0
012630 177777                                   -1
012632 100000                                   100000
012634 000000                                   0
012636 000000                           2$:     0                       ;RES
012640 000000                                   0
012642 177777                                   -1
012644 177777                                   -1
012646 077777                           3$:     77777                   ;ERROR RES.
012650 177777                                   -1
012652 177777                                   -1
012654 177777                                   -1
012656 040200                           4$:     40200                   ;FPS BEFORE EXECUTION.
012660 040206                                   40206                   ;FPS AFTER EXECUTION.
012662 177777                                   -1                      ;FEC
012664 040204                                   40204                   ;ERROR FPS.
012666 104055                           5$:     ERROR   +55
012670 000401                                   BR      6$
012672 104056                                   ERROR   +56             ;BUT EZBT X ST421 TO 062 INTO 262
012674                                  6$:
                                        ;
                                        YYY4:
012674 104413                                   LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
012676 004767 000124                            JSR     PC,STCDFS
012702 077777                           1$:     77777                   ;AC0
012704 177777                                   -1
012706 100000                                   100000
012710 000000                                   0
```

```
012712  000000                  2$:     0                               ;RES
012714  000000                          0
012716  177777                          -1
012720  177777                          -1
012722  077777                  3$:     77777                           ;ERROR RES.
012724  177777                          -1
012726  177777                          -1
012730  177777                          -1
012732  040200                  4$:     40200                           ;FPS BEFORE EXECUTION.
012734  040206                          40206                           ;FPS AFTER EXECUTION.
012736  177777                          -1                              ;FEC
012740  140206                          140206                          ;ERROR FPS.
012742  104055                  5$:     ERROR   +55
012744  000401                          BR      6$
012746  104057                          ERROR   +57                     ;BUT FIV ST262 TO 123 INTO 103
012750                          6$:
                                ;
012750                          YYY5:
012750  104413                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
012752  004767  000050                  JSR     PC,STCDFS
012756  177777                  1$:     177777                          ;AC0
012760  177777                          -1
012762  100000                          100000
012764  000000                          0
012766  100000                  2$:     100000                          ;RES
012770  000000                          0
012772  177777                          -1
012774  177777                          -1
012776  000000                  3$:     0                               ;ERROR RES.
013000  000000                          0
013002  177777                          -1
013004  177777                          -1
013006  047200                  4$:     47200                           ;FPS BEFORE EXECUTION.
013010  147216                          147216                          ;FPS AFTER EXECUTION.
013012  000010                          10                              ;FEC
013014  047206                          47206                           ;ERROR FPS.
013016  104060                  5$:     ERROR   +60                     ;BUT FIV ST262 FAIL TO 103 INT 123
013020  000401                          BR      6$
013022  104061                          ERROR   +61                     ;BUT FLAG ST 147 X TO ST 361 INTO 365
013024  000535                  6$:     BR      YYYDONE
                                ;THIS SUBROUTINE, STCDFS, IS USED TO SET UP THE OPERANDS, EXECUTE
                                ;THE STCDF INSTRUCTION AND CHECK THE RESULTS. A CALL
                                ;TO IT IS MADE THUS:
                                ;
                                ;               JSR     PC,@#STCFDS
                                ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
                                ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               FEC:    .WORD   X               ;EXPECTED FEC
                                ;               ERFPS:  .WORD   X               ;ERROR FPS.
                                ;               ERR1:   ERROR   +X              ;DATA ERROR.
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   +X              ;FPS ERROR.
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
```

```
                                        ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                        ;THE STCFD INSTRUCTION IS EXECUTED.
                                        ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                        ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
                                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
                                        ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
                                        ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
                                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                        ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                        ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                        ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
                                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

013026  012601                  STCDFS: MOV     (SP)+,R1        ;PICK UP THE POINTER TO THE OPERANDS.
013030  012700  000200                  MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
013034  170100                          LDFPS   R0
013036  010100                          MOV     R1,R0           ;LOAD AC0.
013040  172410                          LDD     (R0),AC0
013042  012700  177777                  MOV     #-1,R0          ;FILL THE OUTPUT BUFFER WITH -1'S.
013046  012702  013310                  MOV     #STCDT,R2
013052  012703  000004                  MOV     #4,R3
013056  010022                  1$:     MOV     R0,(R2)+
013060  077302                          SOB     R3,1$
013062  016100  000030                  MOV     30(R1),R0       ;LOAD THE FPS.
013066  170100                          LDFPS   R0
013070  012737  013102  001236          MOV     #2$,@#$TMP2
013076  012700  013310                  MOV     #STCDT,R0       ;SET UP THE DESTINATION ADDRESS.
013102  176010                  2$:     STCDF   AC0,(R0)        ;TEST INSTRUCTION.

013104  170204                          STFPS   R4              ;GET THE FPS.
013106  170305                          STST    R5              ;GET THE FEC.
013110  010102                          MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
013112  010237  001240                  MOV     R2,@#$TMP3
013116  062702  000010                  ADD     #10,R2
013122  010237  001244                  MOV     R2,@#$TMP5
013126  012737  013310  001242          MOV     #STCDT,@#$TMP4
013134  010437  001250                  MOV     R4,@#$TMP7
013140  016137  000032  001252          MOV     32(R1),@#$TMP10

013146  010102                          MOV     R1,R2           ;CHECK THE RESULT.
013150  062702  000010                  ADD     #10,R2
013154  012703  013310                  MOV     #STCDT,R3
013160  012700  000004                  MOV     #4,R0
013164  022223                  3$:     CMP     (R2)+,(R3)+
013166  001014                          BNE     15$             ;BRANCH IF INCORRECT.
013170  077003                          SOB     R0,3$

013172  016102  000032                  MOV     32(R1),R2
013176  020204                          CMP     R2,R4           ;IS THE FPS CORRECT?
013200  001025                          BNE     20$             ;BRANCH IF FPS INCORRECT.
013202  005702                          TST     R2              ;IF EXPECTED FPS IS NEGATIVE, THEN
013204  100003                          BPL     4$              ;GO AHEAD AND CHECK THE FEC.
013206  026105  000034                  CMP     34(R1),R5
013212  001027                          BNE     25$             ;BRANCH IF FEC IS INCORRECT.
013214  000161  000046          4$:     JMP     46(R1)          ;RETURN.
```

```
2236
2237                                    ;RESULT INCORRECT:
2238 013220  010102                    15$:    MOV     R1,R2           ;SEE IF ERROR WAS ANTICIPATED.
2239 013222  062702  000020                    ADD     #20,R2
2240 013226  012703  013310                    MOV     #STCDT,R3
2241 013232  012700  000004                    MOV     #4,R0
2242 013236  022223                    16$:    CMP     (R2)+,(R3)+
2243 013240  001003                            BNE     17$             ;BRANCH IF NOT ANTICIPATED.
2244 013242  077003                            SOB     R0,16$
2245 013244  000161  000040                    JMP     40(R1)          ;IF ERROR WAS ANTICIPATED RETURN.
2246                                    ;OTHERWISE REPORT RESULT INCORRECT HERE.
2247 013250                            17$:
2248 013250  104051                    18$:    ERROR   +51             ;DATA ERROR
2249 013252  000760                            BR      4$
2250
2251                                    ;FPS INCORRECT:
2252 013254  020461  000034            20$:    CMP     R4,34(R1)               ;WAS THE ERROR ANTICIPATED.
2253 013260  001002                            BNE     21$             ;BRANCH IF NOT ANTICIPATED.
2254 013262  000161  000044                    JMP     44(R1)          ;IF IT WAS ANTICIPATED RETURN.
2255
2256                                    ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE.
2257 013266                            21$:
2258 013266  104052                    22$:    ERROR   +52             ;FPS X
2259 013270  000751                            BR      4$
2260
2261                                    ;REPORT FEC INCORRECT:
2262 013272  016137  000036  001256    25$:    MOV     36(R1),@#$TMP12
2263 013300  010537  001254                    MOV     R5,@#$TMP11
2264 013304  104053                    26$:    ERROR   +53             ;FEC X
2265 013306  000742                            BR      4$
2266 013310  177777  177777  177777  STCDT:  -1,-1,-1,-1
     013316  177777
2267 013320                            YYYDONE:
     013320  104412                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
2273                                    ;;**********************************************************************
                                        ;*TEST 14       STCFD WITH ILLEGAL ACCUMULATOR TEST
                                        ;*
                                        ;*THIS TEST STCFD WITH ILLEGAL AC 6.
                                        ;*
                                        ;;**********************************************************************
     013322  000004                     TST14:  SCOPE
2274
2275 013324                             ZZZ1:
     013324  104413                             LPERR                  ;SET UP THE LOOP ON ERROR ADDRESS.
2276 013326  012700  040000                    MOV     #40000,R0       ;DISSABLE INTERRUPTS.
2277 013332  170100                            LDFPS   R0
2278 013334  012737  013342  001236            MOV     #ZZZ2,@#$TMP2
2279 013342  176006                    ZZZ2:   STCFD   AC0,AC6         ;THIS TEST INSTRUCTION SHOULD CAUSE AN ERROR.
2280
2281 013344  170204                            STFPS   R4              ;GET FPS.
2282 013346  170305                            STST    R5              ;GET FEC.
2283 013350  020427  140000                    CMP     R4,#140000      ;IS FPS CORRECT?
2284 013354  001004                            BNE     ZZZ10           ;BRANCH IF INCORRECT FPS.
```

```
 2285 013356  022705  000002                    CMP     #2,R5           ;IS FEC CORRECT?
 2286 013362  001010                            BNE     ZZZ15           ;BRANCH IF INCORRECT.
 2287 013364  000415                            BR      ZZZDONE
 2288
 2289                                   ;REPORT FPS INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR.
 2290 013366  010437  001242            ZZZ10:  MOV     R4,@#$TMP4
 2291 013372  012737  140000  001240            MOV     #140000,@#$TMP3
 2292 013400  104062                    1$:     ERROR   +62             ;BUT FDST ST767 X TO 567 INTO 577
 2293 013402  000406                            BR      ZZZDONE
 2294
 2295                                   ;REPORT FEC INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR.
 2296 013404  010537  001242            ZZZ15:  MOV     R5,@#$TMP4
 2297 013410  012737  000002  001240            MOV     #2,@#$TMP3
 2298 013416  104063                    1$:     ERROR   +63             ;FEC<---2 ST577 X
 2299 013420                            ZZZDONE:
      013420  104412                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                         ;SEE IF THE USER HAS EXPRESSED
                                                                         ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                         ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                         ;THE USER TYPED CONTROL G?).
 2300
 2306                                   ;;*****************************************************************
                                        ;*TEST 15       CLRD TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS.
                                        ;*
                                        ;;*****************************************************************
      013422  000004                    TST15:  SCOPE
 2307 013424                            AAB1:
      013424  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 2308 013426  012700  013612                    MOV     #AABTP1,R0      ;SET UP OUTPUT BUFFER
 2309 013432  012701  013602                    MOV     #AABBF0,R1
 2310 013436  012702  000004                    MOV     #4,R2
 2311 013442  012021                    1$:     MOV     (R0)+,(R1)+
 2312 013444  077202                            SOB     R2,1$
 2313 013446  012700  013602                    MOV     #AABBF0,R0      ;SET UP DESTINATION OPERAND ADDRESS.
 2314 013452  012701  000213                    MOV     #213,R1         ;SET UP FPS.
 2315 013456  170101                            LDFPS   R1
 2316 013460  012737  013466  001236            MOV     #2$,@#$TMP2
 2317 013466  170410                    2$:     CLRD    (R0)            ;TEST INSTRUCTION.
 2318
 2319 013470  170205                            STFPS   R5              ;GET FPS.
 2320 013472  012702  000004                    MOV     #4,R2           ;SEE IF RESULT CLEAR, 0.
 2321 013476  012701  013602                    MOV     #AABBF0,R1
 2322 013502  005721                    3$:     TST     (R1)+
 2323 013504  001010                            BNE     AAB2            ;BRANCH IF RESULT INCORRECT, NOT 0.
 2324 013506  077203                            SOB     R2,3$
 2325 013510  022705  000204                    CMP     #204,R5         ;SEE IF FPS IS CORRECT.
 2326 013514  001014                            BNE     AAB3            ;BRANCH IF INCORRECT.
 2327 013516  020027  013602                    CMP     R0,#AABBF0      ;SEE IF R0 IS CORRECT.
 2328 013522  001020                            BNE     AAB4            ;BRANCH IF R0 IS INCORRECT.
 2329 013524  000442                            BR      AABDONE
 2330
 2331                                   ;RESULT NOT 0, REPORT ERROR.
 2332 013526  012737  013602  001240    AAB2:   MOV     #AABBF0,@#$TMP3
 2333 013534  012737  013622  001242            MOV     #AABTP2,@#$TMP4
 2334 013542  104064                    1$:     ERROR   +64             ;BAD DATA = 0 X 11+ZERO ST770 X
```

```
2335 013544  000432                          BR      AABDONE
2336
2337                                         ;REPORT FPS INCORRECT:
2338 013546  010437  001242                  AAB3:   MOV     R4,@#$TMP4
2339 013552  012737  000204  001240                  MOV     #204,@#$TMP3
2340 013560  104065                          1$:     ERROR   +65             ;BAD FPS
2341 013562  000423                                  BR      AABDONE
2342
2343                                         ;REPORT R0 INCORRECT.
2344 013564  010037  001242                  AAB4:   MOV     R0,@#$TMP4
2345 013570  012737  013602  001240                  MOV     #AABBF0,@#$TMP3
2346 013576  104066                          1$:     ERROR   +66
2347 013600  000414                                  BR      AABDONE
2348
2349                                         ;THIS IS THE TEST DATA BUFFER, OUTPUT DATA BUFFER.
2350 013602  073475                          AABBF0: 73475
2351 013604  067707                                  67707
2352 013606  127347                                  127347
2353 013610  056770                                  56770
2354                                         ;THIS IS THE DATA USED TO SET UP THE OUTPUT BUFFER.
2355 013612  073475                          AABTP1: 73475
2356 013614  067707                                  67707
2357 013616  127347                                  127347
2358 013620  056770                                  56770
2359                                         ;THIS IS THE EXPECTED DATA, RESULT:
2360 013622  000000                          AABTP2: 0
2361 013624  000000                                  0
2362 013626  000000                                  0
2363 013630  000000                                  0
2364 013632                                  AABDONE:
     013632  104412                                  RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                             ;SEE IF THE USER HAS EXPRESSED
                                                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                             ;THE USER TYPED CONTROL G?).
2365
2371                                         ;;*****************************************************************
                                             ;*TEST 16       CLRD WITH ILLEGAL ACCUMULATOR TEST
                                             ;*
                                             ;*THIS IS A TEST OF CLRD WITH ILLEGAL AC7.
                                             ;*
                                             ;;*****************************************************************
     013634  000004                          TST16:  SCOPE
2372 013636                                  CCB1:
     013636  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2373 013640  012700  040200                          MOV     #40200,R0       ;SET UP THE FPS, NO INTERRUPTS AND FD=1.
2374 013644  170100                                  LDFPS   R0
2375 013646  012737  013654  001236                  MOV     #CCB2,@#$TMP2
2376 013654  170407                          CCB2:   CLRD    AC7             ;TEST INSTRUCTION.
2377
2378 013656  170204                                  STFPS   R4              ;GET FPS.
2379 013660  170305                                  STST    R5              ;GET FEC.
2380 013662  020427  140200                          CMP     R4,#140200              ;IS THE FPS CORRECT?
2381 013666  001004                                  BNE     CCB10           ;BRANCH IF FPS IS INCORRECT.
2382 013670  022705  000002                          CMP     #2,R5           ;IS THE FEC CORRECT?
2383 013674  001010                                  BNE     CCB15           ;BRANCH IF FEC IS INCORRECT.
2384 013676  000415                                  BR      CCBDONE
```

```
                                        ;REPORT INCORRECT FPS:
013700  010437  001242                  CCB10:  MOV     R4,@#$TMP4
013704  012737  140200  001240                  MOV     #140200,@#$TMP3
013712  104067                          1$:     ERROR   +67             ;BUT FDST ST 700X TO 607 INTO 677
013714  000406                                  BR      CCBDONE

                                        ;REPORT INCORRECT FEC:
013716  010537  001242                  CCB15:  MOV     R5,@#$TMP4
013722  012737  000002  001240                  MOV     #2,@#$TMP3
013730  104070                          1$:     ERROR   +70             ;FEC<---2 ST 677 X
013732                                  CCBDONE:
013732  104412                                  RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).

                                            ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
                                        ;;*****************************************************************
                                        ;*TEST 17       SEE ABOVE COMMENT FOR TEST TITLE
                                        ;*
                                        ;*THIS IS A TEST OF THE SPECIAL
                                        ;*DEST FLOWS USING THE NEGD INST
                                        ;*WITH MODE ZERO AND ILLEGAL
                                        ;*AC7.
                                        ;*
                                        ;;*****************************************************************
013734  000004                          TST17:  SCOPE

013736                                  VVB1:
013736  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
013740  012700  040200                          MOV     #40200,R0               ;SET UP THE FPS, FID=1 AND FD=1.
013744  170100                                  LDFPS   R0
013746  012737  013754  001236                  MOV     #VVB2,@#$TMP2

013754  170707                          VVB2:   NEGD    AC7             ;TEST INSTRUCTION.

013756  170204                                  STFPS   R4              ;GET FPS.
013760  170305                                  STST    R5              ;GET FEC.

013762  022704  140200                          CMP     #140200,R4              ;IS FPS CORRECT?
013766  001004                                  BNE     VVB10           ;BRANCH IF FPS IS INCORRECT.
013770  022705  000002                          CMP     #2,R5           ;IS FEC CORRECT?
013774  001010                                  BNE     VVB15           ;BRANCH IF FEC IS INCORRECT.
013776  000415                                  BR      VVBDONE

                                        ;REPORT INCORRECT FPS:
014000  012737  140200  001240          VVB10:  MOV     #140200,@#$TMP3
014006  010437  001242                          MOV     R4,@#$TMP4
014012  104176                          1$:     ERROR   +176            ;FPS BAD
014014  000406                                  BR      VVBDONE

                                        ;REPORT FEC INCORRECT:
014016  012737  000002  001240          VVB15:  MOV     #2,@#$TMP3
014024  010537  001242                          MOV     R5,@#$TMP4
014030  104177                          1$:     ERROR   +177            ;FEC BAD
```

```
2435
2436 014032                          VVBDONE:
     014032  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
2437
2445                                          ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 0 TEST
2446                                  ;;**********************************************************************
                                      ;*TEST 20       SEE ABOVE COMMENT FOR TEST TITLE
                                      ;*
                                      ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                      ;*SOURCE FLOWS.  THE NEGD INSTRUCTION
                                      ;*IS USED TO TEST MODE 0
                                      ;*
                                      ;;**********************************************************************
     014034  000004                   TST20:  SCOPE
2447
2448 014036                           DDB1:
     014036  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2449 014040  012700  000200                  MOV     #200,R0         ;SET FD MODE.
2450 014044  170100                          LDFPS   R0
2451 014046  012700  014210                  MOV     #DDBTP1,R0      ;SET UP AC0.
2452 014052  172410                          LDD     (R0),AC0                ;SET AC0 = 0
2453 014054  005000                          CLR     R0              ;CLEAR THE FPS.
2454 014056  170100                          LDFPS   R0
2455 014060  012700  014220                  MOV     #DDBTP2,R0              ;LOAD AC0 TO BE A FLOATING 0.
2456 014064  172410                          LDF     (R0),AC0        ;SET AC0=ZERO
2457                                                                 ;FLOAT
2458 014066  012700  000201                  MOV     #201,R0         ;SET FD MODE.
2459 014072  170100                          LDFPS   R0
2460 014074  012737  014102  001236          MOV     #DDB2,@#$TMP2
2461
2462 014102  170700                   DDB2:  NEGD    AC0             ;TEST INSTRUCTION.
2463
2464 014104  170205                          STFPS   R5              ;GET FPS.
2465 014106  012700  000200                  MOV     #200,R0         ;SET FD MODE.
2466 014112  170100                          LDFPS   R0
2467 014114  012700  014230                  MOV     #DDBBF0,R0      ;GET THE RESULT OUT OF AC0.
2468 014120  174010                          STD     AC0,(R0)
2469                                                                 ;SEE IF THE RESULT IS CORRECT.
2470 014122  012701  000004                  MOV     #4,R1
2471 014126  005720                   1$:    TST     (R0)+
2472 014130  001005                          BNE     DDB5            ;BRANCH IF THE RESULT IS INCORRECT.
2473 014132  077103                          SOB     R1,1$
2474 014134  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
2475 014140  001014                          BNE     DDB6            ;BRANCH IF THE FPS IS INCORRECT.
2476 014142  000442                          BR      DDBDONE
2477
2478                                  ;RESULT INCORRECT, REPORT FAILURE:
2479 014144  012737  014220  001242   DDB5:  MOV     #DDBTP2,@#$TMP4 ;EXPECT D0
2480 014152  012737  014240  001240          MOV     #DDBTP3,@#$TMP3 ;PREV F0 IMPURE
2481 014160  012737  014230  001244          MOV     #DDBBF0,@#$TMP5 ;GOT
2482 014166  104071                   1$:    ERROR   +71
2483 014170  000427                          BR      DDBDONE
2484
```

```
                                        ;REPORT FPS INCORRECT:
014172  012737  000204  001240  DDB6:   MOV     #204,@#$TMP3
014200  010537  001242                  MOV     R5,@#$TMP4
014204  104072                  1$:     ERROR   +72
014206  000420                          BR      DDBDONE

                                        ;THESE ARE TEST DATA TABLES AND AN OUTPUT BUFFER.
014210  101112                  DDBTP1: 101112
014212  131415                          131415
014214  161710                          161710
014216  111213                          111213
014220  000000                  DDBTP2: 0
014222  000000                          0
014224  000000                          0
014226  000000                          0

014230  177777                  DDBBF0: -1
014232  177777                          -1
014234  177777                          -1
014236  177777                          -1
014240  000000                  DDBTP3: 0
014242  000000                          0
014244  161710                          161710
014246  111213                          111213

014250                          DDBDONE:
014250  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).

                                        ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 1 TEST
                                ;;**************************************************************
                                ;*TEST 21       SEE ABOVE COMMENT FOR TEST TITLE
                                ;*
                                ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                ;*SOURCE FLOWS.  THE NEGD INSTRUCTION
                                ;*IS USED TO TEST MODE 1
                                ;*
                                ;;**************************************************************
014252  000004                  TST21:  SCOPE

014254                          EEB1:
014254  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
014256  012700  014364                  MOV     #EEBTP1,R0      ;SET UP THE DATA BUFFER.
014262  012701  014414                  MOV     #EEBBF1,R1
014266  012702  000004                  MOV     #4,R2
014272  012021          1$:     MOV     (R0)+,(R1)+
014274  077202                  SOB     R2,1$
014276  012700  000200                  MOV     #200,R0         ;SET FD MODE.
014302  170100                  LDFPS   R0
014304  012700  014414                  MOV     #EEBBF1,R0      ;SET UP THE OPERAND ADDRESS.
014310  012737  014324  001236          MOV     #EEB2,@#$TMP2
014316  012737  014424  000004          MOV     #EEB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF ERROR.
014324  170710          EEB2:   NEGD    (R0)                    ;TEST INSTRUCTION.
```

```
2528 014326  170205                        STFPS   R5              ;GET FPS.
2529 014330  012701  014414                MOV     #EEBBF1,R1      ;SEE IF RESULT IS CORRECT.
2530 014334  012702  000004                MOV     #4,R2
2531 014340  005721                1$:     TST     (R1)+
2532 014342  001046                        BNE     EEB15           ;BRANCH IF NOT CORRECT.
2533 014344  077203                        SOB     R2,1$
2534
2535 014346  020027  014414                CMP     R0,#EEBBF1      ;IS R0 CORRECT?
2536 014352  001055                        BNE     EEB20                   ;BRANCH IF NOT CORRECT.
2537 014354  022705  000204                CMP     #204,R5         ;IS THE FPS CORRECT?
2538 014360  001061                        BNE     EEB25           ;BRANCH IF NOT CORRECT.
2539 014362  000466                        BR      EEBDONE
2540
2541                               ;THESE ARE TEST DATA TABLES AND A BUFFER.
2542 014364  000177                EEBTP1: 177
2543 014366  167574                        167574
2544 014370  137271                        137271
2545 014372  107675                        107675
2546 014374  000000                EEBTP2: 0
2547 014376  000000                        0
2548 014400  000000                        0
2549 014402  000000                        0
2550 014404  177777                EEBBF0: -1
2551 014406  177777                        -1
2552 014410  177777                        -1
2553 014412  177777                        -1
2554 014414  177777                EEBBF1: -1
2555 014416  177777                        -1
2556 014420  177777                        -1
2557 014422  177777                        -1
2558
2559                               ;IF A TRAP TO 4 OCCURS COME HERE:
2560 014424  011602                EEB10:  MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
2561 014426  020227  014326                CMP     R2,#EEB2+2
2562 014432  001405                        BEQ     1$              ;BRANCH IF YES.
2563 014434  020227  014330                CMP     R2,#EEB2+4
2564 014440  001402                        BEQ     1$              ;BRANCH IF YES.
2565 014442  000137  046250                JMP     @#CPSPUR        ;OTHERWISE GO REPORT A SPURIOUS TRAP TO 4.
2566                               ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP TO 4.
2567 014446  022626                1$:     CMP     (SP)+,(SP)+     ;RESET THE STACK.
2568 014450  010237  001236                MOV     R2,@#$TMP2
2569 014454  104107                2$:     ERROR   +107            ;ODD ADRES
2570 014456  000430                        BR      EEBDONE         ;BUT FDSTX IN ST 771
2571
2572                               ;REPORT RESULT INCORRECT.
2573 014460  012737  014374  001242 EEB15:  MOV     #EEBTP2,@#$TMP4
2574 014466  012737  014364  001240        MOV     #EEBTP1,@#$TMP3
2575 014474  012737  014414  001244        MOV     #EEBBF1,@#$TMP5
2576 014502  104073                1$:     ERROR   +73             ;BAD DATA X11+0 ST 312X
2577 014504  000415                        BR      EEBDONE
2578
2579                               ;R0 INCORRECT:
2580 014506  012737  014414  001240 EEB20:  MOV     #EEBBF1,@#$TMP3
2581 014514  010037  001242                MOV     R0,@#$TMP4
2582 014520  104074                1$:     ERROR   +74             ;R0 BADX
2583 014522  000406                        BR      EEBDONE
2584
```

```
                                        ;REPORT FPS INCORRECT:
014524  010537  001240                  EEB25:  MOV     R5,@#$TMP3
014530  012737  000204  001244                  MOV     #204,@#$TMP5
014536  104075                          1$:     ERROR   +75             ;FPS X

014540                                  EEBDONE:
014540  104412                                  RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).

                                        ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 2 TEST
                                        ;;*****************************************************************
                                        ;*TEST 22       SEE ABOVE COMMENT FOR TEST TITLE
                                        ;*
                                        ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                        ;*SOURCE FLOWS.  THE ABSD INSTRUCTION
                                        ;*IS USED TO TEST MODE 2
                                        ;*
                                        ;*****************************************************************
014542  000004                          TST22:  SCOPE

014544                                  FFB1:
014544  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
014546  012700  014654                          MOV     #FFBTP1,R0              ;SET UP THE DATA BUFFER.
014552  012701  014704                          MOV     #FFBBF1,R1
014556  012702  000004                          MOV     #4,R2
014562  012021                          1$:     MOV     (R0)+,(R1)+
014564  077202                                  SOB     R2,1$
014566  012700  000200                          MOV     #200,R0         ;SET FD.
014572  170100                                  LDFPS   R0
014574  012700  014704                          MOV     #FFBBF1,R0      ;SET UP THE OPERAND ADDRESS.
014600  012737  014614  001236                  MOV     #FFB2,@#$TMP2
014606  012737  014714  000004                  MOV     #FFB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

014614  170620                  FFB2:   ABSD    (R0)+                   ;TEST INSTRUCTION.

014616  170205                          STFPS   R5                      ;GET FPS.
014620  012701  014704                  MOV     #FFBBF1,R1                      ;CHECK RESULT.
014624  012702  000004                  MOV     #4,R2
014630  005721                  1$:     TST     (R1)+
014632  001046                          BNE     FFB15           ;BRANCH IF INCORRECT.
014634  077203                          SOB     R2,1$

014636  020027  014714                  CMP     R0,#FFBBF1+10   ;IS R0 CORRECT?
014642  001055                          BNE     FFB20           ;BRANCH IF INCORRECT.
014644  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
014650  001061                          BNE     FFB25           ;BRANCH IF INCORRECT.
014652  000466                          BR      FFBDONE

                                        ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
014654  000177                  FFBTP1: 177
014656  167574                          167574
014660  137271                          137271
014662  107675                          107675
014664  000000                  FFBTP2: 0
```

```
 2628 014666  000000                          0
 2629 014670  000000                          0
 2630 014672  000000                          0
 2631 014674  177777                FFBBF0:   -1
 2632 014676  177777                          -1
 2633 014700  177777                          -1
 2634 014702  177777                          -1
 2635 014704  177777                FFBBF1:   -1
 2636 014706  177777                          -1
 2637 014710  177777                          -1
 2638 014712  177777                          -1
 2639
 2640                               ;IF A TRAP TO 4 OCCURS COME HERE.
 2641 014714  011602                FFB10:    MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
 2642 014716  020227  014616                   CMP     R2,#FFB2+2
 2643 014722  001405                          BEQ     1$              ;BRANCH IF YES.
 2644 014724  020227  014620                   CMP     R2,#FFB2+4
 2645 014730  001402                          BEQ     1$              ;BRANCH IF YES.
 2646 014732  000137  046250                   JMP     @#CPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
 2647                               ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
 2648 014736  022626                1$:       CMP     (SP)+,(SP)+
 2649 014740  010237  001236                   MOV     R2,@#$TMP2
 2650 014744  104076                2$:       ERROR   +76             ;ODD ADRES
 2651 014746  000430                          BR      FFBDONE         ;BUT FDSTX IN ST 771
 2652
 2653                               ;REPORT RESULT INCORRECT:
 2654 014750  012737  014664  001240 FFB15:    MOV     #FFBTP2,@#$TMP3
 2655 014756  012737  014654  001242          MOV     #FFBTP1,@#$TMP4
 2656 014764  012737  014704  001244          MOV     #FFBBF1,@#$TMP5
 2657 014772  104077                1$:       ERROR   +77             ;BAD DATA X11*0 ST 312X
 2658 014774  000415                          BR      FFBDONE
 2659
 2660                               ;REPORT R0 INCORRECT:
 2661 014776  012737  014710  001240 FFB20:    MOV     #FFBBF1+4,@#$TMP3
 2662 015004  010037  001242                   MOV     R0,@#$TMP4
 2663 015010  104100                1$:       ERROR   +100            ;R0 BADX
 2664 015012  000406                          BR      FFBDONE
 2665
 2666                               ;REPORT FPS INCORRECT:
 2667 015014  010537  001240        FFB25:    MOV     R5,@#$TMP3
 2668 015020  012737  000204  001244          MOV     #204,@#$TMP5
 2669 015026  104101                1$:       ERROR   +101            ;FPS X
 2670
 2671 015030                        FFBDONE:
      015030  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                               ;SEE IF THE USER HAS EXPRESSED
                                               ;THE DESIRE TO CHANGE THE SOFTWARE
                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                               ;THE USER TYPED CONTROL G?).
 2672                                   ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
 2673                               ;;*****************************************************************
                                    ;*TEST 23       SEE ABOVE COMMENT FOR TEST TITLE
                                    ;*
                                    ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                    ;*SOURCE FLOWS.  THE ABSD INSTRUCTION
                                    ;*IS USED TO TEST MODE 4
                                    ;*
```

```
                                ;;*************************************************************
        015032  000004          TST23:  SCOPE
2674
```

```
2676 015034                          GGB1:
     015034  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2677 015036  012700  015144                 MOV     #GGBTP1,R0              ;SET UP THE DATA BUFFER.
2678 015042  012701  015164                 MOV     #GGBBF0,R1
2679 015046  012702  000004                 MOV     #4,R2
2680 015052  012021                  1$:    MOV     (R0)+,(R1)+
```

```
2682 015054  077202                        SOB     R2,1$
2683 015056  012700  000200                MOV     #200,R0          ;SET FD.
2684 015062  170100                        LDFPS   R0
2685 015064  012700  015174                MOV     #GGBBF1,R0       ;SET UP THE OPERAND ADDRESS.
2686 015070  012737  015104  001236        MOV     #GGB2,@#$TMP2
2687 015076  012737  015204  000004        MOV     #GGB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
2688
2689 015104  170640                GGB2:   ABSD    -(R0)            ;TEST INSTRUCTION.
2690
2691 015106  170205                        STFPS   R5               ;GET FPS.
2692 015110  012701  015164                MOV     #GGBBF0,R1               ;CHECK RESULT.
2693 015114  012702  000004                MOV     #4,R2
2694 015120  005721                1$:     TST     (R1)+
2695 015122  001046                        BNE     GGB15            ;BRANCH IF INCORRECT.
2696 015124  077203                        SOB     R2,1$
2697
2698 015126  020027  015164                CMP     R0,#GGBBF0       ;IS R0 CORRECT?
2699 015132  001055                        BNE     GGB20            ;BRANCH IF INCORRECT.
2700 015134  022705  000204                CMP     #204,R5          ;IS THE FPS CORRECT?
2701 015140  001061                        BNE     GGB25            ;BRANCH IF INCORRECT.
2702 015142  000466                        BR      GGBDONE
2703
2704                               ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
2705 015144  000177                GGBTP1: 177
2706 015146  117273                        117273
2707 015150  147576                        147576
2708 015152  177071                        177071
2709 015154  000000                GGBTP2: 0
2710 015156  000000                        0
2711 015160  000000                        0
2712 015162  000000                        0
2713 015164  177777                GGBBF0: -1
2714 015166  177777                        -1
2715 015170  177777                        -1
2716 015172  177777                        -1
2717 015174  177777                GGBBF1: -1
2718 015176  177777                        -1
2719 015200  177777                        -1
2720 015202  177777                        -1
2721
2722                               ;IF A TRAP TO 4 OCCURS COME HERE.
2723 015204  011602                GGB10:  MOV     (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2724 015206  020227  015106                CMP     R2,#GGB2+2
2725 015212  001405                        BEQ     1$               ;BRANCH IF YES.
2726 015214  020227  015110                CMP     R2,#GGB2+4
2727 015220  001402                        BEQ     1$               ;BRANCH IF YES.
2728 015222  000137  046250                JMP     @#CPSPUR         ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
2729                               ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
2730 015226  022626                1$:     CMP     (SP)+,(SP)+
2731 015230  010237  001236                MOV     R2,@#$TMP2
2732 015234  104102                2$:     ERROR   +102             ;ODD ADRES
2733 015236  000430                        BR      GGBDONE          ;BUT FDSTX IN ST 771
2734
2735                               ;REPORT RESULT INCORRECT:
2736 015240  012737  015154  001240 GGB15:  MOV     #GGBTP2,@#$TMP3
2737 015246  012737  015144  001242        MOV     #GGBTP1,@#$TMP4
2738 015254  012737  015164  001244        MOV     #GGBBF0,@#$TMP5
```

```
 2739 015262  104103                    1$:     ERROR   +103            ;BAD DATA X11*0 ST 312X
 2740 015264  000415                            BR      GGBDONE
 2741
 2742                                    ;REPORT R0 INCORRECT:
 2743 015266  012737  015164  001240     GGB20:  MOV     #GGBBF01,@#$TMP3
 2744 015274  010037  001242                    MOV     R0,@#$TMP4
 2745 015300  104104                    1$:     ERROR   +104            ;R0 BADX
 2746 015302  000406                            BR      GGBDONE
 2747
 2748                                    ;REPORT FPS INCORRECT:
 2749 015304  010537  001240             GGB25:  MOV     R5,@#$TMP3
 2750 015310  012737  000204  001244             MOV     #204,@#$TMP5
 2751 015316  104105                    1$:     ERROR   +105            ;FPS X
 2752
 2753 015320                            GGBDONE:
      015320  104412                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                         ;SEE IF THE USER HAS EXPRESSED
                                                                         ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                         ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                         ;THE USER TYPED CONTROL G?).
 2754                                           ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
 2755                                    ;;*****************************************************************
                                         ;*TEST 24       SEE ABOVE COMMENT FOR TEST TITLE
                                         ;*
                                         ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                         ;*SOURCE FLOWS.  THE ABSD INSTRUCTION
                                         ;*IS USED TO TEST MODE 3
                                         ;*
                                         ;;*****************************************************************
      015322  000004                     TST24:  SCOPE
 2756
 2757 015324                             HHB1:
      015324  104413                             LPERR                  ;SET UP THE LOOP ON ERROR ADDRESS.
 2758 015326  012700  015434                     MOV     #HHBTP1,R0             ;SET UP THE DATA BUFFER.
 2759 015332  012701  015464                     MOV     #HHBBF0,R1
 2760 015336  012702  000010                     MOV     #10,R2
 2761 015342  012021                    1$:     MOV     (R0)+,(R1)+
 2762 015344  077202                             SOB     R2,1$
 2763 015346  012700  000200                     MOV     #200,R0        ;SET FD.
 2764 015352  170100                             LDFPS   R0
 2765 015354  012700  015474                     MOV     #HHBBF1,R0     ;SET UP THE OPERAND ADDRESS.
 2766 015360  012737  015374  001236             MOV     #HHB2,@#$TMP2
 2767 015366  012737  015504  000004             MOV     #HHB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
 2768
 2769 015374  170630                     HHB2:   ABSD    @(R0)+         ;TEST INSTRUCTION.
 2770
 2771 015376  170205                             STFPS   R5             ;GET FPS.
 2772 015400  012701  015464                     MOV     #HHBBF0,R1             ;CHECK RESULT.
 2773 015404  012702  000004                     MOV     #4,R2
 2774 015410  005721                    1$:     TST     (R1)+
 2775 015412  001052                             BNE     HHB15          ;BRANCH IF INCORRECT.
 2776 015414  077203                             SOB     R2,1$
 2777 015416  020027  015476                     CMP     R0,#HHBBF1+2   ;IS R0 CORRECT?
 2778 015422  001061                             BNE     HHB20          ;BRANCH IF INCORRECT.
 2779 015424  022705  000204                     CMP     #204,R5        ;IS THE FPS CORRECT?
 2780 015430  001065                             BNE     HHB25          ;BRANCH IF INCORRECT.
 2781 015432  000472                             BR      HHBDONE
```

```
                                        ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
015434  000177                  HHBTP1: 177
015436  147576                          147576
015440  177071                          177071
015442  107576  015464  177777          107576,HHBBF0,-1,-1,-1
015450  177777  177777
015454  000000  000000  000000  HHBTP2: 0,0,0,0
015462  000000
                                HHBBF0: -1
015464  177777                          -1
015466  177777                          -1
015470  177777                          -1
015472  177777                          -1
015474  177777                  HHBBF1: -1
015476  177777                          -1
015500  177777                          -1
015502  177777                          -1

                                ;IF A TRAP TO 4 OCCURS COME HERE.
015504  011602                  HHB10:  MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
015506  020227  015376                  CMP     R2,#HHB2+2
015512  001405                          BEQ     1$              ;BRANCH IF YES.
015514  020227  015400                  CMP     R2,#HHB2+4
015520  001402                          BEQ     1$              ;BRANCH IF YES.
015522  000137  046250                  JMP     @#CPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
                                ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
015526  022626                  1$:     CMP     (SP)+,(SP)+
015530  010237  001236                  MOV     R2,@#$TMP2
015534  104106                  2$:     ERROR   +106            ;ODD ADRES
015536  000430                          BR      HHBDONE         ;BUT FDSTX IN ST 771

                                ;REPORT RESULT INCORRECT:
015540  012737  015454  001240  HHB15:  MOV     #HHBTP2,@#$TMP3
015546  012737  015434  001242          MOV     #HHBTP1,@#$TMP4
015554  012737  015464  001244          MOV     #HHBBF0,@#$TMP5
015562  104110                  1$:     ERROR   +110            ;BAD DATA X11*0 ST 3127
015564  000415                          BR      HHBDONE

                                ;REPORT R0 INCORRECT:
015566  012737  015476  001240  HHB20:  MOV     #HHBBF1+2,@#$TMP3
015574  010037  001242                  MOV     R0,@#$TMP4
015600  104111                  1$:     ERROR   +111            ;R0 INCORRECT.
015602  000406                          BR      HHBDONE
                                ;REPORT FPS INCORRECT:
015604  010537  001240          HHB25:  MOV     R5,@#$TMP3
015610  012737  000204  001244          MOV     #204,@#$TMP5
015616  104112                  1$:     ERROR   +112            ;FPSX

015620                          HHBDONE:
015620  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).
                                        ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
                                ;;**************************************************************
                                ;*TEST 25        SEE ABOVE COMMENT FOR TEST TITLE
```

```
                                        ;*
                                        ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                        ;*SOURCE FLOWS.  THE NEGD INSTRUCTION
                                        ;*IS USED TO TEST MODE 5
                                        ;*
                                        ;;*****************************************************************
     015622  000004                     TST25:  SCOPE
2831
2832 015624                             IIB1:
     015624  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2833 015626  012700  015734                     MOV     #IIBTP1,R0              ;SET UP THE DATA BUFFER.
2834 015632  012701  015764                     MOV     #IIBBF0,R1
2835 015636  012702  000010                     MOV     #10,R2
2836 015642  012021                     1$:     MOV     (R0)+,(R1)+
2837 015644  077202                             SOB     R2,1$
2838 015646  012700  000200                     MOV     #200,R0         ;SET FD.
2839 015652  170100                             LDFPS   R0
2840 015654  012700  015776                     MOV     #IIBBF1+2,R0    ;SET UP THE OPERAND ADDRESS.
2841 015660  012737  015674  001236             MOV     #IIB2,@#$TMP2
2842 015666  012737  016004  000004             MOV     #IIB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
2843
2844 015674  170750                     IIB2:   NEGD    @-(R0)          ;TEST INSTRUCTION.
2845
2846 015676  170205                             STFPS   R5              ;GET FPS.
2847 015700  012701  015764                     MOV     #IIBBF0,R1              ;CHECK RESULT.
2848 015704  012702  000004                     MOV     #4,R2
2849 015710  005721                     1$:     TST     (R1)+
2850 015712  001052                             BNE     IIB15           ;BRANCH IF INCORRECT.
2851 015714  077203                             SOB     R2,1$
2852 015716  020027  015774                     CMP     R0,#IIBBF1      ;IS R0 CORRECT?
2853 015722  001061                             BNE     IIB20           ;BRANCH IF INCORRECT.
2854 015724  022705  000204                     CMP     #204,R5         ;IS THE FPS CORRECT?
2855 015730  001065                             BNE     IIB25           ;BRANCH IF INCORRECT.
2856 015732  000472                             BR      IIBDONE
2857
2858                                    ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
2859 015734  000176                     IIBTP1: 176
2860 015736  177074                             177074
2861 015740  127374                             127374
2862 015742  157677  015764  177777             157677,IIBBF0,-1,-1,-1
     015750  177777  177777
2863 015754  000000                     IIBTP2: 0
2864 015756  000000                             0
2865 015760  000000                             0
2866 015762  000000                             0
2867 015764  177777                     IIBBF0: -1
2868 015766  177777                             -1
2869 015770  177777                             -1
2870 015772  177777                             -1
2871 015774  177777                     IIBBF1: -1
2872 015776  177777                             -1
2873 016000  177777                             -1
2874 016002  177777                             -1
2875
2876                                    ;IF A TRAP TO 4 OCCURS COME HERE.
2877 016004  011602                     IIB10:  MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2878 016006  020227  015676                     CMP     R2,#IIB2+2
```

```
                                        BEQ     1$                      ;BRANCH IF YES.
                                        CMP     R2,#IIB2+4
                                        BEQ     1$                      ;BRANCH IF YES.
                                        JMP     @#CPSPUR                ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
                                ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
                                1$:     CMP     (SP)+,(SP)+
                                        MOV     R2,@#$TMP2
                                2$:     ERROR   +113                    ;ODD ADRES
                                        BR      IIBDONE                 ;BUT FDSTX IN ST 771

                                ;REPORT RESULT INCORRECT:
                                IIB15:  MOV     #IIBTP2,@#$TMP3
                                        MOV     #IIBTP1,@#$TMP4
                                        MOV     #IIBBF0,@#$TMP5
                                1$:     ERROR   +114                    ;BAD DATA X11*0 ST 3127
                                        BR      IIBDONE

                                ;REPORT R0 INCORRECT:
                                IIB20:  MOV     #IIBBF1,@#$TMP3
                                        MOV     R0,@#$TMP4
                                1$:     ERROR   +115                    ;R0 BADX
                                        BR      IIBDONE
                                ;REPORT FPS INCORRECT:
                                IIB25:  MOV     R5,@#$TMP3
                                        MOV     #204,@#$TMP5
                                1$:     ERROR   +116                    ;FPSX

                                IIBDONE:
016120  104412                          RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).


                                        ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 6 TEST
                                ;;**************************************************************
                                ;*TEST 26       SEE ABOVE COMMENT FOR TEST TITLE
                                ;*
                                ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                ;*SOURCE FLOWS.  THE ABSD INSTRUCTION
                                ;*IS USED TO TEST MODE 6
                                ;*
                                ;;**************************************************************
016122  000004                  TST26:  SCOPE

                                JJB1:
016124  104413                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
016126  012700  016236                  MOV     #JJBTP1,R0                      ;SET UP THE DATA BUFFER.
016132  012701  016260                  MOV     #JJBBF0,R1
016136  012702  000004                  MOV     #4,R2
016142  012021                  1$:     MOV     (R0)+,(R1)+
016144  077202                          SOB     R2,1$
016146  012700  000200                  MOV     #200,R0                 ;SET FD.
016152  170100                          LDFPS   R0
016154  012700  016251                  MOV     #JJBBF0-7,R0            ;SET UP THE OPERAND ADDRESS.
016160  012737  016174  001236          MOV     #JJB2,@#$TMP2
016166  012737  016300  000004          MOV     #JJB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
```

```
                                JJB2:   ABSD    7(R0)           ;TEST INSTRUCTION.

                                        STFPS   R5              ;GET FPS.
                                        MOV     #JJBBF0,R1              ;CHECK RESULT.
                                        MOV     #4,R2
                                1$:     TST     (R1)+
                                        BNE     JJB15           ;BRANCH IF INCORRECT.
                                        SOB     R2,1$
                                        CMP     R0,#JJBBF0-7    ;IS R0 CORRECT?
                                        BNE     JJB15           ;BRANCH IF INCORRECT.
                                        CMP     #204,R5         ;IS THE FPS CORRECT?
                                        BNE     JJB20           ;BRANCH IF INCORRECT.
                                        BR      JJBDONE

                                ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
                                JJBTP1: 177
                                        161524
                                        131273
                                        107174,
                                JJBTP2: 0
                                        0
                                        0
                                        0
                                JJBBF0: -1
                                        -1
                                        -1
                                        -1
                                JJBBF1: -1
                                        -1
                                        -1
                                        -1

                                ;IF A TRAP TO 4 OCCURS COME HERE.
                                JJB10:  MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
                                        CMP     R2,#JJB2+2
                                        BEQ     1$              ;BRANCH IF YES.
                                        CMP     R2,#JJB2+4
                                        BEQ     1$              ;BRANCH IF YES.
                                        JMP     @#CPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.

                                ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
                                1$:     CMP     (SP)+,(SP)+
                                        MOV     R2,@#$TMP2
                                2$:     ERROR   +117            ;ODD ADRES
                                        BR      JJBDONE         ;BUT FDSTX IN ST 771

                                ;REPORT RESULT INCORRECT:
                                JJB15:  MOV     #JJBTP2,@#$TMP3
                                        MOV     #JJBTP1,@#$TMP4
                                        MOV     #JJBBF0,@#$TMP5
                                1$:     ERROR   +120            ;BAD DATA X11*0 ST 3127
                                        BR      JJBDONE

                                ;REPORT R0 INCORRECT:
                                JJB20:  MOV     #JJBBF0-7,@#$TMP3
                                        MOV     R0,@#$TMP4
                                1$:     ERROR   +124            ;R0 BADX
```

```
                                        BR      JJBDONE
                                ;REPORT FPS INCORRECT:
001240                          JJB25:  MOV     R5,@#$TMP3
001244                                  MOV     #204,@#$TMP5
                                1$:     ERROR   +122                    ;FPSX
                                JJBDONE:
                                        RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
                                        ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
                                ;;**************************************************************
                                ;*TEST 27       SEE ABOVE COMMENT FOR TEST TITLE
                                ;*
                                ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                ;*SOURCE FLOWS.  THE ABSD INSTRUCTION
                                ;*IS USED TO TEST MODE 6
                                ;*
                                ;;**************************************************************
000004                          TST27:  SCOPE

                                KKB1:
                                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
016532                                  MOV     #KKBTP1,R0                      ;SET UP THE DATA BUFFER.
016562                                  MOV     #KKBBF0,R1
000010                                  MOV     #10,R2
                                1$:     MOV     (R0)+,(R1)+
                                        SOB     R2,1$
000200                                  MOV     #200,R0                 ;SET FD.
                                        LDFPS   R0
016563                                  MOV     #KKBBF1-7,R0    ;SET UP THE OPERAND ADDRESS.
001236                                  MOV     #KKB2,@#$TMP2
000004                                  MOV     #KKB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

000007                  KKB2:   NEGD    @7(R0)                  ;TEST INSTRUCTION.

                                STFPS   R5                      ;GET FPS.
016562                          MOV     #KKBBF0,R1                      ;CHECK RESULT.
000004                          MOV     #4,R2
                        1$:     TST     (R1)+
                                BNE     KKB15                   ;BRANCH IF INCORRECT.
                                SOB     R2,1$
016563                          CMP     R0,#KKBBF1-7    ;IS R0 CORRECT?
                                BNE     KKB20                   ;BRANCH IF INCORRECT.
000204                          CMP     #204,R5                 ;IS THE FPS CORRECT?
                                BNE     KKB20                   ;BRANCH IF INCORRECT.
                                BR      KKBDONE

                        ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
                        KKBTP1: 177
                                167574
                                137271
016562  177777                  107675,KKBBF0,-1,-1,-1
177777  177777
                        KKBTP2: 0
                                0
```

```
3021 016556  000000                          0
3022 016560  000000                          0
3023 016562  177777              KKBBF0:     -1
3024 016564  177777                          -1
3025 016566  177777                          -1
3026 016570  177777                          -1
3027 016572  177777              KKBBF1:     -1
3028 016574  177777                          -1
3029 016576  177777                          -1
```

```
3031 016600  177777                        -1
3032
3033                                ;IF A TRAP TO 4 OCCURS COME HERE.
3034 016602  011602                 KKB10:  MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
3035 016604  020227  016472                 CMP     R2,#KKB2+2
3036 016610  001405                         BEQ     1$              ;BRANCH IF YES.
3037 016612  020227  016474                 CMP     R2,#KKB2+4
3038 016616  001402                         BEQ     1$              ;BRANCH IF YES.
3039 016620  000137  046250                 JMP     @#CPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3040                                ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3041 016624  022626                 1$:     CMP     (SP)+,(SP)+
3042 016626  010237  001236                 MOV     R2,@#$TMP2
3043 016632  104123                 2$:     ERROR   +123            ;ODD ADRES
3044 016634  000430                         BR      KKBDONE         ;BUT FDSTX IN ST 771
3045
3046                                ;REPORT RESULT INCORRECT:
3047 016636  012737  016552  001240 KKB15:  MOV     #KKBTP2,@#$TMP3
3048 016644  012737  016532  001242         MOV     #KKBTP1,@#$TMP4
3049 016652  012737  016562  001244         MOV     #KKBBF0,@#$TMP5
3050 016660  104124                 1$:     ERROR   +124            ;BAD DATA X11*0 ST 3127
3051 016662  000415                         BR      KKBDONE
3052
3053                                ;REPORT R0 INCORRECT:
3054 016664  012737  016563  001240 KKB20:  MOV     #KKBBF1-7,@#$TMP3
3055 016672  010037  001242                 MOV     R0,@#$TMP4
3056 016676  104125                 1$:     ERROR   +125            ;R0 BADX
3057 016700  000406                         BR      KKBDONE
3058                                ;REPORT FPS INCORRECT:
3059 016702  010537  001240         KKB25:  MOV     R5,@#$TMP3
3060 016706  012737  000204  001244         MOV     #204,@#$TMP5
3061 016714  104126                 1$:     ERROR   +126            ;FPSX
3062
3063 016716                         KKBDONE:
     016716  104412                         RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                             ;THE USER TYPED CONTROL G?).
3064                                    ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 6, GR7
3065                                ;;**************************************************************
                                    ;*TEST 30       SEE ABOVE COMMENT FOR TEST TITLE
                                    ;*
                                    ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                    ;*SOURCE FLOWS.  THE NEGD INSTRUCTION
                                    ;*IS USED TO TEST MODE 6
                                    ;*
                                    ;;**************************************************************
     016720  000004                 TST30:  SCOPE
3066 016722                         LLB1:
     016722  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3067 016724  012700  017022                 MOV     #LLBTP1,R0              ;SET UP THE DATA BUFFER.
3068 016730  012701  017042                 MOV     #LLBBF0,R1
3069 016734  012702  000004                 MOV     #4,R2
3070 016740  012021                 1$:     MOV     (R0)+,(R1)+
3071 016742  077202                         SOB     R2,1$
3072 016744  012700  000200                 MOV     #200,R0         ;SET FD.
3073 016750  170100                         LDFPS   R0
```

```
3074 016752 012737 016766 001236          MOV     #LLB2,@#$TMP2
3075 016760 012737 017062 000004          MOV     #LLB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
3076
3077 016766 170767 000050          LLB2:  NEGD    LLBBF0           ;TEST INSTRUCTION.
3078
3079 016772 170205                        STFPS   R5               ;GET FPS.
3080 016774 012701 017042                 MOV     #LLBBF0,R1               ;CHECK RESULT.
3081 017000 012702 000004                 MOV     #4,R2
3082 017004 005721                 1$:    TST     (R1)+
3083 017006 001043                        BNE     LLB15            ;BRANCH IF INCORRECT.
3084 017010 077203                        SOB     R2,1$
3085 017012 022705 000204                 CMP     #204,R5          ;IS THE FPS CORRECT?
3086 017016 001052                        BNE     LLB25            ;BRANCH IF INCORRECT.
3087 017020 000457                        BR      LLBDONE
3088
3089                               ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
3090 017022 000127                 LLBTP1: 127
3091 017024 137475                         137475
3092 017026 147372                         147372
3093 017030 117057                         117057
3094 017032 000000                 LLBTP2: 0
3095 017034 000000                         0
3096 017036 000000                         0
3097 017040 000000                         0
3098 017042 177777                 LLBBF0: -1
3099 017044 177777                         -1
3100 017046 177777                         -1
3101 017050 177777                         -1
3102 017052 177777                 LLBBF1: -1
3103 017054 177777                         -1
3104 017056 177777                         -1
3105 017060 177777                         -1
3106
3107                               ;IF A TRAP TO 4 OCCURS COME HERE.
3108 017062 011602                 LLB10: MOV     (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
3109 017064 020227 016770                 CMP     R2,#LLB2+2
3110 017070 001405                        BEQ     1$               ;BRANCH IF YES.
3111 017072 020227 016772                 CMP     R2,#LLB2+4
3112 017076 001402                        BEQ     1$               ;BRANCH IF YES.
3113 017100 000137 046250                 JMP     @#CPSPUR         ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3114                               ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3115 017104 022626                 1$:    CMP     (SP)+,(SP)+
3116 017106 010237 001236                 MOV     R2,@#$TMP2
3117 017112 104127                 2$:    ERROR   +127             ;ODD ADRES
3118 017114 000421                        BR      LLBDONE          ;BUT FDSTX IN ST 771
3119
3120                               ;REPORT RESULT INCORRECT:
3121 017116 012737 017032 001240 LLB15:  MOV     #LLBTP2,@#$TMP3
3122 017124 012737 017022 001242          MOV     #LLBTP1,@#$TMP4
3123 017132 012737 017042 001244          MOV     #LLBBF0,@#$TMP5
3124 017140 104130                 1$:    ERROR   +130             ;BAD DATA X11*0 ST 3127
3125 017142 000406                        BR      LLBDONE
3126                               ;REPORT FPS INCORRECT:
3127 017144 010537 001240          LLB25: MOV     R5,@#$TMP3
3128 017150 012737 000204 001244          MOV     #204,@#$TMP5
3129 017156 104131                 1$:    ERROR   +131             ;FPSX
3130
```

```
3131 017160                          LLBDONE:
     017160  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                             ;THE USER TYPED CONTROL G?).
3132                                         ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 7, GR7
3133                                 ;;**************************************************************
                                     ;*TEST 31       SEE ABOVE COMMENT FOR TEST TITLE
                                     ;*
                                     ;*THIS IS A TEST THE NEGF,  ABSF AND TSTF
                                     ;*SOURCE FLOWS.  THE ABSD INSTRUCTION
                                     ;*IS USED TO TEST MODE 7
                                     ;*
                                     ;**************************************************************
     017162  000004                  TST31:  SCOPE
3134
3135 017164                          MMB1:
     017164  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3136 017166  012700  017264                  MOV     #MMBTP1,R0              ;SET UP THE DATA BUFFER.
3137 017172  012701  017314                  MOV     #MMBBF0,R1
3138 017176  012702  000010                  MOV     #10,R2
3139 017202  012021                  1$:     MOV     (R0)+,(R1)+
3140 017204  077202                          SOB     R2,1$
3141 017206  012700  000200                  MOV     #200,R0         ;SET FD.
3142 017212  170100                          LDFPS   R0
3143 017214  012737  017230  001236          MOV     #MMB2,@#$TMP2
3144 017222  012737  017334  000004          MOV     #MMB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
3145
3146 017230  170677  000070          MMB2:   ABSD    @MMBBF1         ;TEST INSTRUCTION.
3147
3148 017234  170205                          STFPS   R5              ;GET FPS.
3149 017236  012701  017314                  MOV     #MMBBF0,R1              ;CHECK RESULT.
3150 017242  012702  000004                  MOV     #4,R2
3151 017246  005721                  1$:     TST     (R1)+
3152 017250  001047                          BNE     MMB15           ;BRANCH IF INCORRECT.
3153 017252  077203                          SOB     R2,1$
3154 017254  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
3155 017260  001056                          BNE     MMB25           ;BRANCH IF INCORRECT.
3156 017262  000463                          BR      MMBDONE
3157
3158                                 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
3159 017264  000137                  MMBTP1: 137
3160 017266  045607                          045607
3161 017270  101230                          101230
3162 017272  045607  017314  177777          45607,MMBBF0,-1,-1,-1
     017300  177777  177777
3163 017304  000000                  MMBTP2: 0
3164 017306  000000                          0
3165 017310  000000                          0
3166 017312  000000                          0
3167 017314  177777                  MMBBF0: -1
3168 017316  177777                          -1
3169 017320  177777                          -1
3170 017322  177777                          -1
3171 017324  177777                  MMBBF1: -1
3172 017326  177777                          -1
```

```
3173 017330  177777                        -1
3174 017332  177777                        -1
3175
3176                                ;IF A TRAP TO 4 OCCURS COME HERE.
3177 017334  011602                MMB10:  MOV     (SP),R2         ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
3178 017336  020227  017232                CMP     R2,#MMB2+2
3179 017342  001405                        BEQ     1$              ;BRANCH IF YES.
3180 017344  020227  017234                CMP     R2,#MMB2+4
3181 017350  001402                        BEQ     1$              ;BRANCH IF YES.
3182 017352  000137  046250                JMP     @#CPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3183                                ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3184 017356  022626                1$:     CMP     (SP)+,(SP)+
3185 017360  010237  001236                MOV     R2,@#$TMP2
3186 017364  104132                2$:     ERROR   +132            ;ODD ADRES
3187 017366  000421                        BR      MMBDONE         ;BUT FDSTX IN ST 771
3188
3189                                ;REPORT RESULT INCORRECT:
3190 017370  012737  017304  001240 MMB15:  MOV     #MMBTP2,@#$TMP3
3191 017376  012737  017264  001242        MOV     #MMBTP1,@#$TMP4
3192 017404  012737  017314  001244        MOV     #MMBBF0,@#$TMP5
3193 017412  104133                1$:     ERROR   +133            ;BAD DATA X11*0 ST 3127
3194 017414  000406                        BR      MMBDONE
3195                                ;REPORT FPS INCORRECT:
3196 017416  010537  001240        MMB25:  MOV     R5,@#$TMP3
3197 017422  012737  000204  001244        MOV     #204,@#$TMP5
3198 017430  104134                1$:     ERROR   +134            ;FPSX
3199
3200 017432                         MMBDONE:
     017432  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                            ;SEE IF THE USER HAS EXPRESSED
                                            ;THE DESIRE TO CHANGE THE SOFTWARE
                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                            ;THE USER TYPED CONTROL G?).
3207                                ;;*****************************************************************
                                    ;*TEST 32       SPECIAL DEST, MODE 0, TEST
                                    ;*
                                    ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                    ;*MODE 0 USING THE NEGD INSTR.
                                    ;*
                                    ;;*****************************************************************
     017434  000004                 TST32:  SCOPE
3208
3209 017436                         NNB1:
     017436  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3210 017440  012700  000200                MOV     #200,R0         ;SET FD.
3211 017444  170100                        LDFPS   R0
3212 017446  012700  017534                MOV     #NNBTP1,R0              ;SET UP AC0.
3213 017452  172410                        LDD     (R0),AC0
3214 017454  012737  017462  001236        MOV     #NNB2,@#$TMP2
3215
3216 017462  170700                 NNB2:  NEGD    AC0             ;TEST INSTRUCTION.
3217
3218 017464  170205                        STFPS   R5              ;GET FPS.
3219 017466  012700  000200                MOV     #200,R0         ;SET FD.
3220 017472  170100                        LDFPS   R0
3221 017474  012700  017554                MOV     #NNBBF0,R0              ;GET THE RESULT.
3222 017500  174010                        STD     AC0,(R0)
```

```
3223 017502  012700  017554                    MOV     #NNBBF0,R0              ;IS THE RESULT CORRECT?
3224 017506  012701  017544                    MOV     #NNBTP2,R1
3225 017512  012702  000004                    MOV     #4,R2
3226 017516  022021                    1$:     CMP     (R0)+,(R1)+
3227 017520  001021                            BNE     NNB10           ;BRANCH IF INCORRECT.
3228 017522  077203                            SOB     R2,1$
3229 017524  022705  000210                    CMP     #210,R5         ;IS THE FPS CORRECT?
3230 017530  001033                            BNE     NNB15           ;BRANCH IF INCORRECT.
3231 017532  000440                            BR      NNBDONE
3232
3233                                    ;THESE ARE DATA TABLES AND A DATA BUFFER.
3234 017534  013572                    NNBTP1: 013572
3235 017536  046013                            46013
3236 017540  057246                            57246
3237 017542  013570                            013570
3238 017544  113572                    NNBTP2: 113572
3239 017546  046013                            46013
3240 017550  057246                            57246
3241 017552  013570                            013570
3242 017554  000000                    NNBBF0: 0
3243 017556  000000                            0
3244 017560  000000                            0
3245 017562  000000                            0
3246
3247                                    ;REPORT RESULT INCORRECT:
3248 017564  012737  017554  001240    NNB10:  MOV     #NNBBF0,@#$TMP3
3249 017572  012737  017544  001242            MOV     #NNBTP2,@#$TMP4
3250 017600  023737  017534  017554            CMP     @#NNBTP1,@#NNBBF0
3251 017606  001002                            BNE     NNB11
3252 017610  104135                    1$:     ERROR   +135            ;E10*200X ST 336
3253 017612  000410                            BR      NNBDONE
3254
3255                                    ;REPORT RESULT INCORRECT:
3256 017614                            NNB11:
3257 017614  104136                    1$:     ERROR   +136            ;BAD DATA NEGF
3258 017616  000406                            BR      NNBDONE
3259
3260                                    ;REPORT FPS INCORRECT:
3261 017620  010537  001242            NNB15:  MOV     R5,@#$TMP4
3262 017624  012737  000210  001240            MOV     #210,@#$TMP3
3263 017632  104137                    1$:     ERROR   +137            ;FPSX
3264
3265 017634                            NNBDONE:
     017634  104412                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
3266                                    ;;**********************************************************************
                                        ;*TEST 33       SPECIAL DEST, MODE 1, TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                        ;*MODE 1 USING THE NEGD INSTR.
                                        ;*
                                        ;;**********************************************************************
     017636  000004                     TST33:  SCOPE
3267
```

```
3268 017640                          00B1:
     017640  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3269 017642  012701  017752                 MOV     #00BTP1,R1      ;SET UP THE DATA BUFFER.
3270 017646  012700  017762                 MOV     #00BTP2,R0
3271 017652  012702  000004                 MOV     #4,R2
3272 017656  012021                  1$:    MOV     (R0)+,(R1)+
3273 017660  077202                         SOB     R2,1$
3274 017662  012700  017752                 MOV     #00BTP1,R0
3275 017666  042710  100000                 BIC     #100000,(R0)    ;MAKE OPERAND POSITIVE.
3276 017672  012737  017706  001236         MOV     #00B2,@#$TMP2
3277 017700  012701  000200                 MOV     #200,R1         ;SET FD.
3278 017704  170101                         LDFPS   R1
3279
3280 017706  170710                  00B2:  NEGD    (R0)            ;TEST INSTRUCTION.
3281 017710  170205                         STFPS   R5              ;GET FPS.
3282 017712  012701  017752                 MOV     #00BTP1,R1              ;IS THE RESULT CORRECT.
3283 017716  012702  017762                 MOV     #00BTP2,R2
3284 017722  012703  000004                 MOV     #4,R3
3285 017726  022122                  1$:    CMP     (R1)+,(R2)+
3286 017730  001020                         BNE     00B10           ;BRANCH IF INCORRECT.
3287 017732  077303                         SOB     R3,1$
3288 017734  022700  017752                 CMP     #00BTP1,R0      ;IS R0 CORRECT.
3289 017740  001024                         BNE     00B15           ;BRANCH IF INCORRECT.
3290 017742  022705  000210                 CMP     #210,R5         ;IS THE FPS CORRECT?
3291 017746  001030                         BNE     00B20           ;BRANCH IF INCORRECT.
3292 017750  000435                         BR      00BDONE
3293
3294                                 ;THESE ARE DATA TABLES AND A DATA BUFFER.
3295 017752  023245                  00BTP1: 023245
3296 017754  026720                          26720
3297 017756  122324                          122324
3298 017760  052672                          52672
3299 017762  123245                  00BTP2: 123245
3300 017764  026720                          26720
3301 017766  122324                          122324
3302 017770  052672                          52672
3303
3304                                 ;REPORT RESULT INCORRECT:
3305 017772  012737  017752  001240  00B10:  MOV     #00BTP1,@#$TMP3
3306 020000  012737  017762  001242          MOV     #00BTP2,@#$TMP4
3307 020006  104140                  1$:    ERROR   +140            ;BAD DATA
3308 020010  000415                         BR      00BDONE
3309
3310                                 ;REPORT R0 INCORRECT:
3311 020012  012737  017752  001240  00B15:  MOV     #00BTP1,@#$TMP3
3312 020020  010037  001242                  MOV     R0,@#$TMP4
3313 020024  104141                  1$:    ERROR   +141            ;SPEC DESTX
3314 020026  000406                         BR      00BDONE         ;R0X
3315
3316                                 ;REPORT FPS INCORRECT:
3317 020030  012737  000210  001240  00B20:  MOV     #210,@#$TMP3
3318 020036  010537  001242                  MOV     R5,@#$TMP4
3319 020042  104142                  1$:    ERROR   +142
3320
3321 020044                          00BDONE:
     020044  104412                         RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
```

```
                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                        ;THE USER TYPED CONTROL G?).
3322                                    ;;**************************************************************
                                        ;*TEST 34       SPECIAL DEST, MODE 2, TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                        ;*MODE 2 USING THE NEGD INSTR.
                                        ;*
                                        ;;**************************************************************
020046  000004                          TST34:  SCOPE
3323 020050
020050  104413                          PPB1:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3324
3325 020052  012701  020162                    MOV     #PPBTP1,R1      ;SET UP THE DATA BUFFER.
3326 020056  012700  020172                    MOV     #PPBTP2,R0
3327 020062  012702  000004                    MOV     #4,R2
3328 020066  012021                    1$:     MOV     (R0)+,(R1)+
3329 020070  077202                            SOB     R2,1$
3330 020072  012700  020162                    MOV     #PPBTP1,R0
3331 020076  042710  100000                    BIC     #100000,(R0)    ;MAKE OPERAND POSITIVE.
3332 020102  012737  020116  001236           MOV     #PPB2,@#$TMP2
3333 020110  012701  000200                    MOV     #200,R1         ;SET FD.
3334 020114  170101                            LDFPS   R1
3335
3336 020116  170720                    PPB2:   NEGD    (R0)+           ;TEST INSTRUCTION.
3337
3338 020120  170205                            STFPS   R5              ;GET FPS.
3339 020122  012701  020162                    MOV     #PPBTP1,R1              ;IS THE RESULT CORRECT.
3340 020126  012702  020172                    MOV     #PPBTP2,R2
3341 020132  012703  000004                    MOV     #4,R3
3342 020136  022122                    1$:     CMP     (R1)+,(R2)+
3343 020140  001020                            BNE     PPB10           ;BRANCH IF INCORRECT.
3344 020142  077303                            SOB     R3,1$
3345 020144  022700  020172                    CMP     #PPBTP1+10,R0   ;IS R0 CORRECT.
3346 020150  001024                            BNE     PPB15           ;BRANCH IF INCORRECT.
3347 020152  022705  000210                    CMP     #210,R5         ;IS THE FPS CORRECT?
3348 020156  001030                            BNE     PPB20           ;BRANCH IF INCORRECT.
3349 020160  000435                            BR      PPBDONE
3350
3351                                    ;THESE ARE DATA TABLES AND A DATA BUFFER.
3352 020162  023245                    PPBTP1: 023245
3353 020164  026720                            26720
3354 020166  122324                            122324
3355 020170  052672                            52672
3356 020172  123245                    PPBTP2: 123245
3357 020174  026720                            26720
3358 020176  122324                            122324
3359 020200  052672                            52672
3360
3361                                    ;REPORT RESULT INCORRECT:
3362 020202  012737  020162  001240     PPB10:  MOV     #PPBTP1,@#$TMP3
3363 020210  012737  020172  001242             MOV     #PPBTP2,@#$TMP4
3364 020216  104143                    1$:     ERROR   +143            ;BAD DATA
3365 020220  000415                            BR      PPBDONE
3366
3367                                    ;REPORT R0 INCORRECT:
```

```
 3368 020222  012737  020172  001240  PPB15:  MOV     #PPBTP1+10,@#$TMP3
 3369 020230  010037  001242                  MOV     R0,@#$TMP4
 3370 020234  104144                  1$:     ERROR   +144                ;SPEC DESTX ROX
 3371 020236  000406                          BR      PPBDONE
 3372
 3373                                 ;REPORT FPS INCORRECT:
 3374 020240  012737  000210  001240  PPB20:  MOV     #210,@#$TMP3
 3375 020246  010537  001242                  MOV     R5,@#$TMP4
 3376 020252  104145                  1$:     ERROR   +145
 3377
 3378 020254                          PPBDONE:
      020254  104412                          RSETUP                      ;GO INITIALIZE THE FPS AND STACK; AND
                                                                          ;SEE IF THE USER HAS EXPRESSED
                                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                          ;THE USER TYPED CONTROL G?).
 3379                                 ;;*****************************************************************
                                      ;*TEST 35       SPECIAL DEST, MODE 4, TEST
                                      ;*
                                      ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                      ;*MODE 4 USING THE NEGD INSTR.
                                      ;*
                                      ;;*****************************************************************
      020256  000004                  TST35:  SCOPE
 3380 020260                          QQB1:
      020260  104413                          LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
 3381 020262  012701  020374                  MOV     #QQBTP1,R1          ;SET UP THE DATA BUFFER.
 3382 020266  012700  020414                  MOV     #QQBTP2,R0
 3383 020272  012702  000004                  MOV     #4,R2
 3384 020276  012021                  1$:     MOV     (R0)+,(R1)+
 3385 020300  077202                          SOB     R2,1$
 3386 020302  012700  020404                  MOV     #QQBTP1+10,R0
 3387 020306  042760  100000  177770          BIC     #100000,-10(R0) ;MAKE OPERAND POSITIVE.
 3388 020314  012737  020330  001236          MOV     #QQB2,@#$TMP2
 3389 020322  012701  000200                  MOV     #200,R1             ;SET FD.
 3390 020326  170101                          LDFPS   R1
 3391
 3392 020330  170740                  QQB2:   NEGD    -(R0)               ;TEST INSTRUCTION.
 3393
 3394 020332  170205                          STFPS   R5                  ;GET FPS.
 3395 020334  012701  020374                  MOV     #QQBTP1,R1                  ;IS THE RESULT CORRECT.
 3396 020340  012702  020414                  MOV     #QQBTP2,R2
 3397 020344  012703  000004                  MOV     #4,R3
 3398 020350  022122                  1$:     CMP     (R1)+,(R2)+
 3399 020352  001024                          BNE     QQB10               ;BRANCH IF INCORRECT.
 3400 020354  077303                          SOB     R3,1$
 3401 020356  022700  020374                  CMP     #QQBTP1,R0          ;IS R0 CORRECT.
 3402 020362  001030                          BNE     QQB15               ;BRANCH IF INCORRECT.
 3403 020364  022705  000210                  CMP     #210,R5             ;IS THE FPS CORRECT?
 3404 020370  001034                          BNE     QQB20               ;BRANCH IF INCORRECT.
 3405 020372  000441                          BR      QQBDONE
 3406
 3407                                 ;THESE ARE DATA TABLES AND A DATA BUFFER.
 3408 020374  023245                  QQBTP1: 023245
 3409 020376  026720                          26720
 3410 020400  122324                          122324
 3411 020402  052672                          52672
```

```
3412 020404  177777  177777  177777          .WORD   -1,-1,-1,-1
     020412  177777
3413 020414  123245                  QQBTP2: 123245
3414 020416  026720                          26720
3415 020420  122324                          122324
3416 020422  052672                          52672
3417
3418                                 ;REPORT RESULT INCORRECT:
3419 020424  012737  020374  001240  QQB10:  MOV     #QQBTP1,@#$TMP3
3420 020432  012737  020414  001242          MOV     #QQBTP2,@#$TMP4
3421 020440  104146                  1$:     ERROR   +146            ;BAD DATA
3422 020442  000415                          BR      QQBDONE
3423
3424                                 ;REPORT R0 INCORRECT:
3425 020444  012737  020374  001240  QQB15:  MOV     #QQBTP1,@#$TMP3
3426 020452  010037  001242                  MOV     R0,@#$TMP4
3427 020456  104147                  1$:     ERROR   +147            ;SPEC DESTX R0X
3428 020460  000406                          BR      QQBDONE
3429
3430
3431                                 ;REPORT FPS INCORRECT:
3432 020462  012737  000210  001240  QQB20:  MOV     #210,@#$TMP3
3433 020470  010537  001242                  MOV     R5,@#$TMP4
3434 020474  104150                  1$:     ERROR   +150
3435
3436 020476                          QQBDONE:
     020476  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                             ;THE USER TYPED CONTROL G?).
3437
3438                                 ;;*****************************************************************
                                     ;*TEST 36       SPECIAL DEST, MODE 3, TEST
                                     ;*
                                     ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                     ;*MODE 3 USING THE NEGD INSTR.
                                     ;*
                                     ;;*****************************************************************
     020500  000004                  TST36:  SCOPE
3439
3440 020502                          RRB1:
     020502  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3441 020504  012701  020622                  MOV     #RRBTP1,R1      ;SET UP THE DATA BUFFER.
3442 020510  012700  020632                  MOV     #RRBTP2,R0
3443 020514  012702  000004                  MOV     #4,R2
3444 020520  012021                  1$:     MOV     (R0)+,(R1)+
3445 020522  077202                          SOB     R2,1$
3446 020524  012700  020642                  MOV     #RRBTP3,R0
3447 020530  012710  020622                  MOV     #RRBTP1,(R0)
3448 020534  042737  100000  020622          BIC     #100000,@#RRBTP1        ;MAKE THE OPERAND POSITIVE.
3449 020542  012737  020556  001236          MOV     #RRB2,@#$TMP2
3450 020550  012701  000200                  MOV     #200,R1         ;SET FD.
3451 020554  170101                          LDFPS   R1
3452
3453 020556  170730                  RRB2:   NEGD    @(R0)+          ;TEST INSTRUCTION.
3454
```

```
3455 020560  170205                        STFPS   R5              ;GET FPS.
3456 020562  012701  020622                MOV     #RRBTP1,R1              ;IS THE RESULT CORRECT.
3457 020566  012702  020632                MOV     #RRBTP2,R2
3458 020572  012703  000004                MOV     #4,R3
3459 020576  022122                1$:     CMP     (R1)+,(R2)+
3460 020600  001021                        BNE     RRB10           ;BRANCH IF INCORRECT.
3461 020602  077303                        SOB     R3,1$
3462 020604  022700  020644                CMP     #RRBTP3+2,R0    ;IS R0 CORRECT.
3463 020610  001025                        BNE     RRB15           ;BRANCH IF INCORRECT.
3464 020612  022705  000210                CMP     #210,R5         ;IS THE FPS CORRECT?
3465 020616  001031                        BNE     RRB20           ;BRANCH IF INCORRECT.
3466 020620  000436                        BR      RRBDONE
3467
3468                               ;THESE ARE DATA TABLES AND A DATA BUFFER.
3469 020622  023245                RRBTP1: 023245
3470 020624  026720                        26720
3471 020626  122324                        122324
3472 020630  052672                        52672
3473 020632  123245                RRBTP2: 123245
3474 020634  026720                        26720
3475 020636  123324                        123324
3476 020640  052672                        52672
3477 020642  020622                RRBTP3: RRBTP1
3478
3479                               ;REPORT RESULT INCORRECT:
3480 020644  012737  020622  001240 RRB10:  MOV     #RRBTP1,@#$TMP3
3481 020652  012737  020632  001242        MOV     #RRBTP2,@#$TMP4
3482 020660  104150                1$:     ERROR   +150            ;BAD DATA
3483 020662  000415                        BR      RRBDONE
3484
3485                               ;REPORT R0 INCORRECT:
3486 020664  012737  020644  001240 RRB15:  MOV     #RRBTP3+2,@#$TMP3
3487 020672  010037  001242                MOV     R0,@#$TMP4
3488 020676  104152                1$:     ERROR   +152            ;SPEC DESTX R0X
3489 020700  000406                        BR      RRBDONE
3490
3491                               ;REPORT FPS INCORRECT:
3492 020702  012737  000210  001240 RRB20:  MOV     #210,@#$TMP3
3493 020710  010537  001242                MOV     R5,@#$TMP4
3494 020714  104153                1$:     ERROR   +153
3495
3496 020716                        RRBDONE:
     020716  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
3497
3498                               ;;*********************************************************************
                                   ;*TEST 37       SPECIAL DEST, MODE 5, TEST
                                   ;*
                                   ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                   ;*MODE 5 USING THE NEGD INSTR.
                                   ;*
                                   ;;*********************************************************************
     020720  000004                TST37:  SCOPE
3499 020722                        SSB1:
```

```
                020722  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
     3500       020724  012701  021044                  MOV     #SSBTP1,R1      ;SET UP THE DATA BUFFER.
     3501       020730  012700  021054                  MOV     #SSBTP2,R0
     3502       020734  012702  000004                  MOV     #4,R2
     3503       020740  012021                  1$:     MOV     (R0)+,(R1)+
     3504       020742  077202                          SOB     R2,1$
     3505       020744  012700  021066                  MOV     #SSBTP3+2,R0
     3506       020750  012760  021044  177776          MOV     #SSBTP1,-2(R0)
     3507       020756  042737  100000  021044          BIC     #100000,@#SSBTP1        ;MAKE THE OPERAND POSITIVE.
     3508       020764  012737  021000  001236          MOV     #SSB2,@#$TMP2
     3509       020772  012701  000200                  MOV     #200,R1         ;SET FD.
     3510       020776  170101                          LDFPS   R1
     3511
     3512       021000  170750                  SSB2:   NEGD    @-(R0)          ;TEST INSTRUCTION.
     3513
     3514       021002  170205                          STFPS   R5              ;GET FPS.
     3515       021004  012701  021044                  MOV     #SSBTP1,R1              ;IS THE RESULT CORRECT.
     3516       021010  012702  021054                  MOV     #SSBTP2,R2
     3517       021014  012703  000004                  MOV     #4,R3
     3518       021020  022122                  1$:     CMP     (R1)+,(R2)+
     3519       021022  001021                          BNE     SSB10           ;BRANCH IF INCORRECT.
     3520       021024  077303                          SOB     R3,1$
     3521       021026  022700  021064                  CMP     #SSBTP3,R0      ;IS R0 CORRECT.
     3522       021032  001025                          BNE     SSB15           ;BRANCH IF INCORRECT.
     3523       021034  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
     3524       021040  001031                          BNE     SSB20           ;BRANCH IF INCORRECT.
     3525       021042  000436                          BR      SSBDONE
     3526
     3527                                          ;THESE ARE DATA TABLES AND A DATA BUFFER.
     3528       021044  023245                  SSBTP1: 023245
     3529       021046  026720                          26720
     3530       021050  122324                          122324
     3531       021052  052672                          52672
     3532       021054  123245                  SSBTP2: 123245
     3533       021056  026270                          26270
     3534       021060  122324                          122324
     3535       021062  052672                          52672
     3536       021064  021044                  SSBTP3: SSBTP1
     3537
     3538                                          ;REPORT RESULT INCORRECT:
     3539       021066  012737  021044  001240  SSB10:  MOV     #SSBTP1,@#$TMP3
     3540       021074  012737  021054  001242          MOV     #SSBTP2,@#$TMP4
     3541       021102  104154                  1$:     ERROR   +154            ;BAD DATA
     3542       021104  000415                          BR      SSBDONE
     3543
     3544                                          ;REPORT R0 INCORRECT:
     3545       021106  012737  021064  001240  SSB15:  MOV     #SSBTP3,@#$TMP3
     3546       021114  010037  001242                  MOV     R0,@#$TMP4
     3547       021120  104155                  1$:     ERROR   +155            ;SPEC DESTX R0X
     3548       021122  000406                          BR      SSBDONE
     3549
     3550                                          ;REPORT FPS INCORRECT:
     3551       021124  012737  000210  001240  SSB20:  MOV     #210,@#$TMP3
     3552       021132  010537  001242                  MOV     R5,@#$TMP4
     3553       021136  104156                  1$:     ERROR   +156
     3554
     3555       021140                          SSBDONE:
```

```
       021140  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                       ;SEE IF THE USER HAS EXPRESSED
                                                                       ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                       ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                       ;THE USER TYPED CONTROL G?).
  3556                                 ;;*********************************************************************
                                       ;*TEST 40       SPECIAL DEST, FLOATING MODE 2, TEST
                                       ;*
                                       ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                       ;*MODE 2 USING THE NEGF INSTR.
                                       ;*
                                       ;;*********************************************************************
       021142  000004                  TST40:  SCOPE
  3557 021144                          TTB1:
       021144  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
  3558 021146  012701  021256                  MOV     #TTBTP1,R1      ;SET UP THE DATA BUFFER.
  3559 021152  012700  021266                  MOV     #TTBTP2,R0
  3560 021156  012702  000004                  MOV     #4,R2
  3561 021162  012021                  1$:     MOV     (R0)+,(R1)+
  3562 021164  077202                          SOB     R2,1$
  3563 021166  012700  021256                  MOV     #TTBTP1,R0
  3564 021172  042710  100000                  BIC     #100000,(R0)    ;MAKE OPERAND POSITIVE.
  3565 021176  012737  021212  001236          MOV     #TTB2,@#$TMP2
  3566 021204  012701  000000                  MOV     #000,R1         ;SET FD.
  3567 021210  170101                          LDFPS   R1
  3568
  3569 021212  170720                  TTB2:   NEGF    (R0)+           ;TEST INSTRUCTION.
  3570
  3571 021214  170205                          STFPS   R5              ;GET FPS.
  3572 021216  012701  021256                  MOV     #TTBTP1,R1              ;IS THE RESULT CORRECT.
  3573 021222  012702  021266                  MOV     #TTBTP2,R2
  3574 021226  012703  000004                  MOV     #4,R3
  3575 021232  022122                  1$:     CMP     (R1)+,(R2)+
  3576 021234  001020                          BNE     TTB10           ;BRANCH IF INCORRECT.
  3577 021236  077303                          SOB     R3,1$
  3578 021240  022700  021262                  CMP     #TTBTP1+4,R0    ;IS R0 CORRECT.
  3579 021244  001024                          BNE     TTB15           ;BRANCH IF INCORRECT.
  3580 021246  022705  000010                  CMP     #010,R5         ;IS THE FPS CORRECT?
  3581 021252  001030                          BNE     TTB20           ;BRANCH IF INCORRECT.
  3582 021254  000435                          BR      TTBDONE
  3583
  3584                                 ;THESE ARE DATA TABLES AND A DATA BUFFER.
  3585 021256  023245                  TTBTP1: 023245
  3586 021260  026720                          26720
  3587 021262  122324                          122324
  3588 021264  052672                          52672
  3589 021266  123245                  TTBTP2: 123245
  3590 021270  026720                          26720
  3591 021272  122324                          122324
  3592 021274  052672                          52672
  3593
  3594                                 ;REPORT RESULT INCORRECT:
  3595 021276  012737  021256  001240  TTB10:  MOV     #TTBTP1,@#$TMP3
  3596 021304  012737  021266  001242          MOV     #TTBTP2,@#$TMP4
  3597 021312  104150                  1$:     ERROR   +150            ;BAD DATA
  3598 021314  000415                          BR      TTBDONE
  3599
```

```
3600                                    ;REPORT R0 INCORRECT:
3601 021316  012737  021262  001240  TTB15:  MOV     #TTBTP1+4,@#$TMP3
3602 021324  010037  001242             MOV     R0,@#$TMP4
3603 021330  104160                1$:  ERROR   +160            ;SPEC DESTX R0X
3604 021332  000406                     BR      TTBDONE
3605
3606                                    ;REPORT FPS INCORRECT:
3607 021334  012737  000010  001240  TTB20:  MOV     #010,@#$TMP3
3608 021342  010537  001242             MOV     R5,@#$TMP4
3609 021346  104161                1$:  ERROR   +161
3610
3611 021350                         TTBDONE:
     021350  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
3612                                    ;TEST TITLE:SPECIAL DEST, MODE2, GR7 (IMMEDIATE)
3613                                    ;;*****************************************************************
                                        ;*TEST 41       SEE ABOVE COMMENT FOR TEST TITLE
                                        ;*
                                        ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                        ;*MODE 2(IMMEDIATE) USING THE NEGD INSTR.
                                        ;*
                                        ;;*****************************************************************
     021352  000004                     TST41:  SCOPE
3614 021354                         UUB1:
     021354  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3615 021356  012700  021502             MOV     #UUBTP2,R0
3616 021362  012701  021430             MOV     #UUBTP1,R1          ;SET UP THE DATA BUFFER.
3617 021366  012702  000004             MOV     #4,R2
3618 021372  012021                1$:  MOV     (R0)+,(R1)+
3619 021374  077202                     SOB     R2,1$
3620 021376  012700  021430             MOV     #UUBTP1,R0
3621 021402  042737  100000  021430     BIC     #100000,@#UUBTP1        ;MAKE THE OPERAND POSITIVE.
3622 021410  012737  021426  001236     MOV     #UUB2,@#$TMP2
3623 021416  012701  000200             MOV     #200,R1         ;SET FD.
3624 021422  170101                     LDFPS   R1
3625 021424  005001                     CLR     R1
3626
3627 021426  170727                UUB2:   NEGD    (R7)+           ;TEST INSTRUCTION.
3628 021430  005201  005201  005201  UUBTP1: 5201,5201,5201,5201
     021436  005201
3629                                    ;NOTE THAT AFTER EXECUTING THIS INSTRUCTION R1 SHOULD CONTAIN 3.
3630 021440  170205                     STFPS   R5              ;GET FPS.
3631 021442  012703  021430             MOV     #UUBTP1,R3      ;IS THE RESULT CORRECT.
3632 021446  012702  021502             MOV     #UUBTP2,R2
3633 021452  012704  000004             MOV     #4,R4
3634 021456  022322                1$:  CMP     (R3)+,(R2)+
3635 021460  001014                     BNE     UUB10           ;BRANCH IF INCORRECT.
3636 021462  077403                     SOB     R4,1$
3637 021464  022701  000003             CMP     #3,R1           ;WAS R1 INCREMENTED CORRECTLY.
3638 021470  001027                     BNE     UUB15           ;BRANCH IF INCORRECT.
3639 021472  022705  000210             CMP     #210,R5         ;IS THE FPS CORRECT?
3640 021476  001015                     BNE     UUB20           ;BRANCH IF INCORRECT.
3641 021500  000436                     BR      UUBDONE
3642
```

```
                                        ;THESE ARE DATA TABLE.
021502  105201                  UUBTP2: 105201
021504  005201                          5201
021506  005201                          5201
021510  005201                          5201

                                        ;REPORT RESULT INCORRECT:
021512  012737  021430  001240  UUB10:  MOV     #UUBTP1,@#$TMP3
021520  012737  021502  001242          MOV     #UUBTP2,@#$TMP4
021526  104162                  1$:     ERROR   +162            ;BAD DATA
021530  000422                          BR      UUBDONE

                                        ;REPORT FPS INCORRECT:
021532  012737  000210  001240  UUB20:  MOV     #210,@#$TMP3
021540  010537  001242                  MOV     R5,@#$TMP4
021544  104163                  1$:     ERROR   +163            ;FPS
021546  000413                          BR      UUBDONE

                                        ;REPORT PC INCORRECTLY INCREMENTED DURING EXECUTION.
021550  162701  000003          UUB15:  SUB     #3,R1
021554  006301                          ASL     R1
021556  012702  021432                  MOV     #UUBTP1+2,R2
021562  010237  001240                  MOV     R2,@#$TMP3
021566  160102                          SUB     R1,R2
021570  010237  001242                  MOV     R2,@#$TMP4
021574  104164                  1$:     ERROR   +164            ;PC BAD CONSTAND B GR7X

021576                          UUBDONE:
021576  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).
                                ;;*****************************************************************
                                ;*TEST 42       SPECIAL DEST, MODE 6, TEST
                                ;*
                                ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                ;*MODE 6 USING THE NEGD INSTR.
                                ;*
                                ;;*****************************************************************
021600  000004                  TST42:  SCOPE
021602                          XXB1:
021602  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
021604  012701  021726                  MOV     #XXBTP1,R1      ;SET UP THE DATA BUFFER.
021610  012700  021736                  MOV     #XXBTP2,R0
021614  012702  000004                  MOV     #4,R2
021620  012021                  1$:     MOV     (R0)+,(R1)+
021622  077202                          SOB     R2,1$
021624  012700  014525                  MOV     #XXBTP1-5201,R0
021630  042737  100000  021726          BIC     #100000,@#XXBTP1;MAKE OPERAND POSITIVE.
021636  012737  021654  001236          MOV     #XXB2,@#$TMP2
021644  012701  000200                  MOV     #200,R1         ;SET FD.
021650  170101                          LDFPS   R1

021652  005001                          CLR     R1
021654  170760  005201          XXB2:   NEGD    5201(R0)                ;TEST INSTRUCTION.
```

```
3687
3688 021660  170205                    STFPS   R5              ;GET FPS.
3689 021662  005701                    TST     R1
```

```
 3691 021664 001030                         BNE     XXB25           ;WAS THE PC CORRECT AFTER EXECUTION?
 3692 021666 012701 021726                  MOV     #XXBTP1,R1              ;IS THE RESULT CORRECT.
 3693 021672 012702 021736                  MOV     #XXBTP2,R2
 3694 021676 012703 000004                  MOV     #4,R3
 3695 021702 022122                 1$:     CMP     (R1)+,(R2)+
 3696 021704 001030                         BNE     XXB10           ;BRANCH IF INCORRECT.
 3697 021706 077303                         SOB     R3,1$
 3698 021710 022700 014525                  CMP     #XXBTP1-5201,R0 ;IS R0 CORRECT.
 3699 021714 001034                         BNE     XXB15           ;BRANCH IF INCORRECT.
 3700 021716 022705 000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
 3701 021722 001040                         BNE     XXB20           ;BRANCH IF INCORRECT.
 3702 021724 000445                         BR      XXBDONE
 3703
 3704                               ;THESE ARE DATA TABLES AND A DATA BUFFER.
 3705 021726 023245                 XXBTP1: 023245
 3706 021730 026720                         26720
 3707 021732 122324                         122324
 3708 021734 052672                         52672
 3709 021736 123245                 XXBTP2: 123245
 3710 021740 026720                         26720
 3711 021742 122324                         122324
 3712 021744 052672                         52672
 3713
 3714
 3715                               ;REPORT PC INCORRECT AFTER EXECUTION.
 3716 021746 012737 021656 001242   XXB25:  MOV     #XXB2+2,@#$TMP4
 3717 021754 012737 021660 001240           MOV     #XXB2+4,@#$TMP3
 3718 021762 104215                 1$:     ERROR   +215            ;PC NOT INCREMENTED BY 2.
 3719 021764 000425                         BR      XXBDONE
 3720
 3721                               ;REPORT RESULT INCORRECT:
 3722 021766 012737 021726 001240   XXB10:  MOV     #XXBTP1,@#$TMP3
 3723 021774 012737 021736 001242           MOV     #XXBTP2,@#$TMP4
 3724 022002 104216                 1$:     ERROR   +216            ;BAD DATA
 3725 022004 000415                         BR      XXBDONE
 3726
 3727                               ;REPORT R0 INCORRECT:
 3728 022006 012737 014525 001240   XXB15:  MOV     #XXBTP1-5201,@#$TMP3
 3729 022014 010037 001242                  MOV     R0,@#$TMP4
 3730 022020 104217                 1$:     ERROR   +217            ;SPEC DESTX ROX
 3731 022022 000406                         BR      XXBDONE
 3732
 3733
 3734                               ;REPORT FPS INCORRECT:
 3735 022024 012737 000210 001240   XXB20:  MOV     #210,@#$TMP3
 3736 022032 010537 001242                  MOV     R5,@#$TMP4
 3737 022036 104220                 1$:     ERROR   +220
 3738
 3739 022040                        XXBDONE:
      022040 104412                         RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                             ;THE USER TYPED CONTROL G?).
 3740
 3741                               ;;*****************************************************************
                                    ;*TEST 43        SPECIAL DEST, MODE 7, TEST
```

```
                                ;*
                                ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
                                ;*MODE 7 USING THE NEGD INSTR.
                                ;*
                                ;;*****************************************************************
     022042  000004             TST43:  SCOPE
3742
3743 022044                     YYB1:
     022044  104413                     LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3744 022046  012701  022176             MOV     #YYBTP1,R1      ;SET UP THE DATA BUFFER.
3745 022052  012700  022206             MOV     #YYBTP2,R0
3746 022056  012702  000004             MOV     #4,R2
3747 022062  012021             1$:     MOV     (R0)+,(R1)+
3748 022064  077202                     SOB     R2,1$
3749 022066  012700  015015             MOV     #YYBTP3-5201,R0
3750 022072  012760  022176  005201     MOV     #YYBTP1,5201(R0)
3751 022100  042737  100000  022176     BIC     #100000,@#YYBTP1        ;MAKE THE OPERAND POSITIVE.
3752 022106  012737  022124  001236     MOV     #YYB2,@#$TMP2
3753 022114  012701  000200             MOV     #200,R1         ;SET FD.
3754 022120  170101                     LDFPS   R1
3755
3756 022122  005001                     CLR     R1
3757 022124  170770  005201     YYB2:   NEGD    @5201(R0)               ;TEST INSTRUCTION.
3758
3759 022130  170205                     STFPS   R5              ;GET FPS.
3760 022132  005701                     TST     R1              ;WAS THE PC CORRECT AFTER EXECUTION?
3761 022134  001031                     BNE     YYB25
3762 022136  012701  022176             MOV     #YYBTP1,R1              ;IS THE RESULT CORRECT.
3763 022142  012702  022206             MOV     #YYBTP2,R2
3764 022146  012703  000004             MOV     #4,R3
3765 022152  022122             1$:     CMP     (R1)+,(R2)+
3766 022154  001031                     BNE     YYB10           ;BRANCH IF INCORRECT.
3767 022156  077303                     SOB     R3,1$
3768 022160  022700  015015             CMP     #YYBTP3-5201,R0 ;IS R0 CORRECT.
3769 022164  001035                     BNE     YYB15           ;BRANCH IF INCORRECT.
3770 022166  022705  000210             CMP     #210,R5         ;IS THE FPS CORRECT?
3771 022172  001041                     BNE     YYB20           ;BRANCH IF INCORRECT.
3772 022174  000446                     BR      YYBDONE
3773
3774                            ;THESE ARE DATA TABLES AND A DATA BUFFER.
3775 022176  023245             YYBTP1: 023245
3776 022200  026720                     26720
3777 022202  122324                     122324
3778 022204  052672                     52672
3779 022206  123245             YYBTP2: 123245
3780 022210  026720                     26720
3781 022212  123324                     123324
3782 022214  052672                     52672
3783 022216  022176             YYBTP3: YYBTP1
3784
3785                            ;REPORT PC INCORRECT AFTER EXECUTION.
3786 022220  016737  177702  001242 YYB25:  MOV     YYB2+2,@#$TMP4
3787 022226  016737  177676  001240         MOV     YYB2+4,@#$TMP3
3788 022234  104221             1$:     ERROR   +221            ;PC NOT INCREMENTED BY 2.
3789 022236  000425                     BR      YYBDONE
3790
3791                            ;REPORT RESULT INCORRECT:
```

```
 3792 022240  012737  022176  001240  YYB10:  MOV     #YYBTP1,@#$TMP3
 3793 022246  012737  022206  001242          MOV     #YYBTP2,@#$TMP4
 3794 022254  104222                  1$:     ERROR   +222            ;BAD DATA
 3795 022256  000415                          BR      YYBDONE
 3796
 3797                                ;REPORT R0 INCORRECT:
 3798 022260  012737  015015  001240  YYB15:  MOV     #YYBTP3-5201,@#$TMP3
 3799 022266  010037  001242          MOV     R0,@#$TMP4
 3800 022272  104223                  1$:     ERROR   +223            ;SPEC DESTX R0X
 3801 022274  000406                          BR      YYBDONE
 3802
 3803                                ;REPORT FPS INCORRECT:
 3804 022276  012737  000210  001240  YYB20:  MOV     #210,@#$TMP3
 3805 022304  010537  001242          MOV     R5,@#$TMP4
 3806 022310  104224                  1$:     ERROR   +224
 3807
 3808 022312                        YYBDONE:
      022312  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                              ;SEE IF THE USER HAS EXPRESSED
                                              ;THE DESIRE TO CHANGE THE SOFTWARE
                                              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                              ;THE USER TYPED CONTROL G?).
 3814                               ;;*****************************************************************
                                    ;*TEST 44       NEGD, ABSD AND TSTD TEST
                                    ;*
                                    ;*THIS IS A TEST OF THE NEGD ABSD AND TSTD INSTRUCTIONS.
                                    ;*
                                    ;;*****************************************************************
      022314  000004                TST44:  SCOPE
 3815                               ;TEST NEGD WITH POS NONZERO OPERAND
 3816 022316                        WWB1:
      022316  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 3817 022320  004767  000634                  JSR     PC,NATSUB
 3818 022324  000000                  1$:     0                       ;FLAG=NEGD.
 3819 022326  016341                  2$:     16341                   ;OPERAND.
 3820 022330  055772                          55772
 3821 022332  021133                          21133
 3822 022334  055447                          55447
 3823 022336  116341                  3$:     116341                  ;RESULT.
 3824 022340  055772                          55772
 3825 022342  021133                          21133
 3826 022344  055447                          55447
 3827 022346  016341                  4$:     16341                   ;ERROR RES.
 3828 022350  055772                          55772
 3829 022352  021133                          21133
 3830 022354  055447                          55447
 3831 022356  000207                  5$:     207                     ;FPS BEFORE EXECUTION.
 3832 022360  000210                          210                     ;FPS AFTER EXECUTION.
 3833 022362  000200                          200                     ;ERROR FPS.
 3834 022364  177777                          -1                      ;FEC
 3835 022366  104200                  6$:     ERROR   +200            ;E10<---E10*200X ST 336
 3836 022370  000401                          BR      7$
 3837 022372  104201                          ERROR   +201            ;BUT ENBT ST 336X WENT TO 053 INTO 453
 3838 022374                        7$:
 3839                               ;TEST NEGD WITH NEG OPERAND.
 3840 022374                        WWB2:
      022374  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
3841 022376  004767  000556                    JSR     PC,NATSUB
3842 022402  000000                  1$:       0                       ;FLAG=NEGD.
3843 022404  152525                  2$:       152525                  ;OPERAND.
3844 022406  053545                            53545
3845 022410  055565                            55565
3846 022412  057505                            57505
3847 022414  052525                  3$:       52525                   ;RESULT.
3848 022416  053545                            53545
3849 022420  055565                            55565
3850 022422  057505                            57505
3851 022424  152525                  4$:       152525                  ;ERROR RES.
3852 022426  053545                            53545
3853 022430  055565                            55565
3854 022432  057505                            57505
3855 022434  000217                  5$:       217                     ;FPS BEFORE EXECUTION.
3856 022436  000200                            200                     ;FPS AFTER EXECUTION.
3857 022440  000210                            210                     ;ERROR FPS.
3858 022442  177777                            -1                      ;FEC
3859 022444  104200                  6$:       ERROR   +200            ;E10<---E10*200X S336
3860 022446  000401                            BR      7$
3861 022450  104202                            ERROR   +202            ;BUT ENBT X ST336 TO 453 INTO 053
3862 022452                         7$:
3863                                ;TEST ABSD WITH POSITIVE OPERAND
3864 022452                         WWB3:
     022452  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3865 022454  004767  000500                    JSR     PC,NATSUB
3866 022460  000001                  1$:       1                       ;FLAG=ABSD.
3867 022462  060705                  2$:       60705                   ;OPERAND.
3868 022464  124735                            124735
3869 022466  060124                            60124
3870 022470  073560                            73560
3871 022472  060705                  3$:       60705                   ;RESULT.
3872 022474  124735                            124735
3873 022476  060124                            60124
3874 022500  073560                            73560
3875 022502  160705                  4$:       160705                  ;ERROR RES.
3876 022504  124735                            124735
3877 022506  060124                            60124
3878 022510  073560                            73560
3879 022512  000217                  5$:       217                     ;FPS BEFORE EXECUTION.
3880 022514  000200                            200                     ;FPS AFTER EXECUTION.
3881 022516  000210                            210                     ;ERROR FPS.
3882 022520  177777                            -1                      ;EITHER BUT OP1B
3883 022522  104203                  6$:       ERROR   +203            ;BUT ST 055 TO 336 INTO 335
3884 022524  000401                            BR      7$
3885 022526  104203                            ERROR   +203            ;OR BUT ENBT ST 335 TO 452 INTO 052
3886 022530                         7$:
3887                                ;TEST ABSD WITH NEG. OPERAND
3888 022530                         WWB4:
     022530  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3889 022532  004767  000422                    JSR     PC,NATSUB
3890 022536  000001                  1$:       1                       ;FLAG=ABSD.
3891 022540  154345                  2$:       154345                  ;OPERAND.
3892 022542  076567                            76567
3893 022544  032123                            32123
3894 022546  043234                            43234
3895 022550  054345                  3$:       54345                   ;RESULT.
```

```
3896 022552  076567                          76567
3897 022554  032123                          32123
3898 022556  043234                          43234
3899 022560  154345                4$:       154345                    ;ERROR RES.
3900 022562  076567                          76567
3901 022564  032123                          32123
3902 022566  043234                          43234
3903 022570  000217                5$:       217                       ;FPS BEFORE EXECUTION.
3904 022572  000200                          200                       ;FPS AFTER EXECUTION.
3905 022574  177777                          -1                        ;ERROR FPS.
3906 022576  177777                          -1
3907 022600  104204                6$:       ERROR   +204              ;E10*E10*200X ST 452
3908 022602  000401                          BR      7$
3909 022604  104171                          ERROR   +171
3910 022606                        7$:
3911                               ;TEST WITH POSITIVE OP
3912 022606                        WWB5:
     022606  104413                          LPERR                     ;SET UP THE LOOP ON ERROR ADDRESS.
3913 022610  004767  000344                  JSR     PC,NATSUB
3914 022614  000002                1$:       2                         ;FLAG=TSTD.
3915 022616  012321                2$:       12321                     ;OPERAND.
3916 022620  045654                          45654
3917 022622  070107                          70107
3918 022624  034543                          34543
3919 022626  012321                3$:       12321                     ;RESULT.
3920 022630  045654                          45654
3921 022632  070107                          70107
3922 022634  034543                          34543
3923 022636  112321                4$:       112321                    ;ERROR RES.
3924 022640  045654                          45654
3925 022642  070107                          70107
3926 022644  034543                          34543
3927 022646  000217                5$:       217                       ;FPS BEFORE EXECUTION.
3928 022650  000200                          200                       ;FPS AFTER EXECUTION.
3929 022652  000210                          210                       ;ERROR FPS.
3930 022654  177777                          -1
3931 022656  104205                6$:       ERROR   +205              ;BUT (OP1B) X ST044 TO 336 INTO 334
3932 022660  000401                          BR      7$
3933 022662  104206                          ERROR   +206              ;BUT ENBT ST 334 TO 453 INTO 053
3934 022664                        7$:
3935                               ;TEST TSTD WITH NEG OP
3936 022664                        WWB6:
     022664  104413                          LPERR                     ;SET UP THE LOOP ON ERROR ADDRESS.
3937 022666  004767  000266                  JSR     PC,NATSUB
3938 022672  000002                1$:       2                         ;FLAG=TSTD.
3939 022674  123765                2$:       123765                    ;OPERAND.
3940 022676  023407                          23407
3941 022700  034510                          34510
3942 022702  045621                          45621
3943 022704  123765                3$:       123765                    ;RESULT.
3944 022706  023407                          23407
3945 022710  034510                          34510
3946 022712  045621                          45621
3947 022714  023765                4$:       23765                     ;ERROR RES.
3948 022716  023407                          23407
3949 022720  034510                          34510
3950 022722  045621                          45621
```

```
3951 022724  000207                5$:     207                     ;FPS BEFORE EXECUTION.
3952 022726  000210                        210                     ;FPS AFTER EXECUTION.
3953 022730  000200                        200                     ;ERROR FPS.
3954 022732  177777                        -1
3955 022734  104207                6$:     ERROR   +207            ;BUT OPB1 ST 055 TO 335 INTO 334
3956 022736  000401                        BR      7$
3957 022740  104210                        ERROR   +210            ;BUT ENBT ST 334 TO 053 INTO 453
3958 022742                        7$:
3959                               ;TEST TSTD 0 OP
3960 022742                        WWB7:
     022742  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3961 022744  004767  000210               JSR     PC,NATSUB
3962 022750  000002                1$:     2                       ;FLAG=TSTD.
3963 022752  000175                2$:     175                     ;OPERAND.
3964 022754  176737                        176737
3965 022756  071727                        71727
3966 022760  037574                        37574
3967 022762  000175                3$:     175                     ;RESULT.
3968 022764  176737                        176737
3969 022766  071727                        71727
3970 022770  037574                        37574
3971 022772  000000                4$:     0                       ;ERROR RES.
3972 022774  000000                        0
3973 022776  000000                        0
3974 023000  000000                        0
3975 023002  000200                5$:     200                     ;FPS BEFORE EXECUTION.
3976 023004  000204                        204                     ;FPS AFTER EXECUTION.
3977 023006  000214                        214                     ;ERROR FPS.
3978 023010  177777                        -1
3979 023012  104211                6$:     ERROR   +211            ;BUT OP1B ST 255 TO 311 OR 312 INTO 310
3980 023014  000401                        BR      7$
3981 023016  104212                        ERROR   +212            ;BUT ENBT ST 310 TO 402 INTO 002
3982 023020                        7$:
3983                               ;TEST TSTD -0 OP FIUV=0
3984 023020                        WWB8:
     023020  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3985 023022  004767  000132               JSR     PC,NATSUB
3986 023026  000002                1$:     2                       ;FLAG=TSTD.
3987 023030  100123                2$:     100123                  ;OPERAND.
3988 023032  021012                        21012
3989 023034  034565                        34565
3990 023036  043210                        43210
3991 023040  100123                3$:     100123                  ;RESULT.
3992 023042  021012                        21012
3993 023044  034565                        34565
3994 023046  043210                        43210
3995 023050  000000                4$:     0                       ;ERROR RES.
3996 023052  000000                        0
3997 023054  000000                        0
3998 023056  000000                        0
3999 023060  040203                5$:     40203                   ;FPS BEFORE EXECUTION.
4000 023062  040214                        040214                  ;FPS AFTER EXECUTION.
4001 023064  140214                        140214                  ;ERROR FPS.
4002 023066  177777                        -1
4003 023070  104211                6$:     ERROR   +211            ;+
4004 023072  000401                        BR      7$
4005 023074  104213                        ERROR   +213            ;BUT FIUV ST 257 TO 355 INTO 255
```

```
4006 023076                          7$:
4007                                 ;TEST TSTD -0 OP FIUV=1
4008 023076                          WWB9:
     023076  104413                         LPERR                ;SET UP THE LOOP ON ERROR ADDRESS.
4009 023100  004767  000054                 JSR     PC,NATSUB
4010 023104  000002                  1$:    2                    ;FLAG=TSTD.
4011 023106  100137                  2$:    100137               ;OPERAND.
4012 023110  024613                         24613
4013 023112  057024                         57024
4014 023114  060137                         60137
4015 023116  100137                  3$:    100137               ;RESULT.
4016 023120  024613                         24613
4017 023122  057024                         57024
4018 023124  060137                         60137
4019 023126  000000                  4$:    0                    ;ERROR RES.
4020 023130  000000                         0
4021 023132  000000                         0
4022 023134  000000                         0
4023 023136  044200                  5$:    44200                ;FPS BEFORE EXECUTION.
4024 023140  144214                         144214               ;FPS AFTER EXECUTION.
4025 023142  044214                         044214               ;ERROR FPS.
4026 023144  000014                         14
4027 023146  104211                  6$:    ERROR   +211         ;+
4028 023150  000401                         BR      7$
4029 023152  104214                         ERROR   +214         ;BUT FIUV ST 257 TO 255 INTO 355
4030 023154                          7$:
4031 023154  000167  000414                 JMP     WWBDONE
4032
4033                                 ;THIS SUBROUTINE, NATSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
4034                                 ;THE EITHER A TSTD, AN ABSD OR A NEGD INSTRUCTION AND CHECK THE RESULTS. A CALL
```

```
                                ;TO IT IS MADE THUS:
                                ;
                                ;       JSR     PC,@#NATSUB
                                ;FLAG:  .WORD   X               ;INSTRUCTION TYPE FLAG.
                                ;ACARG: .WORD   X,X,X,X         ;OPERAND
                                ;RES:   .WORD   X,X,X,X         ;EXPECTED RESULT
                                ;ERRES: .WORD   X,X,X,X         ;ERROR RESULT
                                ;FPSB:  .WORD   X               ;FPS BEFORE EXECUTION
                                ;FPSA:  .WORD   X               ;FPS AFTER EXECUTION
                                ;FEC:   .WORD   X               ;EXPECTED FEC
                                ;ERFPS: .WORD   X               ;ERROR FPS.
                                ;ERR1:  ERROR   +X              ;DATA ERROR.
                                ;       BR      CONT
                                ;ERR2:  ERROR   +X              ;FPS ERROR.
                                ;CONT:                          ;RETURN ADDRESS
                                ;
                                ;THE OPERAND IS SET UP IN NATBF1. THEN
                                ;THE EITHER THE TSTD, NEGD OR ABSD INSTRUCTION IS EXECUTED.
                                ;NATSUB USES THE FIRST OPERAND AS A FLAG TO DETERMINE WHICH INSTRUCTION
                                ;IS TO BE EXECUTED: 0 = NEGD, 1 = ABSD, 2 = TSTD.
                                ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA. IF THIS TOO IS CORRECT NATSUB RETURNS CONTROL
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD NATSUB
                                ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN NATSUB WILL RETURN
                                ;TO THE ERROR CALL AT ERR2, OTHERWISE NATSUB ITSELF
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                ;INSTRUCTION IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN NATSUB
                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND NATSUB WILL
                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

023160  012601          NATSUB: MOV     (SP)+,R1                ;GET A POINTER TO THE ARGUMENTS.
023162  010102                  MOV     R1,R2           ;COPY THE OPERAND.
023164  062702  000002          ADD     #2,R2
023170  012703  023562          MOV     #NATBF1,R3
023174  012704  000004          MOV     #4,R4
023200  012223          1$:     MOV     (R2)+,(R3)+
023202  077402                  SOB     R4,1$
023204  016100  000032          MOV     32(R1),R0       ;LOAD THE FPS.
023210  170100                  LDFPS   R0
023212  012700  023562          MOV     #NATBF1,R0      ;SET UP THE OPERAND ADDRESS.
023216  011102                  MOV     (R1),R2         ;GET THE FLAG TO DETERMINE WHICH
023220  006302                  ASL     R2              ;INSTRUCTION TO EXECUTE.
023222  006302                  ASL     R2              ;0 = NEGD, 1 = ABSD, 2 = TSTD
023224  012703  023240          MOV     #NATINS,R3
023230  060203                  ADD     R2,R3
023232  010337  001236          MOV     R3,@#$TMP2
023236  000113                  JMP     (R3)            ;GO EXECUTE THE INSTRUCTION.
023240  170710          NATINS: NEGD    (R0)
023242  000403                  BR      2$
023244  170610                  ABSD    (R0)
023246  000401                  BR      2$
023250  170510                  TSTD    (R0)
```

```
                                        2$:     STFPS   R4              ;GET THE FPS.
                                                STST    R5              ;GET THE FEC.
                                                MOV     R1,R2
                                                ADD     #2,R2
                                                MOV     R2,@#$TMP3
                                                ADD     #10,R2
                                                MOV     R2,@#$TMP5
                                                MOV     #NATBF1,@#$TMP4
                                                MOV     R4,@#$TMP7
                                                MOV     34(R1),@#$TMP10
                                                MOV     R1,R0           ;WAS THE RESULT CORRECT?
                                                ADD     #12,R0
                                                MOV     #NATBF1,R2
                                                MOV     #4,R3
                                        3$:     CMP     (R0)+,(R2)+
                                                BNE     10$             ;BRANCH IF INCORRECT.
                                                SOB     R3,3$
                                                CMP     34(R1),R4       ;WAS THE FPS CORRECT?
                                                BNE     15$             ;BRANCH IF INCORRECT.
                                                TST     34(R1)          ;IF THE EXPECTED FPS WAS NEGATIVE CHECK THE FEC.
                                                BPL     4$
                                                CMP     40(R1),R5       ;WAS THE FEC CORRECT.
                                                BNE     20$             ;BRANCH IF INCORRECT.
                                        4$:     JMP     50(R1)          ;RETURN.

                                        ;THE RESULT WAS INCORRECT BUT WAS THIS FAILURE ANTICIPATED?
                                        ;SEE IF THE RESULT WAS ANTICIPATED:
                                        10$:
                                                MOV     (R1),R5
                                                ASL     R5
                                                ASL     R5
                                                ADD     #NATER1,R5
                                                MOV     R1,R0
                                                ADD     #22,R0
                                                MOV     #NATBF1,R2
                                                MOV     #4,R3
                                        11$:    CMP     (R0)+,(R2)+
                                                BNE     12$             ;BRANCH IF NOT ANTICIPATED.
                                                SOB     R3,11$

                                        ;THE ERROR WAS ANTICIPATED SO RETURN.
                                                JMP     42(R1)

                                        ;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE.
                                        12$:    JMP     (R5)            ;GO TO THE PROPER ERROR CALL.

                                        ;THE FPS WAS INCORRECT.
                                        15$:    CMP     36(R1),R5       ;WAS THIS ERROR ANTICIPATED?
                                                BNE     16$             ;BRANCH IF NOT ANTICIPATED.

                                        ;THE FPS ERROR WAS ANTICIPATED SO RETURN.
                                                JMP     46(R1)

                                        ;THE FPS FAILURE WAS NOT ANTICIPATED SO REPORT IT HERE.
                                        16$:    MOV     (R1),R2
                                                ASL     R2
                                                ASL     R2
```

```
                                        ADD     #NATER2,R2
                                        JMP     (R2)            ;GO TO THE PROPER ERROR CALL.

                                ;REPORT THAT THE FEC WAS INCORRECT.
                        20$:    MOV     40(R1),@#$TMP12
                                MOV     R5,@#$TMP11
                                MOV     (R1),R2
                                ASL     R2
                                ASL     R2
                                ADD     #NATER3,R2
                                JMP     (R2)            ;GO TO THE PROPER ERROR CALL.

                                ;THESE ARE THE ERROR CALLS FOR EACH INDIVIDUAL INSTRUCTION AND CONDITION.
                        NATER1: ERROR   +165            ;NEGD BAD DATA
                                BR      NATRET
                                ERROR   +166            ;ABSD BAD DATA
                                BR      NATRET
                                ERROR   +167            ;TSTD BAD DATA
                        NATRET: JMP     50(R1)

                                ;FPS INCORRECT:
                        NATER2: ERROR   +170            ;NEGD FPSX
                                BR      NATRET
                                ERROR   +171            ;ABSD FPSX
                                BR      NATRET
                                ERROR   +172            ;TSTD FPSX
                                BR      NATRET

                                ;FEC INCORRECT:
                        NATER3: ERROR   +173            ;NEGD FECX
                                BR      NATRET
                                ERROR   +174            ;ABSD FECX
                                BR      NATRET
                                ERROR   +175            ;TSTD FECX
                                BR      NATRET

                                .WORD   -1
                        NATBF1: .WORD   -1,-1,-1,-1,-1

                        WWBDONE:
                                RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                        ;SEE IF THE USER HAS EXPRESSED
                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                        ;THE USER TYPED CONTROL G?).


;;*****************************************************************
;*TEST 45        SOURCE MODES, MODE 1 (FL=0), TEST
;*
;*  THIS IS A TEST OF SOURCE MODE 1
;*  USING THE LDFPS INSTR
;*
;;*****************************************************************
                        TST45:  SCOPE
```

```
                                ;TEST BUFFER AND DATA:
023656  177777                          -1
023660  147517                  AACTP1: 147517
023662  177777                          -1

                                ;REPORT R0 INCORRECT.
023664  012737  023660  001240  AAC10:  MOV     #AACTP1,@#$TMP3
023672  010037  001242                  MOV     R0,@#$TMP4
023676  104225                  1$:     ERROR   +225            ;R0 BAD BUT FSRC FAILED
023700  000425                          BR      AACDONE

                                ;REPORT FPS INCORRECT.
023702  012737  147517  001240  AAC11:  MOV     #147517,@#$TMP3 ;REPORT FPS INCORRECT.
023710  010537  001242                  MOV     R5,@#$TMP4
023714  104226                  1$:     ERROR   +226
023716  000416                          BR      AACDONE

                                ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
                                ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
                                ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
023720                          AAC20:
023720  011602                          MOV     (SP),R2
023722  020227  023636                  CMP     R2,#AAC2+2
023726  001405                          BEQ     1$
023730  020227  023640                  CMP     R2,#AAC2+4
023734  001402                          BEQ     1$
023736  000137  046250                  JMP     @#CPSPUR
023742  022626                  1$:     CMP     (SP)+,(SP)+
023744  010237  001236                  MOV     R2,@#$TMP2
023750  104227                  2$:     ERROR   +227            ;ODD ADRES
023752  000400                          BR      AACDONE         ;BUT FDSTX IN ST 771

023754                          AACDONE:
        023754  104412                  RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
```

```
                                                        ;THE USER TYPED CONTROL G?).

4252
4253
4254 .                                  ;;*****************************************************************
                                        ;*TEST 46        SOURCE MODES, MODE 2 (FL=0), TEST
                                        ;*
                                        ;*  THIS IS A TEST OF SOURCE MODE 2
                                        ;*  USING THE LDFPS INSTR
                                        ;*
                                        ;;*****************************************************************
     023756  000004                     TST46:  SCOPE
4255
4256 023760                             BBC1:
     023760  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4257
4258 023762  012700  024040                    MOV     #BBCTP1,R0      ;SET UP TEST DATA IN BUFFER.
4259 023766  012710  145212                    MOV     #145212,(R0)
4260 023772  012737  145212  001240            MOV     #145212,@#$TMP3 ;SAVE DATA IN CASE OF ERROR.
4261 024000  012737  024014  001236            MOV     #BBC2,@#$TMP2
4262 024006  012737  024100  000004            MOV     #BBC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
4263
4264 024014  170120                     BBC2:   LDFPS   (R0)+           ;TEST INSTRUCTION.
4265
4266 024016  170205                             STFPS   R5              ;GET FPS
4267
4268 024020  020027  024042                    CMP     R0,#BBCTP1+2    ;IS R0 CORRECT?
4269 024024  001007                            BNE     BBC10           ;BR IF NOT.
4270 024026  022705  145212                    CMP     #145212,R5      ;IS THE FPS CORRECT?
4271 024032  001013                            BNE     BBC11           ;BR IF NOT.
4272 024034  000436                            BR      BBCDONE
4273
4274
4275                                    ;TEST BUFFER AND DATA:
4276 024036  177777                             -1
4277 024040  177777                     BBCTP1: .WORD   -1
4278 024042  177777                             -1
4279
4280
4281                                    ;REPORT R0 INCORRECT.
4282 024044  012737  024042  001240     BBC10:  MOV     #BBCTP1+2,@#$TMP3
4283 024052  010037  001242                    MOV     R0,@#$TMP4
4284 024056  104230                     1$:    ERROR   +230            ;R0 BAD BUT FSRC FAILED
4285 024060  000424                            BR      BBCDONE
4286
4287                                    ;REPORT FPS INCORRECT.
4288 024062  012737  145212  001240     BBC11:  MOV     #145212,@#$TMP3 ;REPORT FPS INCORRECT.
4289 024070  010537  001242                    MOV     R5,@#$TMP4
4290 024074  104231                     1$:    ERROR   +231
4291 024076  000415                            BR      BBCDONE
4292
4293                                    ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
4294                                    ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
4295                                    ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
4296 024100                             BBC20:
4297 024100  011602                            MOV     (SP),R2
4298 024102  020227  024016                    CMP     R2,#BBC2+2
4299 024106  001405                            BEQ     1$
```

```
4300 024110  020227  024020            CMP     R2,#BBC2+4
4301 024114  001402                    BEQ     1$
4302 024116  000137  046250            JMP     @#CPSPUR
4303 024122  022626              1$:   CMP     (SP)+,(SP)+
4304 024124  010237  001236            MOV     R2,@#$TMP2
4305 024130  104232              2$:   ERROR   +232            ;ODD ADRES
4306                                                           ;BUT FDSTX IN ST 771
4307
4308 024132                      BBCDONE:
     024132  104412                    RSETUP                 ;GO INITIALIZE THE FPS AND STACK; AND
                                                               ;SEE IF THE USER HAS EXPRESSED
                                                               ;THE DESIRE TO CHANGE THE SOFTWARE
                                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                               ;THE USER TYPED CONTROL G?).
4309
4310
4311                             ;;*****************************************************************
                                 ;*TEST 47       SOURCE MODES, MODE 4 (FL=0), TEST
                                 ;*
                                 ;*  THIS IS A TEST OF SOURCE MODE 4
                                 ;*  USING THE LDFPS INSTR
                                 ;*
                                 ;;*****************************************************************
     024134  000004              TST47:  SCOPE
4312
4313 024136                      DDC1:
     024136  104413                    LPERR                  ;SET UP THE LOOP ON ERROR ADDRESS.
4314
4315 024140  012700  024230            MOV     #DDCTP1+2,R0   ;SET UP THE TEST DATA BUFFER.
4316 024144  012760  105252  177776    MOV     #105252,-2(R0)
4317 024152  012737  105252  001240    MOV     #105252,@#$TMP3 ;SAVE DATA IN CASE OF ERROR.
4318 024160  012737  024174  001236    MOV     #DDC2,@#$TMP2
4319 024166  012737  024274  000004    MOV     #DDC20,@#ERRVEC
4320 024174  170140              DDC2:  LDFPS   -(R0)
4321 024176  170205                     STFPS   R5
4322 024200  020027  024226            CMP     R0,#DDCTP1
4323 024204  001015                    BNE     DDC10
4324 024206  022705  105252            CMP     #105252,R5
4325 024212  001021                    BNE     DDC11
4326 024214  000444                    BR      DDCDONE
4327
4328 024216  177777  177777  177777    -1,-1,-1,-1
     024224  177777
4329 024226  177777              DDCTP1: -1
4330 024230  177777  177777  177777    -1,-1,-1,-1
     024236  177777
4331
4332 024240  012737  024226  001240  DDC10:  MOV     #DDCTP1,@#$TMP3
4333 024246  010037  001242                  MOV     R0,@#$TMP4
4334 024252  104233                  1$:     ERROR   +233           ;R0 BAD BUT FSRC FAILED
4335 024254  000424                          BR      DDCDONE
4336 024256  012737  105252  001240  DDC11:  MOV     #105252,@#$TMP3 ;REPORT FPS INCORRECT.
4337 024264  010537  001242                  MOV     R5,@#$TMP4
4338 024270  104234                  1$:     ERROR   +234
4339 024272  000415                          BR      DDCDONE
4340 024274  011602              DDC20:  MOV     (SP),R2
4341 024276  020227  024176            CMP     R2,#DDC2+2
```

```
 4342 024302  001405                        BEQ     1$
 4343 024304  020227  024200                CMP     R2,#DDC2+4
 4344 024310  001402                        BEQ     1$
 4345 024312  000137  046250                JMP     @#CPSPUR
 4346 024316  022626                1$:     CMP     (SP)+,(SP)+
 4347 024320  010237  001236                MOV     R2,@#$TMP2
 4348 024324  104235                2$:     ERROR   +235            ;DDD ADRES
 4349 024326                      DDCDONE:
      024326  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
 4350                             ;;*******************************************************************
                                  ;*TEST 50       SOURCE MODES, MODE 3 (FL=0), TEST
                                  ;*
                                  ;*  THIS IS A TEST OF SOURCE MODE 3
                                  ;*  USING THE LDFPS INSTR
                                  ;*
                                  ;;*******************************************************************
      024330  000004              TST50:  SCOPE
 4351 024332                      EEC1:
      024332  104413                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4352 024334  012700  024436              MOV     #EECTP2,R0
 4353 024340  012710  024426              MOV     #EECTP1,(R0)
 4354 024344  012767  103456  000054      MOV     #T03456,EECTP1
 4355 024352  012737  103456  001240      MOV     #103456,@#$TMP3
 4356 024360  012737  024374  001236      MOV     #EEC2,@#$TMP2
 4357 024366  012737  024504  000004      MOV     #EEC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
 4358 024374  170130              EEC2:   LDFPS   @(R0)+          ;TEST INSTRUCTION.
 4359 024376  170205                      STFPS   R5              ;GET THE FPS.
 4360 024400  020027  024440              CMP     R0,#EECTP2+2    ;IS R0 CORRECT?
 4361 024404  001021                      BNE     EEC10           ;BR IF NOT.
 4362 024406  022705  103456              CMP     #103456,R5      ;IS THE FPS CORRECT?
 4363 024412  001025                      BNE     EEC11           ;BR IF NOT.
 4364 024414  000450                      BR      EECDONE
 4365
 4366
 4367                                  ;TEST BUFFER AND DATA:
 4368 024416  177777  177777  177777          -1,-1,-1,-1
      024424  177777
 4369 024426  177777              EECTP1: -1
 4370 024430  177777  177777  177777          -1,-1,-1
 4371 024436  024426  177777  177777  EECTP2: EECTP1,-1,-1,-1,
      024444  177777  000000
 4372
 4373
 4374                                  ;REPORT R0 INCORRECT.
 4375 024450  012737  024440  001240  EEC10:  MOV     #EECTP2+2,@#$TMP3
 4376 024456  010037  001242              MOV     R0,@#$TMP4
 4377 024462  104236                1$:     ERROR   +236            ;R0 BAD BUT FSRC FAILED
 4378 024464  000424                      BR      EECDONE
 4379
 4380                                  ;REPORT FPS INCORRECT.
 4381 024466  012737  103456  001240  EEC11:  MOV     #103456,@#$TMP3 ;REPORT FPS INCORRECT.
 4382 024474  010537  001242              MOV     R5,@#$TMP4
 4383 024500  104237                1$:     ERROR   +237
```

```
 4384 024502  000415                        BR      EECDONE
 4385                                ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
 4386                                ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
 4387                                ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
 4388 024504  011602                EEC20:  MOV     (SP),R2
 4389 024506  020227  024376                CMP     R2,#EEC2+2
 4390 024512  001405                        BEQ     1$
 4391 024514  020227  024400                CMP     R2,#EEC2+4
 4392 024520  001402                        BEQ     1$
 4393 024522  000137  046250                JMP     @#CPSPUR
 4394 024526  022626                1$:     CMP     (SP)+,(SP)+
 4395 024530  010237  001236                MOV     R2,@#$TMP2
 4396 024534  104240                2$:     ERROR   +240            ;DDD ADRES
 4397 024536                        EECDONE:
      024536  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                              ;SEE IF THE USER HAS EXPRESSED
                                              ;THE DESIRE TO CHANGE THE SOFTWARE
                                              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                              ;THE USER TYPED CONTROL G?).
 4398                               ;;*****************************************************************
                                    ;*TEST 51        SOURCE MODES, MODE 5 (FL=0), TEST
                                    ;*
                                    ;*  THIS IS A TEST OF SOURCE MODE 5
                                    ;*  USING THE LDFPS INSTR
                                    ;*
                                    ;;*****************************************************************
      024540  000004                TST51:  SCOPE
 4399 024542                        FFC1:
      024542  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4400 024544  012700  024644                MOV     #FFCTP2+2,R0    ;SET UP THE TEST DATA BUFFER.
 4401 024550  012760  024632  177776        MOV     #FFCTP1,-2(R0)
 4402 024556  012737  045412  024632        MOV     #45412,@#FFCTP1
 4403 024564  012737  045412  001240        MOV     #45412,@#$TMP3  ;SAVE DATA IN CASE OF ERROR.
 4404 024572  012737  024542  001236        MOV     #FFC1,@#$TMP2
 4405 024600  012737  024706  000004        MOV     #FFC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
 4406 024606  170150                FFC2:   LDFPS   @-(R0)          ;TEST INSTRUCTION.
 4407 024610  170205                        STFPS   R5              ;GET THE FPS.
 4408 024612  020027  024642                CMP     R0,#FFCTP2      ;IS R0 CORRECT?
 4409 024616  001015                        BNE     FFC10           ;BR IF NOT.
 4410 024620  022705  045412                CMP     #45412,R5       ;IS THE FPS CORRECT?
 4411 024624  001021                        BNE     FFC11           ;BR IF NOT.
 4412 024626  000444                        BR      FFCDONE
 4413
 4414
 4415                                ;TEST BUFFER AND DATA:
 4416 024630  177777                        -1
 4417 024632  177777                FFCTP1: -1
 4418 024634  177777  177777  177777        -1,-1,-1
 4419 024642  024632  177777  177777  FFCTP2: FFCTP1,-1,-1,-1
      024650  177777
 4420
 4421
 4422                                ;REPORT R0 INCORRECT.
 4423 024652  012737  024642  001240  FFC10:  MOV     #FFCTP2,@#$TMP3
 4424 024660  010037  001242                MOV     R0,@#$TMP4
 4425 024664  104241                1$:     ERROR   +241            ;R0 BAD BUT FSRC FAILED
 4426 024666  000424                        BR      FFCDONE
```

```
                                        ;REPORT FPS INCORRECT.
024670  012737  045412  001240  FFC11:  MOV     #45412,@#$TMP3  ;REPORT FPS INCORRECT.
024676  010537  001242                  MOV     R5,@#$TMP4
024702  104242                  1$:     ERROR   +242
024704  000415                          BR      FFCDONE
                                        ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
                                        ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
                                        ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
024706  011602                  FFC20:  MOV     (SP),R2
024710  020227  024610                  CMP     R2,#FFC2+2
024714  001405                          BEQ     1$
024716  020227  024612                  CMP     R2,#FFC2+4
024722  001402                          BEQ     1$
024724  000137  046250                  JMP     @#CPSPUR
024730  022626                  1$:     CMP     (SP)+,(SP)+
024732  010237  001236                  MOV     R2,@#$TMP2
024736  104243                  2$:     ERROR   +243            ;ODD ADRES
024740                          FFCDONE:
024740  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).
                                ;;**************************************************************
                                ;*TEST 52       SOURCE MODES, MODE 6 (FL=0), TEST
                                ;*
                                ;*  THIS IS A TEST OF SOURCE MODE 6
                                ;*  USING THE LDFPS INSTR
                                ;*
                                ;;**************************************************************
024742  000004                  TST52:  SCOPE
024744                          GGC1:
024744  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
024746  012700  017635                  MOV     #GGCTP1-5201,R0 ;SET UP THE TEST DATA BUFFER.
024752  012737  046543  025036          MOV     #46543,@#GGCTP1
024760  012737  046543  001240          MOV     #46543,@#$TMP3  ;SAVE DATA IN CASE OF ERROR.
024766  012737  025004  001236          MOV     #GGC2,@#$TMP2
024774  005001                          CLR     R1
024776  012737  025124  000004          MOV     #GGC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
025004  170160  005201          GGC2:   LDFPS   5201(R0)                ;TEST INSTRUCTION.
025010  170204                          STFPS   R4              ;GET THE FPS.
025012  005701                          TST     R1              ;WAS PC CORRECT AFTER EXECUTION?
025014  001033                          BNE     GGC25           ;BR IF NOT.
025016  020027  017635                  CMP     R0,#GGCTP1-5201 ;IS R0 CORRECT?
025022  001012                          BNE     GGC10           ;BR IF NOT.
025024  022704  046543                  CMP     #46543,R4       ;IS THE FPS CORRECT?
025030  001016                          BNE     GGC11           ;BR IF NOT.
025032  000451                          BR      GGCDONE


                                ;TEST BUFFER AND DATA:
                                        -1
025036  177777  177777  177777  GGCTP1: -1,-1,-1,-1
025044  177777
025046  177777                          -1
```

```
                                        ;REPORT R0 INCORRECT.
025050  012737  017635  001240  GGC10:  MOV     #GGCTP1-5201,@#$TMP3
025056  010037  001242                  MOV     R0,@#$TMP4
025062  104244                  1$:     ERROR   +244            ;R0 BAD BUT FSRC FAILED
025064  000434                          BR      GGCDONE

                                        ;REPORT FPS INCORRECT.
025066  012737  046543  001240  GGC11:  MOV     #46543,@#$TMP3  ;REPORT FPS INCORRECT.
025074  010437  001242                  MOV     R4,@#$TMP4
025100  104245                  1$:     ERROR   +245
025102  000425                          BR      GGCDONE

                                        ;REPORT PC INCORRECT AFTER INSTRUCTION.
025104  012737  025010  001240  GGC25:  MOV     #GGC2+4,@#$TMP3
025112  012737  025006  001242          MOV     #GGC2+2,@#$TMP4
025120  104246                  1$:     ERROR   +246            ;PC X
025122  000415                          BR      GGCDONE
                                        ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
                                        ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
                                        ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
025124  011602                  GGC20:  MOV     (SP),R2
025126  020227  025006                  CMP     R2,#GGC2+2
025132  001405                          BEQ     1$
025134  020227  025010                  CMP     R2,#GGC2+4
025140  001402                          BEQ     1$
025142  000137  046250                  JMP     @#CPSPUR
025146  022626                  1$:     CMP     (SP)+,(SP)+
025150  010237  001236                  MOV     R2,@#$TMP2
025154  104247                  2$:     ERROR   +247            ;ODD ADRES
025156                          GGCDONE:
025156  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).
                                        ;;**************************************************************
                                        ;*TEST 53       SOURCE MODES, MODE 7 (FL=0), TEST
                                        ;*
                                        ;*  THIS IS A TEST OF SOURCE MODE 7
                                        ;*  USING THE LDFPS INSTR
                                        ;*
                                        ;;**************************************************************
025160  000004                  TST53:  SCOPE
025162                          HHC1:
025162  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
025164  012700  020071                  MOV     #HHCTP2-5201,R0 ;SET UP THE TEST DATA BUFFER.
025170  012760  025262  005201          MOV     #HHCTP1,5201(R0)
025176  012737  004547  025262          MOV     #4547,@#HHCTP1
025204  012737  004547  001240          MOV     #4547,@#$TMP3   ;SAVE DATA IN CASE OF ERROR.
025212  012737  025230  001236          MOV     #HHC2,@#$TMP2
025220  005001                          CLR     R1
025222  012737  025356  000004          MOV     #HHC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
025230  170170  005201          HHC2:   LDFPS   @5201(R0)               ;TEST INSTRUCTION.
025234  170204                          STFPS   R4              ;GET THE FPS.
025236  005701                          TST     R1              ;WAS PC CORRECT AFTER EXECUTION?
025240  001036                          BNE     HHC25           ;BR IF NOT.
025242  020027  020071                  CMP     R0,#HHCTP2-5201 ;IS R0 CORRECT?
```

```
4514 025246  001015                        BNE     HHC10           ;BR IF NOT.
4515 025250  022704  004547                CMP     #4547,R4        ;IS THE FPS CORRECT?
4516 025254  001021                        BNE     HHC11           ;BR IF NOT.
4517 025256  000454                        BR      HHCDONE
4518
4519
4520                                ;TEST BUFFER AND DATA:
4521 025260  177777                        -1
4522 025262  177777  177777  177777  HHCTP1: .WORD -1,-1,-1,-1
     025270  177777
4523 025272  177777  177777  177777  HHCTP2: .WORD -1,-1,-1,-1
     025300  177777
4524
4525                                ;REPORT R0 INCORRECT.
4526 025302  012737  020071  001240  HHC10:  MOV     #HHCTP2-5201,@#$TMP3
4527 025310  010037  001242                MOV     R0,@#$TMP4
4528 025314  104250                  1$:   ERROR   +250            ;R0 BAD BUT FSRC FAILED
4529 025316  000434                        BR      HHCDONE
4530
4531                                ;REPORT FPS INCORRECT.
4532 025320  012737  004547  001240  HHC11:  MOV     #4547,@#$TMP3   ;REPORT FPS INCORRECT.
4533 025326  010437  001242                MOV     R4,@#$TMP4
4534 025332  104251                  1$:   ERROR   +251
4535 025334  000425                        BR      HHCDONE
4536
4537                                ;REPORT PC INCORRECT AFTER INSTRUCTION.
4538 025336  012737  025234  001240  HHC25:  MOV     #HHC2+4,@#$TMP3
4539 025344  012737  025232  001242          MOV     #HHC2+2,@#$TMP4
4540 025352  104252                  1$:   ERROR   +252            ;PC X
4541 025354  000415                        BR      HHCDONE
4542                                ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
4543                                ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
4544                                ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
4545 025356  011602                HHC20:  MOV     (SP),R2
4546 025360  020227  025232                CMP     R2,#HHC2+2
4547 025364  001405                        BEQ     1$
4548 025366  020227  025234                CMP     R2,#HHC2+4
4549 025372  001402                        BEQ     1$
4550 025374  000137  046250                JMP     @#CPSPUR
4551 025400  022626                  1$:   CMP     (SP)+,(SP)+
4552 025402  010237  001236                MOV     R2,@#$TMP2
4553 025406  104253                  2$:   ERROR   +253            ;DDD ADDRESS
4554 025410                         HHCDONE:
     025410  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                            ;SEE IF THE USER HAS EXPRESSED
                                            ;THE DESIRE TO CHANGE THE SOFTWARE
                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                            ;THE USER TYPED CONTROL G?).
4555
4556
4563
4564                                ;;**********************************************************************
                                    ;*TEST 54       SOURCE MODES, MODE 2 GR7 (FL=1), TEST
                                    ;*
                                    ;* THIS IS A TEST OF THE LDCLD WITH
                                    ;* IMMEDIATE ADDRESSING MODE
                                    ;*
```

```
                                        ;;*****************************************************************
     025412  000004                     TST54:  SCOPE
4565
4566 025414                             IIC1:
     025414  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4567 025416  012737  025442  001236             MOV     #IIC2,@#$TMP2   ;SAVE DATA IN CASE OF ERROR.
4568 025424  012737  025514  000004             MOV     #IIC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
4569 025432  012700  000300                     MOV     #300,R0
4570 025436  170100                             LDFPS   R0
4571 025440  005001                             CLR     R1
4572
4573 025442  177027                     IIC2:   LDCLD   (R7)+,AC0               ;TEST INSTRUCTION.
4574 025444  005201                             5201
4575 025446  005201                             5201
4576 025450  005201                             5201
4577 025452  005201                             5201
4578
4579 025454  020127  000003                     CMP     R1,#3           ;WAS PC CORRECT AFTER EXECUTION?
4580 025460  001421                             BEQ     IICDONE         ;BR IF YES.
4581
4582
4583                                    ;REPORT PC INCORRECT AFTER INSTRUCTION.
4584 025462  012704  025446             IIC3:   MOV     #IIC2+4,R4
4585 025466  162701  000003                     SUB     #3,R1
4586 025472  006301                             ASL     R1
4587 025474  160104                             SUB     R1,R4
4588 025476  010437  001242                     MOV     R4,@#$TMP4
4589 025502  012737  025446  001240             MOV     #IIC2+4,@#$TMP3
4590 025510  104254                     1$:     ERROR   +254            ;BAD CONSTANT
4591 025512  000404                             BR      IICDONE
4592                                    ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
4593                                    ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
4594                                    ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
4595 025514  011637  001236             IIC20:  MOV     (SP),@#$TMP2
4596 025520  022626                             CMP     (SP)+,(SP)+
4597 025522  104255                     1$:     ERROR   +255            ;BAD CONSTANT ODD ADD
4598
4599 025524                             IICDONE:
     025524  104412                             RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                         ;SEE IF THE USER HAS EXPRESSED
                                                                         ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                         ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                         ;THE USER TYPED CONTROL G?).
4600
4607
4608                                    ;;*****************************************************************
                                        ;*TEST 55       SOURCE MODES, MODE 2 (FL=1), TEST
                                        ;*
                                        ;* THIS IS A TEST OF THE LDCLD INSTR
                                        ;* WITH MODE 2.
                                        ;*
                                        ;;*****************************************************************
     025526  000004                     TST55:  SCOPE
4609
4610 025530                             TCC1:
     025530  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4611 025532  016737  000014  001236             MOV     TCC2,@#$TMP2    ;SAVE DATA IN CASE OF ERROR.
```

```
 4612 025540  012700  000300                MOV     #300,R0
 4613 025544  170100                        LDFPS   R0
 4614 025546  012700  025642                MOV     #TCCBF0,R0      ;SET UP THE TEST DATA BUFFER.
 4615 025552  177020                TCC2:   LDCLD   (R0)+,AC0               ;TEST INSTRUCTION.
 4616
 4617 025554  170204                        STFPS   R4              ;GET THE FPS.
 4618 025556  012701  025652                MOV     #TCCBF1,R1      ;GET THE RESULT.
 4619 025562  012702  000200                MOV     #200,R2
 4620 025566  170102                        LDFPS   R2
 4621 025570  174011                        STD     AC0,(R1)
 4622 025572  020027  025646                CMP     R0,#TCCBF0+4    ;IS R0 CORRECT?
 4623 025576  001407                        BEQ     TCC3
 4624                               ;REPORT R0 INCORRECT.
 4625 025600  010037  001242                MOV     R0,@#$TMP4
 4626 025604  012737  025646  001240        MOV     #TCCBF0+4,@#$TMP3
 4627 025612  104256                1$:     ERROR   +256            ;BAD CONST
 4628 025614  000422                        BR      TCCDONE
 4629
 4630 025616  022704  000300        TCC3:   CMP     #300,R4 ;IS THE FPS CORRECT?
 4631 025622  001417                        BEQ     TCCDONE
 4632
 4633                               ;REPORT FPS INCORRECT.
 4634 025624  010437  001242                MOV     R4,@#$TMP4
 4635 025630  012737  000300  001240        MOV     #300,@#$TMP3
 4636 025636  104257                1$:     ERROR   +257            ;FPS X
 4637 025640  000410                        BR      TCCDONE
 4638
 4639
 4640                               ;TEST BUFFER AND DATA:
 4641 025642  001234  067076  054321  TCCBF0: .WORD   01234,67076,54321,012345
      025650  012345
 4642 025652  177777  177777  177777  TCCBF1: -1,-1,-1,-1
      025660  177777
 4643
 4644 025662                        TCCDONE:
      025662  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                              ;SEE IF THE USER HAS EXPRESSED
                                              ;THE DESIRE TO CHANGE THE SOFTWARE
                                              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                              ;THE USER TYPED CONTROL G?).
 4645
 4646
 4653
 4654                                  ;;*****************************************************************
                                       ;*TEST 56       LDCIF AND LDCLF TEST
                                       ;*
                                       ;* THIS IS A TEST OF THE LDCIF AND
                                       ;* THE LDCLF INSTRUCTIONS.
                                       ;*
                                       ;;*****************************************************************
      025664  000004                   TST56:  SCOPE
 4655
 4656
 4657                                  ;ZERO   OPERAND FL=0
 4658
 4659 025666                            KKC1:
      025666  104413                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
4660 025670  004737  027020                    JSR     PC,@#LDCFSUB    ;GO EXECUTE INSTRUCTION.
4661
4662 025674  000000  000000          1$:       .WORD   0,0             ;FSRC OPERAND.
4663 025700  000000  000000          2$:       .WORD   0,0             ;EXPECTED RESULT.
4664 025704  177777  177777          3$:       .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
4665 025710  000000                  4$:       0                       ;FPS BEFORE EXECUTION.
4666 025712  000004                            4                       ;FPS AFTER EXECUTION.
4667 025714  177777                            -1                      ;ANTICIPATED ERRONEOUS FPS.
4668 025716  104260                  5$:       ERROR   +260            ;REPORT RESULT INCORRECT.
4669 025720  000401                            BR      6$
4670 025722  104261                            ERROR   +261            ;REPORT FPS INCORRECT.
4671 025724                         6$:
4672                                ;ZERO   OPERAND FL=0
4673
4674 025724                         KKC2:
     025724  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4675 025726  004737  027020                    JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
4676
4677 025732  000000  177777          1$:       .WORD   0,-1            ;FSRC OPERAND.
4678 025736  000000  000000          2$:       .WORD   0,0             ;EXPECTED RESULT.
4679 025742  004177  177400          3$:       4177,177400             ;ANTICIPATED ERRONEOUS RESULT.
4680 025746  000000                  4$:       0                       ;FPS BEFORE EXECUTION.
4681 025750  000004                            4                       ;FPS AFTER EXECUTION.
4682 025752  177777                            -1                      ;ANTICIPATED ERRONEOUS FPS.
4683 025754  104262                  5$:       ERROR   +262            ;(BUT FL) ST
4684 025756  000401                            BR      6$              ;277 TO 300
4685 025760  104261                            ERROR   +261            ;INTO 301
4686 025762                         6$:
4687                                ;ZERO   OPERAND FL=1
4688
4689 025762                         KKC3:
     025762  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4690 025764  004737  027020                    JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
```

```
4692
4693 025770  000000  000000          1$:     .WORD   0,0             ;FSRC OPERAND.
4694 025774  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
4695 026000  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
4696 026004  000100                  4$:     100                     ;FPS BEFORE EXECUTION.
4697 026006  000104                          104                     ;FPS AFTER EXECUTION.
4698 026010  000004                          4                       ;ANTICIPATED ERRONEOUS FPS.
4699 026012  104260                  5$:     ERROR   +260            ;REPORT RESULT INCORRECT.
4700 026014  000401                          BR      6$
4701 026016  104263                          ERROR   +263    ;FL WAS CLR'ED
4702 026020                          6$:
4703                                 ;OPERAND        POSITIVE        FL=0
4704 026020                          KKC4:
     026020  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4705 026022  004737  027020                  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
4706 026026  040000  000000          1$:     .WORD   40000,0         ;FSRC OPERAND.
4707 026032  043600  000000          2$:     .WORD   43600,0         ;EXPECTED RESULT.
4708 026036  047600  000000          3$:     .WORD   47600,0         ;ANTICIPATED ERRONEOUS RESULT.
4709 026042  000017                  4$:     17                      ;FPS BEFORE EXECUTION.
4710 026044  000000                          0                       ;FPS AFTER EXECUTION.
4711 026046  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
4712 026050  104264                  5$:     ERROR   +264    ;ST 107 BAD
4713 026052  000401                          BR      6$              ;CONSTANT 231 INSD
4714 026054  104261                          ERROR   +261            ;215
4715 026056                          6$:
4716                                 ;OPERAND=1,     FL=0
4717 026056                          KKC5:
     026056  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4718 026060  004737  027020                  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
4719 026064  000001  000000          1$:     .WORD   1,0             ;FSRC OPERAND.
4720 026070  040200  000000          2$:     .WORD   40200,0         ;EXPECTED RESULT.
4721 026074  044200  000000          3$:     .WORD   44200,0         ;ANTICIPATED ERRONEOUS RESULT.
4722 026100  000017                  4$:     17                      ;FPS BEFORE EXECUTION.
4723 026102  000000                          0                       ;FPS AFTER EXECUTION.
4724 026104  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
4725 026106  104264                  5$:     ERROR   +264            ;REPORT RESULT INCORRECT.
4726 026110  000401                          BR      6$
4727 026112  104261                          ERROR   +261            ;REPORT FPS INCORRECT.
4728 026114                          6$:
4729
4730
4731                                 ;OPERAND=       PATTERN FL=0
4732 026114                          KKC6:
     026114  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4733 026116  004737  027020                  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
4734 026122  000252  000000          1$:     .WORD   252,0           ;FSRC OPERAND.
4735 026126  042052  000000          2$:     .WORD   42052,0         ;EXPECTED RESULT.
4736 026132  046052  000000          3$:     .WORD   46052,0         ;ANTICIPATED ERRONEOUS RESULT.
4737 026136  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
4738 026140  000000                          0                       ;FPS AFTER EXECUTION.
4739 026142  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
4740 026144  104264                  5$:     ERROR   +264            ;REPORT RESULT INCORRECT.
4741 026146  000401                          BR      6$
4742 026150  104261                          ERROR   +261            ;REPORT FPS INCORRECT.
4743 026152                          6$:
4744
4745                                 ;OPERAND=-40000 FL=0
```

```
 4746 026152                        KKC7:
      026152  104413                       LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4747 026154  004737  027020               JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4748 026160  140000  000000        1$:    .WORD   -40000,0                ;FSRC OPERAND.
 4749 026164  143600  000000        2$:    .WORD   143600,0                ;EXPECTED RESULT.
 4750 026170  043600  000000        3$:    .WORD   43600,0         ;ANTICIPATED ERRONEOUS RESULT.
 4751 026174  000007                4$:    7                       ;FPS BEFORE EXECUTION.
 4752 026176  000010                       10                      ;FPS AFTER EXECUTION.
 4753 026200  177777                       -1                      ;ANTICIPATED ERRONEOUS FPS.
 4754 026202  104265                5$:    ERROR   +265            ;(SET SIGN) ST 146
 4755 026204  000401                       BR      6$
 4756 026206  104261                       ERROR   +261            ;REPORT FPS INCORRECT.
 4757 026210                        6$:
 4758
 4759                               ;OPERAND=-1     FL-0
 4760 026210                        KKC8:
      026210  104413                       LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4761 026212  004737  027020               JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4762 026216  177777  000000        1$:    .WORD   -1,0            ;FSRC OPERAND.
 4763 026222  140200  000000        2$:    .WORD   140200,0                ;EXPECTED RESULT.
 4764 026226  144000  000400        3$:    .WORD   144000,400              ;ANTICIPATED ERRONEOUS RESULT.
 4765 026232  000000                4$:    0                       ;FPS BEFORE EXECUTION.
 4766 026234  000010                       10                      ;FPS AFTER EXECUTION.
 4767 026236  177777                       -1                      ;ANTICIPATED ERRONEOUS FPS.
 4768 026240  104266                5$:    ERROR   +266            ;ST 372 TO 152 INTO
 4769 026242  000401                       BR      6$              ;112 (BUF XNBT)
 4770 026244  104261                       ERROR   +261            ;REPORT FPS INCORRECT.
 4771 026246                        6$:
 4772
 4773                               ;OPERAND=PATTERN          FL=0
 4774 026246                        KKC9:
      026246  104413                       LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4775 026250  004737  027020               JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4776 026254  125252  000000        1$:    .WORD   125252,0                ;FSRC OPERAND.
 4777 026260  143652  126000        2$:    .WORD   143652,126000           ;EXPECTED RESULT.
 4778 026264  043652  126000        3$:    .WORD   43652,126000            ;ANTICIPATED ERRONEOUS RESULT.
 4779 026270  000007                4$:    7                       ;FPS BEFORE EXECUTION.
 4780 026272  000010                       10                      ;FPS AFTER EXECUTION.
 4781 026274  177777                       -1                      ;ANTICIPATED ERRONEOUS FPS.
 4782 026276  104265                5$:    ERROR   +265            ;REPORT RESULT INCORRECT.
 4783 026300  000401                       BR      6$
 4784 026302  104261                       ERROR   +261            ;REPORT FPS INCORRECT.
 4785 026304                        6$:
 4786
 4787                               ;OPERAND          POS     FL-1
 4788 026304                        KKC10:
      026304  104413                       LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4789 026306  004737  027020               JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4790 026312  040000  000000        1$:    .WORD   40000,0         ;FSRC OPERAND.
 4791 026316  047600  000000        2$:    .WORD   47600,0         ;EXPECTED RESULT.
 4792 026322  043600  000000        3$:    .WORD   43600,0         ;ANTICIPATED ERRONEOUS RESULT.
 4793 026326  000117                4$:    117                     ;FPS BEFORE EXECUTION.
 4794 026330  000100                       100                     ;FPS AFTER EXECUTION.
 4795 026332  177777                       -1                      ;ANTICIPATED ERRONEOUS FPS.
 4796 026334  104267                5$:    ERROR   +267    ;ST 107 CONSTANT
 4797 026336  000401                       BR      6$              ;BAD 237 INST 217
 4798 026340  104261                       ERROR   +261            ;REPORT FPS INCORRECT.
```

```
 4799 026342                          6$:
 4800
 4801                                 ;OPERAND=1        FL=1
 4802 026342                          KKC11:
      026342  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4803 026344  004737  027020                 JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4804 026350  000000  000001          1$:    .WORD   0,1             ;FSRC OPERAND.
 4805 026354  040200  000000          2$:    .WORD   40200,0         ;EXPECTED RESULT.
 4806 026360  034200  000000          3$:    .WORD   34200,0         ;ANTICIPATED ERRONEOUS RESULT.
 4807 026364  000100                  4$:    100                     ;FPS BEFORE EXECUTION.
 4808 026366  000100                         100                     ;FPS AFTER EXECUTION.
 4809 026370  177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
 4810 026372  104267                  5$:    ERROR   +267            ;REPORT RESULT INCORRECT.
 4811 026374  000401                         BR      6$
 4812 026376  104261                         ERROR   +261            ;REPORT FPS INCORRECT.
 4813 026400                          6$:
 4814
 4815                                 ;OPERAND=       PATTERN FL=1
 4816 026400                          KKC12:
      026400  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4817 026402  004737  027020                 JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4818 026406  000000  000252          1$:    .WORD   0,252           ;FSRC OPERAND.
 4819 026412  042052  000000          2$:    .WORD   42052,0         ;EXPECTED RESULT.
 4820 026416  036052  000000          3$:    .WORD   36052,0         ;ANTICIPATED ERRONEOUS RESULT.
 4821 026422  000111                  4$:    111                     ;FPS BEFORE EXECUTION.
 4822 026424  000100                         100                     ;FPS AFTER EXECUTION.
 4823 026426  177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
 4824 026430  104267                  5$:    ERROR   +267            ;REPORT RESULT INCORRECT.
 4825 026432  000401                         BR      6$
 4826 026434  104261                         ERROR   +261            ;REPORT FPS INCORRECT.
 4827 026436                          6$:
 4828
 4829                                 ;OPERAND=-40000,0       FL=1
 4830 026436                          KKC13:
      026436  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4831 026440  004737  027020                 JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4832 026444  140000  000000          1$:    .WORD   -40000,0                ;FSRC OPERAND.
 4833 026450  147600  000000          2$:    .WORD   147600,0                ;EXPECTED RESULT.
 4834 026454  047600  000000          3$:    .WORD   47600,0         ;ANTICIPATED ERRONEOUS RESULT.
 4835 026460  000107                  4$:    107                     ;FPS BEFORE EXECUTION.
 4836 026462  000110                         110                     ;FPS AFTER EXECUTION.
 4837 026464  177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
 4838 026466  104265                  5$:    ERROR   +265    ;SET SIGN
 4839 026470  000401                         BR      6$
 4840 026472  104261                         ERROR   +261            ;REPORT FPS INCORRECT.
 4841 026474                          6$:
 4842
 4843                                 ;OPERAND=-1,-1  FL=1
 4844 026474                          KKC14:
      026474  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4845 026476  004737  027020                 JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4846 026502  177777  177777          1$:    .WORD   -1,-1           ;FSRC OPERAND.
 4847 026506  140200  000000          2$:    .WORD   140200,0                ;EXPECTED RESULT.
 4848 026512  150000  000000          3$:    .WORD   150000,0                ;ANTICIPATED ERRONEOUS RESULT.
 4849 026516  000100                  4$:    100                     ;FPS BEFORE EXECUTION.
 4850 026520  000110                         110                     ;FPS AFTER EXECUTION.
 4851 026522  177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
```

```
4852 026524  104266                5$:     ERROR   +266            ;(BUT XNBT)
4853 026526  000401                        BR      6$
4854 026530  104261                        ERROR   +261            ;REPORT FPS INCORRECT.
4855 026532                        6$:
4856
4857                               ;OPERAND=-PATTERN        FL=1,  ROUND MODE
4858 026532                        KKC15:
     026532  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4859 026534  004737  027020                JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
4860 026540  125252  125252        1$:     .WORD   125252,125252           ;FSRC OPERAND.
4861 026544  147652  125253        2$:     .WORD   147652,125253           ;EXPECTED RESULT.
4862 026550  047652  125253        3$:     .WORD   47652,125253            ;ANTICIPATED ERRONEOUS RESULT.
4863 026554  000105                4$:     105                     ;FPS BEFORE EXECUTION.
4864 026556  000110                        110                     ;FPS AFTER EXECUTION.
4865 026560  177777                        -1                      ;ANTICIPATED ERRONEOUS FPS.
4866 026562  104265                5$:     ERROR   +265            ;REPORT RESULT INCORRECT.
4867 026564  000401                        BR      6$
4868 026566  104261                        ERROR   +261            ;REPORT FPS INCORRECT.
4869 026570                        6$:
4870
4871                               ;OPERAND=77777,177500    FL=1,  ROUND MODE
4872 026570                        KKC16:
     026570  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4873 026572  004737  027020                JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
4874 026576  077777  177500        1$:     .WORD   77777,177500            ;FSRC OPERAND.
4875 026602  047777  177777        2$:     .WORD   47777,177777            ;EXPECTED RESULT.
4876 026606  047777  177776        3$:     .WORD   47777,177776            ;ANTICIPATED ERRONEOUS RESULT.
4877 026612  000117                4$:     117                     ;FPS BEFORE EXECUTION.
4878 026614  000100                        100                     ;FPS AFTER EXECUTION.
4879 026616  177777                        -1                      ;ANTICIPATED ERRONEOUS FPS.
4880 026620  104270                5$:     ERROR   +270            ;ST 631 INTO RND
4881 026622  000401                        BR      6$
4882 026624  104261                        ERROR   +261            ;REPORT FPS INCORRECT.
4883 026626                        6$:
4884
4885                               ;OPERAND=40000,000100    FL=1,  ROUND MODE
4886 026626                        KKC17:
     026626  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4887 026630  004737  027020                JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
4888 026634  040000  000100        1$:     .WORD   40000,100               ;FSRC OPERAND.
4889 026640  047600  000001        2$:     .WORD   47600,1         ;EXPECTED RESULT.
4890 026644  047600  000000        3$:     .WORD   47600,0         ;ANTICIPATED ERRONEOUS RESULT.
4891 026650  000102                4$:     102                     ;FPS BEFORE EXECUTION.
4892 026652  000100                        100                     ;FPS AFTER EXECUTION.
4893 026654  177777                        -1                      ;ANTICIPATED ERRONEOUS FPS.
4894 026656  104270                5$:     ERROR   +270            ;REPORT RESULT INCORRECT.
4895 026660  000401                        BR      6$
4896 026662  104261                        ERROR   +261            ;REPORT FPS INCORRECT.
4897 026664                        6$:
4898
4899                               ;OPERAND=40000,000100    FL=1,  TRUNC MODE
4900 026664                        KKC18:
     026664  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4901 026666  004737  027020                JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
4902 026672  040000  000100        1$:     .WORD   40000,100               ;FSRC OPERAND.
4903 026676  047600  000000        2$:     .WORD   47600,0         ;EXPECTED RESULT.
4904 026702  047600  000001        3$:     .WORD   47600,1         ;ANTICIPATED ERRONEOUS RESULT.
```

```
 4905 026706  000157                4$:     157                     ;FPS BEFORE EXECUTION.
 4906 026710  000140                        140                     ;FPS AFTER EXECUTION.
 4907 026712  177777                        -1                      ;ANTICIPATED ERRONEOUS FPS.
 4908 026714  104271                5$:     ERROR   +271            ;ST 631 ... INTO TRNC
 4909 026716  000401                        BR      6$
 4910 026720  104261                        ERROR   +261            ;REPORT FPS INCORRECT.
 4911 026722                        6$:
 4912                               ;OPERAND=100000,0 (MOST NEG #)  FL=0
 4913 026722                        KKC19:
      026722  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4914 026724  004737  027020                JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4915 026730  100000  000000        1$:     .WORD   100000,0                ;FSRC OPERAND.
 4916 026734  144000  000000        2$:     .WORD   144000,0                ;EXPECTED RESULT.
 4917 026740  143600  000000        3$:     .WORD   143600,0                ;ANTICIPATED ERRONEOUS RESULT.
 4918 026744  000007                4$:     7                       ;FPS BEFORE EXECUTION.
 4919 026746  000010                        10                      ;FPS AFTER EXECUTION.
 4920 026750  177777                        -1                      ;ANTICIPATED ERRONEOUS FPS.
 4921 026752  104272                5$:     ERROR   +272            ;ST 630 RH*R14+1
 4922 026754  000401                        BR      6$
 4923 026756  104261                        ERROR   +261            ;REPORT FPS INCORRECT.
 4924 026760                        6$:
 4925
 4926                               ;OPERAND=100000,0       FL=1
 4927 026760                        KKC20:
      026760  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4928 026762  004737  027020                JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
 4929 026766  100000  000000        1$:     .WORD   100000,0                ;FSRC OPERAND.
 4930 026772  150000  000000        2$:     .WORD   150000,0                ;EXPECTED RESULT.
 4931 026776  147600  000000        3$:     .WORD   147600,0                ;ANTICIPATED ERRONEOUS RESULT.
 4932 027002  000107                4$:     107                     ;FPS BEFORE EXECUTION.
 4933 027004  000110                        110                     ;FPS AFTER EXECUTION.
 4934 027006  177777                        -1                      ;ANTICIPATED ERRONEOUS FPS.
 4935 027010  104272                5$:     ERROR   +272            ;REPORT RESULT INCORRECT.
 4936 027012  000401                        BR      6$
 4937 027014  104261                        ERROR   +261            ;REPORT FPS INCORRECT.
 4938 027016  000506                6$:     BR      KKCDONE
 4939
 4940                               ;THIS SUBROUTINE, LDCFSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
 4941                               ;THE LDCIF OR LDCLF INSTRUCTION AND CHECK THE RESULTS. A CALL
 4942                               ;TO IT IS MADE THUS:
 4943                               ;
 4944                               ;               JSR     PC,@#LDCFSUB
 4945                               ;               ACARG:  .WORD   X,X             ;AC OPERAND
 4946                               ;               RES:    .WORD   X,X             ;EXPECTED RESULT
 4947                               ;               ERRES:  .WORD   X,X             ;ERROR RESULT
 4948                               ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
 4949                               ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
 4950                               ;               ERFPS:  .WORD   X               ;ERROR FPS
 4951                               ;               ERR1:   ERROR   +X              ;DATA ERROR
 4952                               ;                       BR      CONT
 4953                               ;               ERR2:   ERROR   +X              ;FPS ERROR
 4954                               ;               CONT:                           ;RETURN ADDRESS
 4955                               ;
 4956                               ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
 4957                               ;THE LDCIF OR LDCLF INSTRUCTION IS EXECUTED.
 4958                               ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
 4959                               ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCFSUB RETURNS CONTROL
```

```
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCFSUB WILL
                                ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCFSUB WILL RETURN
                                ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCFSUB ITSELF
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                ;LDCIF OR LDCLF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCFSUB
                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCFSUB
                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

027020  012601                  LDCFSUB:  MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
027022  016100  000014                    MOV     14(R1),R0       ;SET THE FPS.
027026  170100                            LDFPS   R0
027030  012737  027040  001236            MOV     #1$,@#$TMP2
027036  010100                            MOV     R1,R0

027040  177010                  1$:       LDCIF   (R0),AC0                ;TEST INSTRUCTION LDCIF OR LDCLF.

027042  170204                            STFPS   R4              ;GET FPS.
027044  012700  027224                    MOV     #LDCT,R0        ;GET THE RESULT.
027050  012702  000200                    MOV     #200,R2
027054  170102                            LDFPS   R2
027056  174010                            STD     AC0,(R0)

027060  012702  027224                    MOV     #LDCT,R2        ;SEE IF THE RESULT WAS CORRECT.
027064  010237  001242                    MOV     R2,@#$TMP4
027070  010137  001240                    MOV     R1,@#$TMP3
027074  010103                            MOV     R1,R3
027076  062703  000004                    ADD     #4,R3
027102  010337  001244                    MOV     R3,@#$TMP5
027106  010437  001250                    MOV     R4,@#$TMP7
027112  016137  000016  001252            MOV     16(R1),@#$TMP10
027120  010100                            MOV     R1,R0
027122  062700  000004                    ADD     #4,R0
027126  012703  000002                    MOV     #2,R3
027132  022022                  2$:       CMP     (R0)+,(R2)+
027134  001006                            BNE     10$             ;BR IF INCORRECT.
027136  077303                            SOB     R3,2$

027140  026104  000016                    CMP     16(R1),R4       ;SEE IF THE FPS WAS CORRECT.
027144  001020                            BNE     15$             ;BR IF INCORRECT.
027146  000161  000030            3$:     JMP     30(R1)          ;RETURN.

                                ;RESULT IN CORRECT SO SEE IF THE FAILURE WAS ANTICIPATED.
027152  012702  027224            10$:    MOV     #LDCT,R2
027156  010100                            MOV     R1,R0
027160  062700  000010                    ADD     #10,R0
027164  012703  000002                    MOV     #2,R3
027170  022022                  11$:      CMP     (R0)+,(R2)+
027172  001003                            BNE     13$
027174  077303                            SOB     R3,11$
027176  000161  000022                    JMP     22(R1)

                                ;THE FAILURE WAS NOT ANTICIPATED SO REPORT THE ERROR HERE.
027202                          13$:
```

```
 5017 027202  104260                14$:    ERROR   +260            ;BAD RES
 5018 027204  000760                        BR      3$
 5019
 5020
 5021                               ;THE FPS WAS INCORRECT SO SEE IF IT WAS ANTICIPATED.
 5022 027206  026104  000020        15$:    CMP     20(R1),R4
 5023 027212  001002                        BNE     16$
 5024 027214  000161  000026              JMP     26(R1)
 5025
 5026                               ;FPS ERROR NOT ANTICIPATED SO REPORT IT HERE.
 5027 027220                        16$:
 5028 027220  104261                17$:    ERROR   +261            ;BAD FPS
 5029 027222  000751                        BR      3$
 5030
 5031                               ;DATA BUFFER:
 5032 027224  000000  000000  000000 LDCT:   .WORD   0,0,0,0
      027232  000000
 5033
 5034 027234                        KKCDONE:
      027234  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                             ;THE USER TYPED CONTROL G?).
 5035
```

```
                                ;;*****************************************************************
                                ;*TEST 57       LDCID AND LDCLD TEST
                                ;*
                                ;* THIS IS A TEST OF LDCID AND LDCLD
                                ;*
                                ;;*****************************************************************
027236  000004                  TST57:  SCOPE
                                ;OPERAND=0      FL=0,   FD=1
                                LLC1:
027240  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
027242  004737  030036                   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
027246  000000  000000          1$:     .WORD   0,0             ;FSRC OPERAND.
027252  000000  000000  000000  2$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
027260  000000
027262  177777  177777  177777  3$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
027270  177777
027272  000213                  4$:     213                     ;FPS BEFORE EXECUTION.
027274  000204                          204                     ;FPS AFTER EXECUTION.
027276  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
027300  104273                  5$:     ERROR   +273            ;REPORT RESULT INCORRECT.
027302  000401                          BR      6$
027304  104274                          ERROR   +274            ;REPORT FPS INCORRECT.
027306                          6$:
                                ;OPERAND=0      FL=0,   FD=1
                                LLC2:
027306  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
027310  004737  030036                   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
027314  000000  177777          1$:     .WORD   0,-1            ;FSRC OPERAND.
027320  000000  000000  000000  2$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
027326  000000
027330  004177  177400  000000  3$:     .WORD   4177,177400,0,0         ;ANTICIPATED ERRONEOUS RESULT.
027336  000000
027340  000200                  4$:     200                     ;FPS BEFORE EXECUTION.
027342  000204                          204                     ;FPS AFTER EXECUTION.
027344  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
027346  104275                  5$:     ERROR   +275            ;(BUT FL)S+277
027350  000401                          BR      6$              ;TO 300 INTO 301
027352  104274                          ERROR   +274            ;REPORT FPS INCORRECT.
027354                          6$:

                                ;OPERAND=0      FL=1    FD=1
027354                          LLC3:
027354  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
027356  004737  030036                   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
027362  000000  000000          1$:     .WORD   0,0             ;FSRC OPERAND.
027366  000000  000000  000000  2$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
027374  000000
027376  177777  177777  177777  3$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
027404  177777
027406  000211                  4$:     211                     ;FPS BEFORE EXECUTION.
027410  000204                          204                     ;FPS AFTER EXECUTION.
027412  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
027414  104273                  5$:     ERROR   +273            ;REPORT RESULT INCORRECT.
027416  000401                          BR      6$
027420  104274                          ERROR   +274            ;REPORT FPS INCORRECT.
027422                          6$:
```

```
5084
5085                                    ;OPERAND=40000  FL=0    FD=1
5086 027422                             LLC4:
     027422  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5087 027424  004737  030036                     JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
5088 027430  040000  000000             1$:     .WORD   40000,0         ;FSRC OPERAND.
5089 027434  043600  000000  000000     2$:     .WORD   43600,0,0,0             ;EXPECTED RESULT.
     027442  000000
5090 027444  047600  000000  000000     3$:     .WORD   47600,0,0,0             ;ANTICIPATED ERRONEOUS RESULT.
     027452  000000
5091 027454  000217                     4$:     217                     ;FPS BEFORE EXECUTION.
5092 027456  000200                             200                     ;FPS AFTER EXECUTION.
5093 027460  177777                             -1                      ;ANTICIPATED ERRONEOUS FPS.
5094 027462  104276                     5$:     ERROR   +276            ;ST 107 BAD CONST
5095 027464  000401                             BR      6$
5096 027466  104274                             ERROR   +274            ;REPORT FPS INCORRECT.
5097 027470                             6$:
5098
5099                                    ;OPERAND=-40000 FL=0    FD=1
5100 027470                             LLC5:
     027470  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5101 027472  004737  030036                     JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
5102 027476  140000  000000             1$:     .WORD   -40000,0                ;FSRC OPERAND.
5103 027502  143600  000000  000000     2$:     .WORD   143600,0,0,0            ;EXPECTED RESULT.
     027510  000000
5104 027512  043600  000000  000000     3$:     .WORD   43600,0,0,0             ;ANTICIPATED ERRONEOUS RESULT.
     027520  000000
5105 027522  000200                     4$:     200                     ;FPS BEFORE EXECUTION.
5106 027524  000210                             210                     ;FPS AFTER EXECUTION.
5107 027526  177777                             -1                      ;ANTICIPATED ERRONEOUS FPS.
5108 027530  104277                     5$:     ERROR   +277            ;(SET SIGN) ST 176
5109 027532  000401                             BR      6$
5110 027534  104274                             ERROR   +274            ;REPORT FPS INCORRECT.
5111 027536                             6$:
5112
5113                                    ;OPERAND=40000,0        FL=1    FD=1
5114 027536                             LLC6:
     027536  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5115 027540  004737  030036                     JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
5116 027544  040000  000000             1$:     .WORD   40000,0         ;FSRC OPERAND.
5117 027550  047600  000000  000000     2$:     .WORD   47600,0,0,0             ;EXPECTED RESULT.
     027556  000000
5118 027560  043600  000000  000000     3$:     .WORD   43600,0,0,0             ;ANTICIPATED ERRONEOUS RESULT.
     027566  000000
5119 027570  000317                             317                     ;FPS BEFORE EXECUTION.
5120 027572  000300                             300                     ;FPS AFTER EXECUTION.
5121 027574  177777                             -1                      ;ANTICIPATED ERRONEOUS FPS.
5122 027576  104300                     5$:     ERROR   +300            ;ST 107 BAD CONS
5123 027600  000401                             BR      6$
5124 027602  104274                             ERROR   +274            ;REPORT FPS INCORRECT.
5125 027604                             6$:
5126
5127                                    ;OPERAND=0,1    FL=1    FD=1
5128 027604                             LLC7:
     027604  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5129 027606  004737  030036                     JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
5130 027612  000000  000001             1$:     .WORD   0,1             ;FSRC OPERAND.
```

```
5131 027616 040200 000000 000000 2$:    .WORD   40200,0,0,0             ;EXPECTED RESULT.
     027624 000000
5132 027626 034200 000000 000000 3$:    .WORD   34200,0,0,0             ;ANTICIPATED ERRONEOUS RESULT.
     027634 000000
5133 027636 000300               4$:    300                     ;FPS BEFORE EXECUTION.
5134 027640 000300                      300                     ;FPS AFTER EXECUTION.
5135 027642 177777                      -1                      ;ANTICIPATED ERRONEOUS FPS.
5136 027644 104300               5$:    ERROR   +300            ;REPORT FPS INCORRECT.
5137 027646 000401                      BR      6$
5138 027650 104274                      ERROR   +274            ;REPORT FPS INCORRECT.
5139 027652                      6$:
5140
5141                             ;OPERAND=77777,177777   FL=1    FD=1
5142 027652                      LLC8:
     027652 104413                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5143 027654 004737 030036               JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
5144 027660 077777 177777       1$:    .WORD   77777,177777            ;FSRC OPERAND.
5145 027664 047777 177777 177000 2$:    .WORD   47777,177777,177000,0           ;EXPECTED RESULT.
     027672 000000
5146 027674 177777 177777 177777 3$:    .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
     027702 177777
5147 027704 000317               4$:    317                     ;FPS BEFORE EXECUTION.
5148 027706 000300                      300                     ;FPS AFTER EXECUTION.
5149 027710 177777                      -1                      ;ANTICIPATED ERRONEOUS FPS.
5150 027712 104273               5$:    ERROR   +273            ;REPORT RESULT INCORRECT.
5151 027714 000401                      BR      6$
5152 027716 104274                      ERROR   +274            ;REPORT FPS INCORRECT.
5153 027720                      6$:
5154
5155                             ;OPERAND=-PATTERN       FL=1    FD=1
5156
5157 027720                      LLC9:
     027720 104413                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5158 027722 004767 000110               JSR     PC,LDCDSUB      ;GO EXECUTE THE INSTRUCTION.
5159 027726 177777 177526       1$:    .WORD   -1,-252         ;FSRC OPERAND.
5160 027732 142052 000000 000000 2$:    .WORD   142052,0,0,0            ;EXPECTED RESULT.
     027740 000000
5161 027742 136052 000000 000000 3$:    .WORD   136052,0,0,0            ;ANTICIPATED ERRONEOUS RESULT.
     027750 000000
5162 027752 000307               4$:    307                     ;FPS BEFORE EXECUTION.
5163 027754 000310                      310                     ;FPS AFTER EXECUTION.
5164 027756 177777                      -1                      ;ANTICIPATED ERRONEOUS FPS.
5165 027760 104300               5$:    ERROR   +300            ;REPORT RESULT INCORRECT.
5166 027762 000401                      BR      6$
5167 027764 104274                      ERROR   +274            ;REPORT FPS INCORRECT.
5168 027766                      6$:
5169
5170                             ;OPERAND=PATTERN        FL=1    FD=1    FT=1
5171 027766                      LLC10:
     027766 104413                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5172 027770 004767 000042               JSR     PC,LDCDSUB      ;GO EXECUTE THE INSTRUCTION.
5173 027774 012345 067012       1$:    .WORD   12345,67012             ;FSRC OPERAND.
5174 030000 047247 025560 050000 2$:    .WORD   47247,025560,050000,0           ;EXPECTED RESULT.
     030006 000000
5175 030010 177777 177777 177777 3$:    .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
     030016 177777
5176 030020 000352               4$:    352                     ;FPS BEFORE EXECUTION.
```

```
5177 030022  000340                        340                     ;FPS AFTER EXECUTION.
5178 030024  177777                        -1                      ;ANTICIPATED ERRONEOUS FPS.
5179 030026  104273                5$:     ERROR   +273            ;REPORT RESULT INCORRECT.
5180 030030  000401                        BR      6$
5181 030032  104274                        ERROR   +274            ;REPORT FPS INCORRECT.
5182 030034  000502                6$:     BR      LLCDONE
5183
5184                               ;THIS SUBROUTINE, LDCDSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
5185                               ;THE LDCID OR LDCLD INSTRUCTION AND CHECK THE RESULTS. A CALL
5186                               ;TO IT IS MADE THUS:
5187                               ;
5188                               ;       JSR     PC,@#LDCDSUB
5189                               ;ACARG: .WORD   X,X             ;AC OPERAND
5190                               ;RES:   .WORD   X,X,X,X         ;EXPECTED RESULT
5191                               ;ERRES: .WORD   X,X,X,X         ;ERROR RESULT
5192                               ;FPSB:  .WORD   X               ;FPS BEFORE EXECUTION
5193                               ;FPSA:  .WORD   X               ;FPS AFTER EXECUTION
5194                               ;ERFPS: .WORD   X               ;ERROR FPS.
5195                               ;ERR1:  ERROR   +X              ;DATA ERROR.
5196                               ;       BR      CONT
5197                               ;ERR2:  ERROR   +X              ;FPS ERROR.
5198                               ;CONT:                          ;RETURN ADDRESS
5199                               ;
5200                               ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
5201                               ;THE LDCID OR LDCLD INSTRUCTION IS EXECUTED.
5202                               ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
5203                               ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCDSUB RETURNS CONTROL
5204                               ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCDSUB
5205                               ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCDSUB WILL RETURN
5206                               ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCDSUB ITSELF
5207                               ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
5208                               ;LDCID OR LDCLD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
5209                               ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
5210                               ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCDSUB
5211                               ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
5212                               ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCDSUB WILL
5213                               ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
5214
5215 030036  012601                LDCDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
5216 030040  016100  000024                MOV     24(R1),R0       ;SET THE FPS.
5217 030044  170100                        LDFPS   R0
5218 030046  012737  030056  001236        MOV     #1$,@#$TMP2
5219 030054  010100                        MOV     R1,R0
5220 030056  177010                1$:     LDCID   (R0),AC0        ;TEST INSTRUCTION, LDCID OR LDCLD.
5221
5222 030060  170204                        STFPS   R4              ;GET FPS.
5223 030062  012700  027224                MOV     #LDCT,R0        ;GET THE RESULT.
5224 030066  012702  000200                MOV     #200,R2
5225 030072  170102                        LDFPS   R2
5226 030074  174010                        STD     AC0,(R0)
5227
5228                               ;SEE IF THE RESULT IS CORRECT.
5229 030076  012702  027224                MOV     #LDCT,R2
5230 030102  010237  001242                MOV     R2,@#$TMP4
5231 030106  010137  001240                MOV     R1,@#$TMP3
5232 030112  010103                        MOV     R1,R3
5233 030114  062703  000004                ADD     #4,R3
```

```
 5234 030120  010337  001244                    MOV     R3,@#$TMP5
 5235 030124  010437  001250                    MOV     R4,@#$TMP7
 5236 030130  016137  000026  001252            MOV     26(R1),@#$TMP10
 5237 030136  010100                            MOV     R1,R0
 5238 030140  062700  000004                    ADD     #4,R0
 5239 030144  012703  000002                    MOV     #2,R3
 5240 030150  022022                  2$:       CMP     (R0)+,(R2)+
 5241 030152  001006                            BNE     10$                     ;BR IF INCORRECT.
 5242 030154  077303                            SOB     R3,2$
 5243
 5244 030156  026104  000026                    CMP     26(R1),R4                       ;IS THE FPS CORRECT?
 5245 030162  001020                            BNE     15$                     ;BR IF INCORRECT.
 5246 030164  000161  000040            3$:     JMP     40(R1)                  ;RETURN.
 5247
 5248                                ;THE RESULT WAS INCORRECT SO SEE IF THE ERROR WAS ANTICIPATED.
 5249 030170  012702  027224          10$:     MOV     #LDCT,R2
 5250 030174  010100                            MOV     R1,R0
 5251 030176  062700  000014                    ADD     #14,R0
 5252 030202  012703  000002                    MOV     #2,R3
 5253 030206  022022                  11$:     CMP     (R0)+,(R2)+
 5254 030210  001003                            BNE     13$
 5255 030212  077303                            SOB     R3,11$
 5256 030214  000161  000032                    JMP     32(R1)
 5257 030220                          13$:
 5258                                ;ERROR NOT ANTICIPATED SO REPORT RESULT INCORRECT HERE.
 5259 030220  104273                  14$:     ERROR   +273                    ;BAD RES
 5260 030222  000760                            BR      3$
 5261
 5262                                ;THE FPS WAS INCORRECT. SEE IF FAILURE WAS ANTICIPATED.
 5263 030224  026104  000030          15$:     CMP     30(R1),R4
 5264 030230  001002                            BNE     16$
 5265 030232  000161  000036                    JMP     36(R1)
 5266                                ;FPS ERROR WAS NOT ANTICIPATED SO REPORT FAILURE HERE.
 5267 030236                          16$:
 5268
 5269 030236  104274                  17$:     ERROR   +274                    ;BAD FPS
 5270 030240  000751                            BR      3$
 5271
 5272 030242                          LLCDONE:
      030242  104412                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                          ;SEE IF THE USER HAS EXPRESSED
                                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                          ;THE USER TYPED CONTROL G?).
 5273
 5282
 5283                               ;;*****************************************************************
                                    ;*TEST 60       LDEXP TEST
                                    ;*
                                    ;* THIS IS A TEST OF THE LDEXP INST
                                    ;* A SUBROUTINE IS USED TO SET UP
                                    ;* OPERANDS, EXECUTE THE LDEXP INST AND
                                    ;* CHECK THE RESULTS.
                                    ;*
                                    ;;*****************************************************************
      030244  000004                TST60:  SCOPE
 5284
```

```
5285                                    ; NON-ZERO RES. VALID EXPON=210 (EXCESS 200)=10
5286 030246                             MMC1:
     030246  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5287 030250  004767  001334                    JSR     PC,LDXSUB       ;GO EXECUTE THE INSTRUCTION.
5288 030254  012345  067012  034567  1$:      .WORD   12345,67012,34567,012345        ;AC0 OPERAND.
     030262  012345
5289 030264  000010                  2$:      .WORD   10              ;EXPONENT OPERAND.
5290 030266  042145  067012  034567  3$:      .WORD   42145,67012,34567,012345                ;EXPECTED RESULT.
     030274  012345
5291 030276  002145  067012  034567  4$:      .WORD   2145,67012,34567,012345         ;ANTICIPATED ERRONEOUS RESULT.
     030304  012345
5292 030306  047217                  5$:      47217                   ;FPS BEFORE EXECUTION.
5293 030310  047200                           47200                   ;FPS AFTER EXECUTION.
5294 030312  147200                           147200                  ;ANTICIPATED ERRONEOUS FPS.
5295 030314  177777                           -1              ;EXPECTED FEC.
5296 030316  104304                  6$:      ERROR   +304            ;E12+E12+200 BAD
5297 030320  000400                           BR      7$              ;ST 624
5298 030322  104305                  7$:      ERROR   +305            ;REPORT FPS INCORRECT.
5299                                                                    ;ST 625 INTO 304
5300                                    ;NON-ZERO       RES     NEG.
5301 030324                             MMC2:
     030324  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5302 030326  004737  031610                    JSR     PC,@#LDXSUB     ;EXPON=377
5303 030332  123456  070123  045670  1$:      .WORD   123456,70123,45670,123456       ;AC0 OPERAND.
     030340  123456
5304 030342  000177                  2$:      .WORD   177             ;EXPONENT OPERAND.
5305 030344  177656  070123  045670  3$:      .WORD   177656,70123,45670,123456               ;EXPECTED RESULT.
     030352  123456
5306 030354  137656  070123  045670  4$:      .WORD   137656,70123,45670,123456               ;ANTICIPATED ERRONEOUS RESULT.
     030362  123456
5307 030364  047207                  5$:      47207                   ;FPS BEFORE EXECUTION.
5308 030366  047210                           47210                   ;FPS AFTER EXECUTION.
5309 030370  147210                           147210                  ;ANTICIPATED ERRONEOUS FPS.
5310 030372  177777                           -1              ;EXPECTED FEC.
5311 030374  104304                  6$:      ERROR   +304            ;REPORT RESULT INCORRECT.
5312 030376  000401                           BR      7$
5313 030400  104305                           ERROR   +305            ;REPORT FPS INCORRECT.
5314 030402                             7$:
5315
5316                                    ;NON-ZERO       RES,    EXP=256=(56)REAL
5317 030402                             MMC3:
     030402  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5318 030404  004737  031610                    JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
5319 030410  073261  057645  043323  1$:      .WORD   73261,057645,43323,101760       ;AC0 OPERAND.
     030416  101760
5320 030420  000056                  2$:      .WORD   56              ;EXPONENT OPERAND.
5321 030422  053461  057645  043323  3$:      .WORD   53461,057645,43323,101760               ;EXPECTED RESULT.
     030430  101760
5322 030432  177777  177777  177777  4$:      .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
     030440  177777
5323 030442  047200                  5$:      47200                   ;FPS BEFORE EXECUTION.
5324 030444  047200                           47200                   ;FPS AFTER EXECUTION.
5325 030446  147200                           147200                  ;ANTICIPATED ERRONEOUS FPS.
5326 030450  177777                           -1              ;EXPECTED FEC.
5327 030452  104301                  6$:      ERROR   +301            ;REPORT RESULT INCORRECT.
5328 030454  000401                           BR      7$
5329 030456  104305                           ERROR   +305            ;REPORT FPS INCORRECT.
```

```
5330 030460                          7$:
5331
5332                                 ;EXP=27 (EXCESS 200)=-151 (OCT)
5333 030460                          MMC4:
     030460  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5334 030462  004737  031610                  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
5335 030466  012223  024252  062720  1$:     .WORD   12223,24252,62720,21222 ;AC0 OPERAND.
     030474  021222
5336 030476  177627                  2$:     .WORD   -151            ;EXPONENT OPERAND.
5337 030500  005623  024252  062720  3$:     .WORD   5623,24252,62720,21222          ;EXPECTED RESULT.
     030506  021222
5338 030510  177777  177777  177777  4$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
     030516  177777
5339 030520  047200                  5$:     47200                   ;FPS BEFORE EXECUTION.
5340 030522  047200                          47200                   ;FPS AFTER EXECUTION.
5341 030524  147200                          147200                  ;ANTICIPATED ERRONEOUS FPS.
5342 030526  177777                          -1              ;EXPECTED FEC.
5343 030530  104301                  6$:     ERROR   +301            ;REPORT RESULT INCORRECT.
5344 030532  000401                          BR      7$
5345 030534  104306                          ERROR   +306            ;(BUT EZBT) ST 544 TO 504 INTO 704 0 (BUT EXBT) ST 704 INTO
5346 030536                          7$:
5347
5348                                 ;EXP=0 (EXCESS 200)=-200 (OCT), POSITIVE FRAC
5349                                 ; FIV=1
5350 030536                          MMC5:
     030536  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5351 030540  004737  031610                  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
5352 030544  030131  032334  035363  1$:     .WORD   30131,32334,35363,73031 ;AC0 OPERAND.
     030552  073031
5353 030554  177600                  2$:     .WORD   -200            ;EXPONENT OPERAND.
5354 030556  000131  032334  035363  3$:     .WORD   00131,32334,35363,73031         ;EXPECTED RESULT.
     030564  073031
5355 030566  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
     030574  000000
5356 030576  042200                  5$:     42200                   ;FPS BEFORE EXECUTION.
5357 030600  142204                          142204                  ;FPS AFTER EXECUTION.
5358 030602  042202                          42202                   ;ANTICIPATED ERRONEOUS FPS.
5359 030604  000012                          12              ;EXPECTED FEC.
5360 030606  104307                  6$:     ERROR   +307            ;(BUT EXBT) ST 704 TO 64 INST 264
5361 030610  000401                          BR      7$
5362 030612  104310                          ERROR   +310    ;(BUT FIU) ST 264 X
5363 030614                          7$:
5364
5365                                 ;EXP=0  (EXCESS 200)=-200 (OCT), NEG FRACT,FIU=1
5366 030614                          MMC6:
     030614  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5367 030616  004737  031610                  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
5368 030622  140414  024344  045464  1$:     .WORD   140414,24344,45464,74045        ;AC0 OPERAND.
     030630  074045
5369 030632  177600                  2$:     .WORD   -200            ;EXPONENT OPERAND.
5370 030634  100014  024344  045464  3$:     .WORD   100014,24344,45464,74045        ;-0             ;EXPECTED RESULT.
     030642  074045
5371 030644  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
     030652  000000
5372 030654  042200                  5$:     42200                   ;FPS BEFORE EXECUTION.
5373 030656  142214                          142214                  ;FPS AFTER EXECUTION.
5374 030660  042214                          42214                   ;ANTICIPATED ERRONEOUS FPS.
```

```
5375 030662  000012                          12                  ;EXPECTED FEC.
5376 030664  104307                  6$:     ERROR   +307        ;REPORT RESULT INCORRECT.
5377 030666  000401                          BR      7$
5378 030670  104310                          ERROR   +310        ;REPORT FPS INCORRECT.
5379 030672                          7$:
5380
5381                                 ;EXP=0 (EXCESS 200)=-200 (OCT),POS FRAC, FIU=0
5382
5383 030672                          MMC7:
     030672  104413                          LPERR               ;SET UP THE LOOP ON ERROR ADDRESS.
5384 030674  004737  031610                  JSR     PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5385 030700  051525  035455  005675  1$:     .WORD   51525,35455,5675,05152  ;AC0 OPERAND.
     030706  005152
5386 030710  177600                  2$:     .WORD   -200        ;EXPONENT OPERAND.
5387 030712  000000  000000  000000  3$:     .WORD   0,0,0,0     ;EXPECTED RESULT.
     030720  000000
5388 030722  000125  035455  005675  4$:     .WORD   00125,35455,5675,05152      ;ANTICIPATED ERRONEOUS RESULT.
     030730  005152
5389 030732  045200                          45200               ;FPS BEFORE EXECUTION.
5390 030734  045204                          45204               ;FPS AFTER EXECUTION.
5391 030736  145204                          145204              ;ANTICIPATED ERRONEOUS FPS.
5392 030740  177777                          -1                  :EXPECTED FEC.
5393 030742  104311                  6$:     ERROR   +311        ;(BUT FIU) ST 264 X          ;REPORT RESULT INCORRECT.
5394 030744  000401                          BR      7$
5395 030746  104302                          ERROR   +302        ;REPORT FPS INCORRECT.
5396 030750                          7$:
5397
5398                                 ;EXP=-1405 (EXCESS 200)=-1605 (OCT), FIU=1
5399 030750                          MMC8:
     030750  104413                          LPERR               ;SET UP THE LOOP ON ERROR ADDRESS.
5400 030752  004737  031610                  JSR     PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5401 030756  061626  062636  046566  1$:     .WORD   61626,62636,46566,67606 ;AC0 OPERAND.
     030764  067606
5402 030766  176173                  2$:     .WORD   -1605       ;EXPONENT OPERAND.
5403 030770  076626  062636  046566  3$:     .WORD   76626,62636,46566,67606     ;EXPECTED RESULT.
     030776  067606
5404 031000  000000  000000  000000  4$:     .WORD   0,0,0,0     ;ANTICIPATED ERRONEOUS RESULT.
     031006  000000
5405 031010  042200                  5$:     42200               ;FPS BEFORE EXECUTION.
5406 031012  142200                          142200              ;FPS AFTER EXECUTION.
5407 031014  042204                          42204               ;ANTICIPATED ERRONEOUS FPS.
5408 031016  000012                          12                  ;EXPECTED FEC.
5409 031020  104312                  6$:     ERROR   +312        ;(BUT EZBT) ST 544 TO 704 INTO 504
5410 031022  000401                          BR      7$
5411 031024  104302                          ERROR   +302        ;REPORT FPS INCORRECT.
5412 031026                          7$:
5413                                 ;EXP=-17416 (EXCESS 200)=-17616 (OCT), FIU=0
5414 031026                          MMC9:
     031026  104413                          LPERR               ;SET UP THE LOOP ON ERROR ADDRESS.
5415 031030  004737  031610                  JSR     PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5416 031034  071727  037475  076777  1$:     .WORD   71727,37475,76777,17273 ;AC0 OPERAND.
     031042  017273
5417 031044  160162                  2$:     .WORD   -17616      ;EXPONENT OPERAND.
5418 031046  000000  000000  000000  3$:     .WORD   0,0,0,0     ;EXPECTED RESULT.
     031054  000000
5419 031056  074527  037475  076777  4$:     .WORD   74527,37475,76777,17273      ;ANTICIPATED ERRONEOUS RESULT.
     031064  017273
```

```
5420 031066  045200                    5$:     45200                   ;FPS BEFORE EXECUTION.
5421 031070  045204                            45204                   ;FPS AFTER EXECUTION.
5422 031072  145200                            145200                  ;ANTICIPATED ERRONEOUS FPS.
5423 031074  177777                            -1              ;EXPECTED FEC.
5424 031076  104313                    6$:     ERROR   +313            ;(BUT FIU) ST 504
5425 031100  000401                            BR      7$
5426 031102  104302                            ERROR   +302            ;REPORT FPS INCORRECT.
5427 031104                            7$:
5428
5429                                   ;EXP=-1601 (EXCESS 200)=-2001 (OCT), FIU=1
5430 031104                            MMC10:
     031104  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5431 031106  004737  031610                    JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
5432 031112  001020  030405  006070    1$:     .WORD   01020,30405,06070,00102 ;AC0 OPERAND.
     031120  000102
5433 031122  175777                    2$:     .WORD   -2001           ;EXPONENT OPERAND.
5434 031124  037620  030405  006070    3$:     .WORD   37620,30405,06070,00102         ;EXPECTED RESULT.
     031132  000102
5435 031134  000000  000000  000000    4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
     031142  000000
5436 031144  042200                    5$:     42200                   ;FPS BEFORE EXECUTION.
5437 031146  142200                            142200                  ;FPS AFTER EXECUTION.
5438 031150  042204                            42204                   ;ANTICIPATED ERRONEOUS FPS.
5439 031152  000012                            12              ;EXPECTED FEC.
5440 031154  104312                    6$:     ERROR   +312            ;(BUT FIU) ST 504
5441 031156  000401                            BR      7$
5442 031160  104302                            ERROR   +302            ;REPORT FPS INCORRECT.
5443 031162                            7$:
5444
5445                                   ;EXP=1206 (EXCESS 200)=1006 (OCT) FIV =1
5446 031162                            MMC11:
     031162  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5447 031164  004737  031610                    JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
5448 031170  012131  014151  016171    1$:     .WORD   12131,14151,16171,10111 ;AC0 OPERAND.
     031176  010111
5449 031200  001006                    2$:     .WORD   1006            ;EXPONENT OPERAND.
5450 031202  041531  014151  016171    3$:     .WORD   41531,14151,16171,10111         ;EXPECTED RESULT.
     031210  010111
5451 031212  000000  000000  000000    4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
     031220  000000
5452 031222  041200                    5$:     41200                   ;FPS BEFORE EXECUTION.
5453 031224  141202                            141202                  ;FPS AFTER EXECUTION.
5454 031226  041204                            41204                   ;ANTICIPATED ERRONEOUS FPS.
5455 031230  000010                            10              ;EXPECTED FEC.
5456 031232  104314                    6$:     ERROR   +314            ;(BUT FIV) ST 104
5457 031234  000401                            BR      7$
5458 031236  104302                            ERROR   +302            ;REPORT FPS INCORRECT.
5459 031240                            7$:
5460
5461                                   ;EXP=16315 (EXCESS 200)=16115 (OCT) FIV=0
5462 031240                            MMC12:
     031240  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5463 031242  004737  031610                    JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
5464 031246  027262  025242  023222    1$:     .WORD   27262,25242,23222,21202 ;AC0 OPERAND.
     031254  021202
5465 031256  016115                    2$:     .WORD   16115           ;EXPONENT OPERAND.
5466 031260  000000  000000  000000    3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
```

```
                031266  000000
 5467 031270    063262  025242  023222  4$:     .WORD   63262,25242,23222,21202          ;ANTICIPATED ERRONEOUS RESULT.
                031276  021202
 5468 031300    046200                  5$:     46200                   ;FPS BEFORE EXECUTION.
 5469 031302    046206                          46206                   ;FPS AFTER EXECUTION.
 5470 031304    146202                          146202                  ;ANTICIPATED ERRONEOUS FPS.
 5471 031306    177777                          -1              ;EXPECTED FEC.
 5472 031310    104315                  6$:     ERROR   +315            ;(BUT FIV) ST 104
 5473 031312    000401                          BR      7$
 5474 031314    104302                          ERROR   +302            ;REPORT FPS INCORRECT.
 5475 031316                            7$:
 5476
 5477                                   ;EXP=11011 (EXCESS 200)=10611 (OCT) FIV=1
 5478
 5479 031316                            MMC13:
                031316  104413                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5480 031320    004737  031610                  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
 5481 031324    030313  032333  034353  1$:     .WORD   30313,32333,34353,36373 ;AC0 OPERAND.
                031332  036373
 5482 031334    010611                  2$:     .WORD   10611           ;EXPONENT OPERAND.
 5483 031336    002313  032333  034353  3$:     .WORD   2313,32333,34353,36373          ;EXPECTED RESULT.
                031344  036373
 5484 031346    000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
                031354  000000
 5485 031356    041200                  5$:     41200                   ;FPS BEFORE EXECUTION.
 5486 031360    141202                          141202                  ;FPS AFTER EXECUTION.
 5487 031362    041204                          41204                   ;ANTICIPATED ERRONEOUS FPS.
 5488 031364    000010                          10              ;EXPECTED FEC.
 5489 031366    104316                  6$:     ERROR   +316            ;(BUT FIV) ST 144
 5490 031370    000401                          BR      7$
 5491 031372    104302                          ERROR   +302            ;REPORT FPS INCORRECT.
 5492 031374                            7$:
 5493
 5494                                   ;EXP=17123 (EXCESS 200)=16723 (OCT) FIV=0
 5495
 5496 031374                            MMC14:
                031374  104413                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5497 031376    004737  031610                  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
 5498 031402    040414  042434  044454  1$:     .WORD   40414,42434,44454,46474 ;AC0 OPERAND.
                031410  046474
 5499 031412    016723                  2$:     .WORD   16723           ;EXPONENT OPERAND.
 5500 031414    000000  000000  000000  3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
                031422  000000
 5501 031424    024614  042434  044454  4$:     .WORD   24614,42434,44454,46474         ;ANTICIPATED ERRONEOUS RESULT.
                031432  046474
 5502 031434    046200                  5$:     46200                   ;FPS BEFORE EXECUTION.
 5503 031436    046206                          46206                   ;FPS AFTER EXECUTION.
 5504 031440    146202                          146202                  ;ANTICIPATED ERRONEOUS FPS.
 5505 031442    177777                          -1              ;EXPECTED FEC.
 5506 031444    104317                  6$:     ERROR   +317            ;(BUT FIV) ST 144
 5507 031446    000401                          BR 7$
 5508 031450    104302                          ERROR   +302            ;REPORT FPS INCORRECT.
 5509 031452                            7$:
 5510
 5511                                   ;EXP=   254 (OCT)=      454 (EXCESS 200)        FIV=1
 5512
 5513 031452                            MMC15:
```

```
        031452  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
  5514  031454  004737  031610                  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
  5515  031460  050515  052535  054555  1$:     .WORD   50515,52535,54555,56575 ;AC0 OPERAND.
        031466  056575
  5516  031470  000254                  2$:     .WORD   254             ;EXPONENT OPERAND.
  5517  031472  013115  052535  054555  3$:     .WORD   13115,52535,54555,56575         ;EXPECTED RESULT.
        031500  056575
  5518  031502  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
        031510  000000
  5519  031512  041200                  5$:     41200                   ;FPS BEFORE EXECUTION.
  5520  031514  141202                          141202                  ;FPS AFTER EXECUTION.
  5521  031516  041204                          41204                   ;ANTICIPATED ERRONEOUS FPS.
  5522  031520  000010                          10              ;EXPECTED FEC.
  5523  031522  104320                  6$:     ERROR   +320                    ;(BUT FIV) ST344
  5524  031524  000401                          BR      7$
  5525  031526  104302                          ERROR   +302            ;REPORT FPS INCORRECT.
  5526  031530                          7$:
  5527
  5528                                  ;EXP=   313 (OCT)=      513(EXCESS 200) FIV=0
  5529
  5530  031530                          MMC16:
        031530  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
  5531  031532  004737  031610                  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
  5532  031536  060616  062636  064656  1$:     .WORD   60616,62636,64656,66676 ;AC0 OPERAND.
        031544  066676
  5533  031546  000313                  2$:     .WORD   313             ;EXPONENT OPERAND.
  5534  031550  000000  000000  000000  3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
        031556  000000
  5535  031560  022616  062636  064656  4$:     .WORD   22616,62636,64656,66676         ;ANTICIPATED ERRONEOUS RESULT.
        031566  066676
  5536  031570  046200                  5$:     46200                   ;FPS BEFORE EXECUTION.
  5537  031572  046206                          46206                   ;FPS AFTER EXECUTION.
  5538  031574  146202                          146202                  ;ANTICIPATED ERRONEOUS FPS.
  5539  031576  177777                          -1              ;EXPECTED FEC.
  5540  031600  104321                  6$:     ERROR   +321                    ;(BUT FIV) ST 344
  5541  031602  000401                          BR      7$
  5542  031604  104302                          ERROR   +302            ;REPORT FPS INCORRECT.
  5543  031606                          7$:
  5544  031606  000540                          BR      MMCDONE
  5545
  5546                                  ;THIS SUBROUTINE, LDXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
  5547                                  ;THE LDEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
  5548                                  ;TO IT IS MADE THUS:
  5549                                  ;
  5550                                  ;               JSR     PC,@#LDXSUB
  5551                                  ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
  5552                                  ;               EXP:    .WORD   X               ;EXPONENT
  5553                                  ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
  5554                                  ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
  5555                                  ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
  5556                                  ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
  5557                                  ;               ERFPS:  .WORD   X               ;ERROR FPS.
  5558                                  ;               FEC:    .WORD   X               ;EXPECTED FEC
  5559                                  ;               ERR1:   ERROR   +X              ;DATA ERROR.
  5560                                  ;                       BR      CONT
  5561                                  ;               ERR2:   ERROR   +X              ;FPS ERROR.
  5562                                  ;               CONT:                           ;RETURN ADDRESS
```

```
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                        ;THE LDEXP INSTRUCTION IS EXECUTED.
                                        ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                        ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDXSUB RETURNS CONTROL
                                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDXSUB
                                        ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDXSUB WILL RETURN
                                        ;TO THE ERROR CALL AT ERR2, OTHERWISE LDXSUB ITSELF
                                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                        ;LDEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                        ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                        ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDXSUB
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDXSUB WILL
                                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

031610  012601                  LDXSUB: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
031612  012700  000200                  MOV     #200,R0         ;LOAD THE AC0 OPERAND.
031616  170100                          LDFPS   R0
031620  010100                          MOV     R1,R0
031622  172410                          LDD     (R0),AC0
031624  012737  031646  001236          MOV     #1$,@#$TMP2
031632  016100  000032                  MOV     32(R1),R0       ;SET UP THE FPS.
031636  170100                          LDFPS   R0
031640  010100                          MOV     R1,R0
031642  062700  000010                  ADD     #10,R0

031646  176410                  1$:     LDEXP   (R0),AC0        ;TEST INSTRUCTION.

031650  170204                          STFPS   R4              ;GET THE FPS.
031652  170305                          STST    R5              ;GET THE FEC.
031654  012700  000200                  MOV     #200,R0         ;GET THE RESULT.
031660  170100                          LDFPS   R0
031662  012700  032100                  MOV     #LDXT,R0
031666  174010                          STD     AC0,(R0)
031670  010437  001250                  MOV     R4,@#$TMP7
031674  016137  000034  001252          MOV     34(R1),@#$TMP10
031702  010537  001254                  MOV     R5,@#$TMP11
031706  016137  000040  001256          MOV     40(R1),@#$TMP12
031714  010102                          MOV     R1,R2
031716  010237  001240                  MOV     R2,@#$TMP3
031722  062702  000010                  ADD     #10,R2
031726  011237  001242                  MOV     (R2),@#$TMP4
031732  062702  000002                  ADD     #2,R2
031736  010237  001244                  MOV     R2,@#$TMP5
031742  012737  032100  001246          MOV     #LDXT,@#$TMP6
031750  012702  032100                  MOV     #LDXT,R2        ;SEE IF THE RESULT WAS CORRECT.
031754  010103                          MOV     R1,R3
031756  062703  000012                  ADD     #12,R3
031762  012700  000004                  MOV     #4,R0
031766  022223                  2$:     CMP     (R2)+,(R3)+
031770  001014                          BNE     10$             ;BRANCH IF NOT CORRECT.
031772  077003                          SOB     R0,2$
031774  020461  000034                  CMP     R4,34(R1)       ;SEE IF THE FPS WAS CORRECT.
032000  001026                          BNE     15$             ;BRANCH IF NOT CORRECT.
032002  005761  000034                  TST     34(R1)
032006  100003                          BPL     3$
```

```
5620 032010  020561  000040                CMP     R5,40(R1)       ;SEE IF THE FEC WAS CORRECT.
5621 032014  001027                        BNE     20$             ;BRANCH IF NOT CORRECT.
5622
5623 032016  000161  000050          3$:   JMP     50(R1)          ;RETURN.
5624
5625                                ;THE RESULT WAS INCORRECT SO SEE IF THE FAILURE WAS ANTICIPATED.
5626 032022  012702  032100          10$:  MOV     #LDXT,R2
5627 032026  010103                        MOV     R1,R3
5628 032030  062703  000022                ADD     #22,R3
5629 032034  012700  000004                MOV     #4,R0
5630 032040  022223                  11$:  CMP     (R2)+,(R3)+
5631 032042  001003                        BNE     12$
5632 032044  077003                        SOB     R0,11$
5633 032046  000161  000042                JMP     42(R1)
5634
5635                                ;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE.
5636 032052                         12$:
5637 032052  104301                  13$:  ERROR   +301                    ;BAD RES
5638 032054  000760                        BR      3$
5639
5640                                ;SEE IF THE FPS ERROR WAS ANTICIPATED.
5641 032056  026104  000036          15$:  CMP     36(R1),R4
5642 032062  001002                        BNE     16$
5643 032064  000161  000046                JMP     46(R1)
5644 032070                         16$:
5645                                ;THE FPS WAS NOT ANTICIPATED SO REPORT IT HERE.
5646 032070  104302                  17$:  ERROR   +302                    ;BAD FPS
5647 032072  000751                        BR      3$                      ;BUT EZBTY8
5648                                                                        ;ST 063
5649
5650 032074                         20$:
5651                                ;REPORT FEC INCORRECT.
5652 032074  104303                  21$:  ERROR   +303                    ;BAD FEC
5653 032076  000747                        BR      3$
5654
5655                                ;DATA BUFFER:
5656 032100  000000  000000  000000 LDXT:   .WORD   0,0,0,0
     032106  000000
5657
5658 032110                         MMCDONE:
     032110  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
5659
5660
5667
5668                                ;;*****************************************************************
                                    ;*TEST 61       DESTINATION MODES, MODE 1 (FL=0), TEST
                                    ;*
                                    ;* THIS IS A TEST OF DESTINATION MODE 1 USING
                                    ;* THE STFPS INSTRUCTION
                                    ;*
                                    ;;*****************************************************************
     032112  000004                 TST61:  SCOPE
5669
```

```
5670
5671 032114                          NNC1:
     032114  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5672 032116  012700  032214                   MOV     #NNCTB0,R0      ;SET UP THE DATA BUFFER.
5673 032122  012701  000006                   MOV     #6,R1
5674 032126  012720  177777           1$:     MOV     #-1,(R0)+
5675 032132  077103                          SOB     R1,1$
5676 032134  012700  102345                   MOV     #102345,R0
5677 032140  012737  032162  001236          MOV     #NNC2,@#$TMP2
5678 032146  012737  032314  000004          MOV     #NNC25,@#ERRVECT ;SET UP FOR TRAPS TO 4.
5679 032154  170100                          LDFPS   R0              ;SET UP FPS.
5680 032156  012700  032220                   MOV     #NNCTB1,R0
5681
5682 032162  170210                   NNC2:   STFPS   (R0)            ;TEST INSTRUCTION.
5683 032164  020027  032220                   CMP     R0,#NNCTB1      ;IS R0 CORRECT?
5684 032170  001017                          BNE     NNC10           ;BRANCH IF NOT CORRECT.
5685 032172  023727  032220  102345          CMP     @#NNCTB1,#102345   ;IS RESULT CORRECT?
5686 032200  001023                          BNE     NNC15           ;BRANCH IF NOT CORRECT.
5687 032202  023727  032222  177777          CMP     @#NNCTB1+2,#-1  ;IS THE RESULT CORRECT?
5688 032210  001030                          BNE     NNC20           ;BRANCH IF NOT CORRECT.
5689 032212  000453                          BR      NNCDONE
5690
5691                                   ;TEST DATA BUFFER:
```

```
 5693 032214  177777  177777          NNCTB0: .WORD   -1,-1
 5694 032220  177777  177777  177777  NNCTB1: .WORD   -1,-1,-1,-1
      032226  177777
 5695
 5696                                 ;REPORT R0 INCORRECT.
 5697 032230  010037  001242          NNC10:  MOV     R0,@#$TMP4
 5698 032234  012737  032220  001240          MOV     #NNCTB1,@#$TMP3
 5699 032242                          1$:
      032242  104377                          ERROR   +377
      032244  000001                          .WORD   1
 5700                                                                ;R0 BAD (BUT
 5701 032246  000435                          BR      NNCDONE        ;    FDST)X
 5702
 5703                                 ;REPORT RESULT INCORRECT.
 5704 032250  012737  102345  001240  NNC15:  MOV     #102345,@#$TMP3          ;    ST 634
 5705 032256  013737  032220  001242          MOV     @#NNCTB1,@#$TMP4
 5706 032264                          1$:
      032264  104377                          ERROR   +377
      032266  000002                          .WORD   2
 5707                                                                ;BAD DATA
 5708 032270  000424                          BR      NNCDONE
 5709
 5710
 5711                                 ;REPORT RESULT INCORRECT.
 5712 032272  012737  177777  001240  NNC20:  MOV     #-1,@#$TMP3
 5713 032300  013737  032222  001242          MOV     @#NNCTB1+2,@#$TMP4
 5714 032306                          1$:
      032306  104377                          ERROR   +377
      032310  000003                          .WORD   3
 5715                                                                ;(BUT GR7,FL)
 5716 032312  000413                          BR      NNCDONE        ;ST 357 TO 416
 5717                                                                ;INTO 417
 5718
 5719                                 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
 5720                                 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
 5721                                 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
 5722 032314  011604                  NNC25:  MOV     (SP),R4
 5723 032316  020427  032164                  CMP     R4,#NNC2+2
 5724 032322  001402                          BEQ     1$
 5725 032324  000137  046250                  JMP     @#CPSPUR
 5726
 5727 032330  011637  001236          1$:     MOV     (SP),@#$TMP2
 5728 032334  022626                          CMP     (SP)+,(SP)+
 5729 032336                          2$:
      032336  104377                          ERROR   +377
      032340  000004                          .WORD   4
 5730                                                                ;(BUT FDST)+ ST634
 5731
 5732 032342                          NNCDONE:
      032342  104412                          RSETUP                 ;GO INITIALIZE THE FPS AND STACK; AND
                                                                      ;SEE IF THE USER HAS EXPRESSED
                                                                      ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                      ;THE USER TYPED CONTROL G?).
 5733
 5734
 5735                                 ;;*****************************************************************
```

```
                                        ;*TEST 62       DESTINATION MODES, MODE 2 (FL=0), TEST
                                        ;*
                                        ;* THIS IS A TEST OF DESTINATION MODE 2 USING
                                        ;* THE STFPS INSTRUCTION
                                        ;*
                                        ;;*****************************************************************
     032344  000004                     TST62:  SCOPE
 5736
 5737
 5738 032346                            00C1:
      032346  104413                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5739 032350  012700  032446                    MOV     #00CTB0,R0      ;SET UP THE DATA BUFFER.
 5740 032354  012701  000006                    MOV     #6,R1
 5741 032360  012720  177777            1$:     MOV     #-1,(R0)+
 5742 032364  077103                            SOB     R1,1$
 5743 032366  012700  105412                    MOV     #105412,R0
 5744 032372  012737  032414  001236            MOV     #00C2,@#$TMP2
 5745 032400  012737  032546  000004            MOV     #00C25,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
 5746 032406  170100                            LDFPS   R0              ;SET UP FPS.
 5747 032410  012700  032452                    MOV     #00CTB1,R0
 5748
 5749 032414  170220                    00C2:   STFPS   (R0)+           ;TEST INSTRUCTION.
 5750 032416  020027  032454                    CMP     R0,#00CTB1+2    ;IS R0 CORRECT?
 5751 032422  001017                            BNE     00C10           ;BRANCH IF NOT CORRECT.
 5752 032424  023727  032452  105412            CMP     @#00CTB1,#105412  ;IS THE RESULT CORRECT?
 5753 032432  001023                            BNE     00C15           ;BRANCH IF NOT CORRECT.
 5754 032434  023727  032454  177777            CMP     @#00CTB1+2,#-1  ;IS THE RESULT CORRECT?
 5755 032442  001030                            BNE     00C20           ;BRANCH IF NOT CORRECT.
 5756 032444  000453                            BR      00CDONE
 5757
 5758                                   ;TEST DATA BUFFER:
 5759 032446  177777  177777            00CTB0: .WORD   -1,-1
 5760 032452  177777  177777  177777    00CTB1: .WORD   -1,-1,-1,-1
      032460  177777
 5761
 5762                                   ;REPORT R0 INCORRECT.
 5763 032462  010037  001242            00C10:  MOV     R0,@#$TMP4
 5764 032466  012737  032454  001240            MOV     #00CTB1+2,@#$TMP3
 5765 032474                            1$:
      032474  104377                           ERROR   +377
      032476  000005                           .WORD   5
 5766                                                                   ;R0 BAD (BUT
 5767 032500  000435                            BR      00CDONE         ;     FDST)X
 5768
 5769                                   ;REPORT RESULT INCORRECT.
 5770 032502  012737  105412  001240    00C15:  MOV     #105412,@#$TMP3         ;     ST 634
 5771 032510  013737  032452  001242            MOV     @#00CTB1,@#$TMP4
 5772 032516                            1$:
      032516  104377                           ERROR   +377
      032520  000006                           .WORD   6
 5773                                                                   ;BAD DATA
 5774 032522  000424                            BR      00CDONE
 5775
 5776
 5777                                   ;REPORT RESULT INCORRECT.
 5778 032524  012737  177777  001240    00C20:  MOV     #-1,@#$TMP3
 5779 032532  013737  032454  001242            MOV     @#00CTB1+2,@#$TMP4
```

```
 5780 032540                        1$:
      032540  104377                       ERROR   +377
      032542  000007                       .WORD   7
 5781                                                          ;(BUT GR7,FL)
 5782 032544  000413                  BR      OOCDONE         ;ST 357 TO 416
 5783                                                          ;INTO 417
 5784
 5785                               ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
 5786                               ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
 5787                               ;TO THE SPURIOUS TRAP TO 4 HANDLER.
 5788 032546  011604                OOC25:  MOV     (SP),R4
 5789 032550  020427  032416               CMP     R4,#OOC2+2
 5790 032554  001402                       BEQ     1$
 5791 032556  000137  046250               JMP     @#CPSPUR
 5792
 5793 032562  011637  001236        1$:     MOV     (SP),@#$TMP2
 5794 032566  022626                       CMP     (SP)+,(SP)+
 5795 032570                        2$:
      032570  104377                       ERROR   +377
      032572  000010                       .WORD   10
 5796                                                          ;(BUT FDST)+ ST634
 5797
 5798 032574                        OOCDONE:
      032574  104412                       RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                   ;SEE IF THE USER HAS EXPRESSED
                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                   ;THE USER TYPED CONTROL G?).
 5799
 5800
 5801
 5802                               ;;*****************************************************************
                                    ;*TEST 63       DESTINATION MODES, MODE 4 (FL=0), TEST
                                    ;*
                                    ;* THIS IS A TEST OF DESTINATION MODE 4 USING
                                    ;* THE STFPS INSTRUCTION
                                    ;*
                                    ;;*****************************************************************
      032576  000004                TST63:  SCOPE
 5803
 5804 032600                        PPC1:
      032600  104413                       LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5805 032602  012700  032700               MOV     #PPCTB0,R0      ;SET UP THE DATA BUFFER.
 5806 032606  012701  000006               MOV     #6,R1
 5807 032612  012720  177777        1$:     MOV     #-1,(R0)+
 5808 032616  077103                       SOB     R1,1$
 5809 032620  012700  105555               MOV     #105555,R0
 5810 032624  012737  032646  001236       MOV     #PPC2,@#$TMP2
 5811 032632  012737  033000  000004       MOV     #PPC25,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
 5812 032640  170100                       LDFPS   R0              ;SET UP FPS.
 5813 032642  012700  032706               MOV     #PPCTB1+2,R0
 5814
 5815 032646  170240                PPC2:   STFPS   -(R0)           ;TEST INSTRUCTION.
 5816 032650  020027  032704               CMP     R0,#PPCTB1      ;IS R0 CORRECT?
 5817 032654  001017                       BNE     PPC10           ;BRANCH IF NOT CORRECT.
 5818 032656  023727  032704  105555       CMP     @#PPCTB1,#105555  ;IS THE RESULT CORRECT?
 5819 032664  001023                       BNE     PPC15           ;BRANCH IF NOT CORRECT.
```

```
 5820 032666  023727  032706  177777          CMP     @#PPCTB1+2,#-1  ;IS THE RESULT CORRECT?
 5821 032674  001030                          BNE     PPC20           ;BRANCH IF NOT CORRECT.
 5822 032676  000453                          BR      PPCDONE
 5823
 5824                                 ;TEST DATA BUFFER:
 5825 032700  177777  177777          PPCTB0: .WORD   -1,-1
 5826 032704  177777  177777  177777  PPCTB1: .WORD   -1,-1,-1,-1
      032712  177777
 5827
 5828                                 ;REPORT R0 INCORRECT.
 5829 032714  010037  001242          PPC10:  MOV     R0,@#$TMP4
 5830 032720  012737  032704  001240          MOV     #PPCTB1,@#$TMP3
 5831 032726                          1$:
      032726  104377                          ERROR   +377
      032730  000011                          .WORD   11
 5832                                                                 ;R0 BAD (BUT
 5833 032732  000435                          BR      PPCDONE         ;    FDST)X
 5834
 5835                                 ;REPORT RESULT INCORRECT.
 5836 032734  012737  105555  001240  PPC15:  MOV     #105555,@#$TMP3                 ;    ST 634
 5837 032742  013737  032704  001242          MOV     @#PPCTB1,@#$TMP4
 5838 032750                          1$:
      032750  104377                          ERROR   +377
      032752  000012                          .WORD   12
 5839                                                                 ;BAD DATA
 5840 032754  000424                          BR      PPCDONE
 5841
 5842
 5843                                 ;REPORT RESULT INCORRECT.
 5844 032756  012737  177777  001240  PPC20:  MOV     #-1,@#$TMP3
 5845 032764  013737  032706  001242          MOV     @#PPCTB1+2,@#$TMP4
 5846 032772                          1$:
      032772  104377                          ERROR   +377
      032774  000013                          .WORD   13
 5847                                                                 ;(BUT GR7,FL)
 5848 032776  000413                          BR      PPCDONE         ;ST 357 TO 416
 5849                                                                 ;INTO 417
 5850
 5851                                 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
 5852                                 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
 5853                                 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
 5854 033000  011604                  PPC25:  MOV     (SP),R4
 5855 033002  020427  032650                  CMP     R4,#PPC2+2
 5856 033006  001402                          BEQ     1$
 5857 033010  000137  046250                  JMP     @#CPSPUR
 5858
 5859 033014  011637  001236          1$:     MOV     (SP),@#$TMP2
 5860 033020  022626                          CMP     (SP)+,(SP)+
 5861 033022                          2$:
      033022  104377                          ERROR   +377
      033024  000014                          .WORD   14
 5862                                                                 ;(BUT FDST)+ ST634
 5863
 5864 033026                          PPCDONE:
      033026  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                               ;SEE IF THE USER HAS EXPRESSED
                                               ;THE DESIRE TO CHANGE THE SOFTWARE
```

```
                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                        ;THE USER TYPED CONTROL G?).


                                ;;*********************************************************************
                                ;*TEST 64       DESTINATION MODES, MODE 3 (FL=0), TEST
                                ;*
                                ;* THIS IS A TEST OF DESTINATION MODE 3 USING
                                ;* THE STFPS INSTRUCTION
                                ;*
                                ;;*********************************************************************
 033030  000004                TST64:  SCOPE

                                QQC1:
 033032  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 033034  012700  033136                MOV     #QQCTB0,R0      ;SET UP THE DATA BUFFER.
 033040  012701  000010                MOV     #10,R1
 033044  012720  177777        1$:     MOV     #-1,(R0)+
 033050  077103                        SOB     R1,1$
 033052  012700  106653                MOV     #106653,R0
 033056  012737  033104  001236        MOV     #QQC2,@#$TMP2
 033064  012737  033242  000004        MOV     #QQC25,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
 033072  170100                        LDFPS   R0              ;SET UP FPS.
 033074  012700  033152                MOV     #QQCTB2,R0
 033100  012710  033142                MOV     #QQCTB1,(R0)

 033104  170230                QQC2:   STFPS   @(R0)+          ;TEST INSTRUCTION.
 033106  020027  033154                CMP     R0,#QQCTB2+2    ;IS R0 CORRECT?
 033112  001021                        BNE     QQC10           ;BRANCH IF NOT CORRECT.
 033114  023727  033142  106653        CMP     @#QQCTB1,#106653  ;IS THE RESULT CORRECT?
 033122  001025                        BNE     QQC15           ;BRANCH IF NOT CORRECT.
 033124  023727  033152  033142        CMP     @#QQCTB2,#QQCTB1  ;IS THE RESULT CORRECT?
 033132  001032                        BNE     QQC20           ;BRANCH IF NOT CORRECT.
 033134  000455                        BR      QQCDONE

                                ;TEST DATA BUFFER:
 033136  177777  177777        QQCTB0: .WORD   -1,-1
 033142  177777  177777  177777  QQCTB1: .WORD   -1,-1,-1,-1
 033150  177777
 033152  177777  177777        QQCTB2: .WORD   -1,-1

                                ;REPORT R0 INCORRECT.
 033156  010037  001242        QQC10:  MOV     R0,@#$TMP4
 033162  012737  033154  001240        MOV     #QQCTB2+2,@#$TMP3
 033170                         1$:
 033170  104377                        ERROR   +377
 033172  000015                        .WORD   15
                                                                ;R0 BAD (BUT
 033174  000435                        BR      QQCDONE         ;    FDST)X

                                ;REPORT RESULT INCORRECT.
 033176  012737  106653  001240  QQC15:  MOV     #106653,@#$TMP3                 ;    ST 634
 033204  013737  033142  001242        MOV     @#QQCTB1,@#$TMP4
 033212                         1$:
 033212  104377                        ERROR   +377
 033214  000016                        .WORD   16
```

```
                                                                ;BAD DATA
033216  000424                          BR      QQCDONE

                        ;REPORT RESULT INCORRECT.
033220  012737  033152  001240  QQC20:  MOV     #QQCTB2,@#$TMP3         ;(BUT FDST)
033226  013737  033144  001242          MOV     @#QQCTB1+2,@#$TMP4
033234                          1$:
033234  104377                          ERROR   +377
033236  000017                          .WORD   17
033240  000413                          BR      QQCDONE


                        ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
                        ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
                        ;TO THE SPURIOUS TRAP TO 4 HANDLER.
033242  011604          QQC25:  MOV     (SP),R4
033244  020427  033106          CMP     R4,#QQC2+2
033250  001402                  BEQ     1$
033252  000137  046250          JMP     @#CPSPUR

033256  011637  001236  1$:     MOV     (SP),@#$TMP2
033262  022626                  CMP     (SP)+,(SP)+
033264                          2$:
033264  104377                  ERROR   +377
033266  000020                  .WORD   20
                                                        ;(BUT FDST)+ ST634

033270                  QQCDONE:
033270  104412                  RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                        ;SEE IF THE USER HAS EXPRESSED
                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                        ;THE USER TYPED CONTROL G?).


                        ;;*****************************************************************
                        ;*TEST 65       DESTINATION MODES, MODE 5 (FL=0), TEST
                        ;*
                        ;* THIS IS A TEST OF DESTINATION MODE 5 USING
                        ;* THE STFPS INSTRUCTION
                        ;*
                        ;;*****************************************************************
033272  000004          TST65:  SCOPE


033274          RRC1:
033274  104413                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
033276  012700  033402          MOV     #RRCTB0,R0      ;SET UP THE DATA BUFFER.
033302  012701  000006          MOV     #6,R1
033306  012720  177777  1$:     MOV     #-1,(R0)+
033312  077103                  SOB     R1,1$
033314  012700  004301          MOV     #004301,R0
033320  012737  033350  001236  MOV     #RRC2,@#$TMP2
033326  012737  033506  000004  MOV     #RRC25,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
033334  170100                  LDFPS   R0              ;SET UP FPS.
```

```
5947 033336  012700  033420                    MOV     #RRCTB2+2,R0
5948 033342  012760  033406  177776            MOV     #RRCTB1,-2(R0)
5949
5950 033350  170250                    RRC2:   STFPS   @-(R0)          ;TEST INSTRUCTION.
5951 033352  020027  033416                    CMP     R0,#RRCTB2      ;IS R0 CORRECT?
5952 033356  001021                            BNE     RRC10           ;BRANCH IF NOT CORRECT.
5953 033360  023727  033406  004301            CMP     @#RRCTB1,#004301  ;IS THE RESULT CORRECT?
5954 033366  001025                            BNE     RRC15           ;BRANCH IF NOT CORRECT.
5955 033370  023727  033416  033406            CMP     @#RRCTB2,#RRCTB1  ;IS THE RESULT CORRECT?
5956 033376  001032                            BNE     RRC20           ;BRANCH IF NOT CORRECT.
5957 033400  000455                            BR      RRCDONE
5958
5959                                   ;TEST DATA BUFFER:
5960 033402  177777  177777            RRCTB0: .WORD   -1,-1
5961 033406  177777  177777  177777    RRCTB1: .WORD   -1,-1,-1,-1
     033414  177777
5962 033416  177777  177777            RRCTB2: .WORD   -1,-1
5963
5964                                   ;REPORT R0 INCORRECT.
5965 033422  010037  001242            RRC10:  MOV     R0,@#$TMP4
5966 033426  012737  033416  001240            MOV     #RRCTB2,@#$TMP3
5967 033434                            1$:
     033434  104377                            ERROR   +377
     033436  000021                            .WORD   21
5968                                                                    ;R0 BAD (BUT
5969 033440  000435                            BR      RRCDONE         ;    FDST)X
5970
5971                                   ;REPORT RESULT INCORRECT.
5972 033442  012737  004301  001240    RRC15:  MOV     #004301,@#$TMP3                 ;    ST 634
5973 033450  013737  033406  001242            MOV     @#RRCTB1,@#$TMP4
5974 033456                            1$:
     033456  104377                            ERROR   +377
     033460  000022                            .WORD   22
5975                                                                    ;BAD DATA
5976 033462  000424                            BR      RRCDONE
5977
5978
5979                                   ;REPORT RESULT INCORRECT.
5980 033464  012737  033416  001240    RRC20:  MOV     #RRCTB2,@#$TMP3                 ;BUT FDST)
5981 033472  013737  033410  001242            MOV     @#RRCTB1+2,@#$TMP4
5982 033500                            1$:
     033500  104377                            ERROR   +377
     033502  000023                            .WORD   23
5983                                                                    ;(BUT GR7,FL)
5984 033504  000413                            BR      RRCDONE         ;ST 357 TO 416
5985                                                                    ;INTO 417
5986
5987                                   ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
5988                                   ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
5989                                   ;TO THE SPURIOUS TRAP TO 4 HANDLER.
5990 033506  011604                    RRC25:  MOV     (SP),R4
5991 033510  020427  033352                    CMP     R4,#RRC2+2
5992 033514  001402                            BEQ     1$
5993 033516  000137  046250                    JMP     @#CPSPUR
5994
5995 033522  011637  001236            1$:     MOV     (SP),@#$TMP2
5996 033526  022626                            CMP     (SP)+,(SP)+
```

```
 5997 033530                          2$:
      033530  104377                        ERROR   +377
      033532  000024                        .WORD   24
 5998                                                    ;(BUT FDST)+ ST634
 5999
 6000 033534                          RRCDONE:
      033534  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                      ;SEE IF THE USER HAS EXPRESSED
                                                                      ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                      ;THE USER TYPED CONTROL G?).
 6001
 6002
 6003                                 ;;*****************************************************************
                                      ;*TEST 66       DESTINATION MODES, MODE 6 (FL=0), TEST
                                      ;*
                                      ;* THIS IS A TEST OF DESTINATION MODE 6 USING
                                      ;* THE STFPS INSTRUCTION
                                      ;*
                                      ;;*****************************************************************
      033536  000004                  TST66:  SCOPE
 6004
 6005
 6006 033540                          SSC1:
      033540  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6007 033542  012700  033652                MOV     #SSCTB0,R0      ;SET UP THE DATA BUFFER.
 6008 033546  012701  000006                MOV     #6,R1
 6009 033552  012720  177777          1$:   MOV     #-1,(R0)+
 6010 033556  077103                        SOB     R1,1$
 6011 033560  012700  102514                MOV     #102514,R0
 6012 033564  012737  033610  001236        MOV     #SSC2,@#$TMP2
 6013 033572  012737  033752  000004        MOV     #SSC25,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
 6014 033600  170100                        LDFPS   R0              ;SET UP FPS.
 6015 033602  005001                        CLR     R1
 6016 033604  012700  026455                MOV     #SSCTB1-5201,R0
 6017
 6018 033610  170260  005201          SSC2: STFPS   5201(R0)                ;TEST INSTRUCTION.
 6019 033614  020127  000000                CMP     R1,#0           ;WAS PC CORRECT AFTER EXECUTION?
 6020 033620  001070                        BNE     SSC30           ;BRANCH IF NOT CORRECT.
 6021 033622  020027  026455                CMP     R0,#SSCTB1-5201 ;IS R0 CORRECT?
 6022 033626  001017                        BNE     SSC10           ;BRANCH IF NOT CORRECT.
 6023 033630  023727  033656  102514        CMP     @#SSCTB1,#102514  ;IS THE RESULT CORRECT?
 6024 033636  001023                        BNE     SSC15           ;BRANCH IF NOT CORRECT.
 6025 033640  023727  033660  177777        CMP     @#SSCTB1+2,#-1  ;IS THE RESULT CORRECT?
 6026 033646  001030                        BNE     SSC20           ;BRANCH IF NOT CORRECT.
 6027 033650  000456                        BR      SSCDONE
 6028
 6029                                 ;TEST DATA BUFFER:
 6030 033652  177777  177777          SSCTB0: .WORD   -1,-1
 6031 033656  177777  177777  177777  SSCTB1: .WORD   -1,-1,-1,-1
      033664  177777
 6032
 6033                                 ;REPORT R0 INCORRECT.
 6034 033666  010037  001242          SSC10:  MOV     R0,@#$TMP4
 6035 033672  012737  026455  001240          MOV     #SSCTB1-5201,@#$TMP3
 6036 033700                          1$:
      033700  104377                        ERROR   +377
```

```
033702  000025                          .WORD   25
```

```
6038                                                                    ;R0 BAD
6039 033704  000440                        BR      SSCDONE
6040
6041                                ;REPORT RESULT INCORRECT.
6042 033706  012737  102534  001240 SSC15:  MOV     #102534,@#$TMP3
6043 033714  013737  033656  001242        MOV     @#SSCTB1,@#$TMP4
6044 033722                         1$:
     033722  104377                        ERROR   +377
     033724  000026                        .WORD   26
6045                                                                    ;BAD DATA
6046 033726  000427                        BR      SSCDONE
6047
6048
6049                                ;REPORT RESULT INCORRECT.
6050 033730  012737  177777  001240 SSC20:  MOV     #-1,@#$TMP3
6051 033736  013737  033660  001242        MOV     @#SSCTB1+2,@#$TMP4
6052 033744                         1$:
     033744  104377                        ERROR   +377
     033746  000027                        .WORD   27
6053                                                                    ;(BUT GR7,FL)
6054 033750  000416                        BR      SSCDONE             ;ST 357 TO 416
6055                                                                    ;INTO 417
6056
6057                                ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
6058                                ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
6059                                ;TO THE SPURIOUS TRAP TO 4 HANDLER.
6060 033752  011604                SSC25:  MOV     (SP),R4
6061 033754  020427  033612               CMP     R4,#SSC2+2
6062 033760  001402                       BEQ     1$
6063 033762  000137  046250               JMP     @#CPSPUR
6064
6065 033766  011637  001236         1$:   MOV     (SP),@#$TMP2
6066 033772  022626                       CMP     (SP)+,(SP)+
6067 033774                         2$:
     033774  104377                        ERROR   +377
     033776  000030                        .WORD   30
6068                                                                    ;(BUT FDST)+ ST634
6069 034000  000402                        BR      SSCDONE
6070
6071                                ;REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
6072 034002                         SSC30:
6073 034002                         1$:
     034002  104377                        ERROR   +377
     034004  000031                        .WORD   31
6074                                                                    ;PC NOT
6075                                                                    ;INCREMENTED
6076                                                                    ;BY 2
6077
6078 034006                         SSCDONE:
     034006  104412                        RSETUP                      ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
6079
6080
6081                                ;;*****************************************************************
```

```
                                        ;*TEST 67       DESTINATION MODES, MODE 7 (FL=0), TEST
                                        ;*
                                        ;* THIS IS A TEST OF DESTINATION MODE 7 USING
                                        ;* THE STFPS INSTRUCTION
                                        ;*
                                        ;;*****************************************************************
     034010  000004                     TST67:  SCOPE
6082
6083 034012                             TTC1:
     034012  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
6084 034014  012700  034132                    MOV     #TTCTB0,R0      ;SET UP THE DATA BUFFER.
6085 034020  012701  000010                    MOV     #10,R1
6086 034024  012720  177777             1$:    MOV     #-1,(R0)+
6087 034030  077103                            SOB     R1,1$
6088 034032  012700  103747                    MOV     #103747,R0
6089 034036  012737  034070  001236            MOV     #TTC2,@#$TMP2
6090 034044  012737  034236  000004            MOV     #TTC25,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
6091 034052  170100                            LDFPS   R0              ;SET UP FPS.
6092 034054  005001                            CLR     R1
6093 034056  012700  026745                    MOV     #TTCTB2-5201,R0
6094 034062  012760  034136  005201            MOV     #TTCTB1,5201(R0)
6095
6096 034070  170270  005201             TTC2:  STFPS   @5201(R0)               ;TEST INSTRUCTION.
6097 034074  022701  000000                    CMP     #0,R1           ;WAS PC CORRECT AFTER EXECUTION?
6098 034100  001072                            BNE     TTC30           ;BRANCH IF NOT CORRECT.
6099 034102  020027  026745                    CMP     R0,#TTCTB2-5201 ;IS R0 CORRECT?
6100 034106  001021                            BNE     TTC10           ;BRANCH IF NOT CORRECT.
6101 034110  023727  034136  103747            CMP     @#TTCTB1,#103747  ;IS THE RESULT CORRECT?
6102 034116  001025                            BNE     TTC15           ;BRANCH IF NOT CORRECT.
6103 034120  023727  034140  177777            CMP     @#TTCTB1+2,#-1  ;IS THE RESULT CORRECT?
6104 034126  001032                            BNE     TTC20           ;BRANCH IF NOT CORRECT.
6105 034130  000460                            BR      TTCDONE
6106
6107                                    ;TEST DATA BUFFER:
6108 034132  177777  177777             TTCTB0: .WORD   -1,-1
6109 034136  177777  177777  177777     TTCTB1: .WORD   -1,-1,-1,-1
     034144  177777
6110 034146  177777  177777             TTCTB2: .WORD   -1,-1
6111
6112                                    ;REPORT R0 INCORRECT.
6113 034152  010037  001242             TTC10:  MOV     R0,@#$TMP4
6114 034156  012737  026745  001240            MOV     #TTCTB2-5201,@#$TMP3
6115 034164                             1$:
     034164  104377                            ERROR   +377
     034166  000032                            .WORD   32
6116                                                                    ;R0 BAD
6117 034170  000440                            BR      TTCDONE
6118
6119
6120                                    ;REPORT RESULT INCORRECT.
6121 034172  012737  103747  001240     TTC15:  MOV     #103747,@#$TMP3
6122 034200  013737  034136  001242            MOV     @#TTCTB1,@#$TMP4
6123 034206                             1$:
     034206  104377                            ERROR   +377
     034210  000033                            .WORD   33
6124                                                                    ;BAD DATA
6125 034212  000427                            BR      TTCDONE
```

```
                                        ;REPORT RESULT INCORRECT.
034214  012737  177777  001240  TTC20:  MOV     #-1,@#$TMP3
034222  013737  034140  001242          MOV     @#TTCTB1+2,@#$TMP4
034230                          1$:
034230  104377                          ERROR   +377
034232  000034                          .WORD   34
                                                                ;(BUT GR7,FL)
034234  000416                          BR      TTCDONE         ;ST 357 TO 416
                                                                ;INTO 417

                                        ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
                                        ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
                                        ;TO THE SPURIOUS TRAP TO 4 HANDLER.
034236  011604                  TTC25:  MOV     (SP),R4
034240  020427  034072                  CMP     R4,#TTC2+2
034244  001402                          BEQ     1$
034246  000137  046250                  JMP     @#CPSPUR
034252  011637  001236          1$:     MOV     (SP),@#$TMP2
034256  022626                          CMP     (SP)+,(SP)+
034260                          2$:
034260  104377                          ERROR   +377
034262  000035                          .WORD   35
                                                                ;(BUT FSDT)+ ST634
034264  000402                          BR      TTCDONE

                                        ;REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
034266                          TTC30:
034266                          1$:
034266  104377                          ERROR   +377
034270  000036                          .WORD   36
                                                                ;PC NOT
                                                                ;INCREMENTED
034272                          TTCDONE:
034272  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).

                                ;;**************************************************************
                                ;*TEST 70        DESTINATION MODES, MODE 2 (FL=1), TEST
                                ;*
                                ;* THIS IS A TEST OF DESTINATION MODE
                                ;* 2 USING STCOL WITH REGISTER 0
                                ;*
                                ;;**************************************************************
034274  000004                  TST70:  SCOPE
034276                          UUC1:
034276  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
034300  012700  000300                  MOV     #300,R0         ;SET UP FPS.
034304  170100                          LDFPS   R0
034306  012700  034356                  MOV     #UUCTP1,R0      ;SET UP THE AC0 OPERAND.
034312  172410                          LDD     (R0),AC0
034314  012737  034326  001236          MOV     #UUC2,@#$TMP2
034322  012700  034370                  MOV     #UUCBF0,R0
```

```
 6170
 6171 034326  175420                  UUC2:   STCDL   AC0,(R0)+        ;TEST INSTRUCTION.
 6172
 6173 034330  020027  034374                  CMP     R0,#UUCBF0+4     ;IS R0 CORRECT?
 6174 034334  001420                          BEQ     UUCDONE          ;BRANCH IF CORRECT.
 6175
 6176                                 ;REPORT R0 INCORRECT.
 6177 034336  010037  001242          UUC3:   MOV     R0,@#$TMP4
 6178 034342  012737  034374  001240          MOV     #UUCBF0+4,@#$TMP3
 6179 034350                          1$:
      034350  104377                          ERROR   +377
      034352  000037                          .WORD   37
 6180                                                                  ;R0 NOT INCR BY 4
 6181 034354  000410                          BR      UUCDONE
 6182                                 ;TEST DATA BUFFER:
 6183 034356  000000  000000  000000  UUCTP1: .WORD   0,0,0,0
      034364  000000
 6184 034366  177777                          -1
 6185 034370  177777  177777  177777  UUCBF0: .WORD   -1,-1,-1
 6186
 6187 034376                          UUCDONE:
      034376  104412                          RSETUP                   ;GO INITIALIZE THE FPS AND STACK; AND
                                               ;SEE IF THE USER HAS EXPRESSED
                                               ;THE DESIRE TO CHANGE THE SOFTWARE
                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                               ;THE USER TYPED CONTROL G?).
 6188
 6195                                 ;;*****************************************************************
                                      ;*TEST 71       DESTINATION MODES, MODE 4 (FL=1), TEST
                                      ;*
                                      ;* THIS IS A TEST OF DESTINATION MODE
                                      ;* 4 USING STCDL WITH REGISTER 0
                                      ;*
                                      ;;*****************************************************************
      034400  000004                  TST71:  SCOPE
 6196
 6197 034402                          VVC1:
      034402  104413                          LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
 6198 034404  012700  000300                  MOV     #300,R0          ;SET UP FPS.
 6199 034410  170100                          LDFPS   R0
 6200 034412  012700  034462                  MOV     #VVCTP1,R0       ;SET UP THE AC0 OPERAND.
 6201 034416  172410                          LDD     (R0),AC0
 6202 034420  012737  034432  001236          MOV     #VVC2,@#$TMP2
 6203 034426  012700  034500                  MOV     #VVCBF0+4,R0
 6204
 6205 034432  175440                  VVC2:   STCDL   AC0,-(R0)                ;TEST INSTRUCTION.
 6206
 6207 034434  020027  034474                  CMP     R0,#VVCBF0       ;IS R0 CORRECT?
 6208 034440  001420                          BEQ     VVCDONE
 6209
 6210                                 ;REPORT R0 INCORRECT.
 6211 034442  010037  001242          VVC3:   MOV     R0,@#$TMP4
 6212 034446  012737  034474  001240          MOV     #VVCBF0,@#$TMP3
 6213 034454                          1$:
      034454  104377                          ERROR   +377
      034456  000040                          .WORD   40
 6214                                                                  ;R0 NOT DECR BY 4
```

```
 6215 034460  000410                          BR      VVCDONE
 6216                                 ;TEST DATA BUFFER:
 6217 034462  000000  000000  000000  VVCTP1: .WORD   0,0,0,0
      034470  000000
 6218 034472  177777                          -1
 6219 034474  177777  177777  177777  VVCBF0: .WORD   -1,-1,-1
 6220
 6221 034502                          VVCDONE:
      034502  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                      ;SEE IF THE USER HAS EXPRESSED
                                                                      ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                      ;THE USER TYPED CONTROL G?).
 6222
 6232                                 ;;*****************************************************************
                                      ;*TEST 72       STCDI AND STCDL TEST
                                      ;*
                                      ;* THIS IS A TEST OF THE STCDI AND
                                      ;* STCDL INSTRUCTIONS. NOTE THAT A
                                      ;* SUBROUTINE, STCSUB, IS USED TO
                                      ;* SET UP THE OPERANDS, EXECUTE THE STC
                                      ;* INSTRUCTION AND CHECK THE RESULT.
                                      ;*
                                      ;;*****************************************************************
      034504  000004                  TST72:  SCOPE
 6233
 6234                                 ;FIRST TEST STC WITH EXP=100 (EXCESS 200)
 6235 034506                          WWC1:
      034506  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6236 034510  004737  035654                  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6237 034514  020000  000000  000000  1$:     .WORD   20000,0,0,0     ;AC0 OPERAND.
      034522  000000
 6238 034524  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
 6239 034530  177777  177777          3$:     .WORD   -1,-1           ;ERROR RES.
 6240 034534  040300                  4$:     40300                   ;FPS BEFORE EXECUTION.
 6241 034536  040304                          40304                   ;FPS AFTER EXECUTION.
 6242 034540  140304                          140304                  ;ANTICIPATED ERRONEOUS FPS.
 6243 034542  177777                          -1              ;REPORT RESULT INCORRECT.
 6244 034544  104322                  5$:     ERROR   +322            ;RESULT INCORP.
 6245 034546  000401                          BR      6$
 6246 034550  104325                          ERROR   +325            ;EITHER (BUT FLAG)
 6247 034552                          6$:                             ;ST 662
 6248                                                                 ;OR CLEAR FLAG
 6249                                                                 ;ST 774
 6250
 6251                                 ;EXP=0 (OCT)   FL=1     FIC=0
 6252 034552                          WWC2:
      034552  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6253 034554  004737  035654                  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6254 034560  040000  000000  000000  1$:     .WORD   40000,0,0,0     ;AC     ;AC0 OPERAND.
      034566  000000
 6255 034570  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
 6256 034574  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
 6257 034600  040313                  4$:     40313                   ;FPS BEFORE EXECUTION.
 6258 034602  040304                          40304                   ;FPS AFTER EXECUTION.
 6259 034604  140304                          140304                  ;ANTICIPATED ERRONEOUS FPS.
 6260 034606  177777                          -1              ;EXPECTED FEC.
```

```
 6261 034610  104322                    5$:     ERROR   +322            ;REPORT RESULT INCORRECT.
 6262 034612  000401                            BR      6$
 6263 034614  104326                            ERROR   +326            ;REPORT FPS INCORRECT.
 6264 034616                            6$:
 6265
 6266                                   ;EXP=37 (OCT)   FL=1    FIC=1
 6267 034616                            WWC4:
      034616  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6268 034620  004737  035654                    JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6269 034624  047667  075757  157737  1$:     .WORD   47667,75757,157737,167773       ;AC0 OPERAND.
      034632  167773
 6270 034634  055675  173757            2$:     .WORD   55675,173757            ;EXPECTED RESULT.
 6271 034640  122102  004021            3$:     .WORD   122102,004021           ;ANTICIPATED ERRONEOUS RESULT.
 6272 034644  040717                    4$:     40717                   ;FPS BEFORE EXECUTION.
 6273 034646  040700                            40700                   ;FPS AFTER EXECUTION.
 6274 034650  140705                            140705                  ;ANTICIPATED ERRONEOUS FPS.
 6275 034652  177777                            -1              ;EXPECTED FEC.
 6276 034654  104327                    5$:     ERROR   +327            ;(BUT ENBT) ST 632
 6277 034656  000401                            BR      6$
 6278 034660  104326                            ERROR   +326            ;REPORT FPS INCORRECT.
 6279 034662                            6$:
 6280
 6281                                   ;EXP=40 (OCT)   FL=1    FIC=1
 6282 034662                            WWC5:
      034662  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6283 034664  004737  035654                    JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6284 034670  050000  000000  000000  1$:     .WORD   50000,0,0,0     ;AC0 OPERAND.
      034676  000000
 6285 034700  000000  000000            2$:     .WORD   0,0             ;EXPECTED RESULT.
 6286 034704  177777  177777            3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
 6287 034710  040700                    4$:     40700                   ;FPS BEFORE EXECUTION.
 6288 034712  140705                            140705                  ;FPS AFTER EXECUTION.
 6289 034714  040705                            40705                   ;ANTICIPATED ERRONEOUS FPS.
 6290 034716  000006                            6               ;EXPECTED FEC.
 6291 034720  104322                    5$:     ERROR   +322            ;REPORT RESULT INCORRECT.
 6292 034722  000401                            BR      6$
 6293 034724  104330                            ERROR   +330            ;(BUT FIC) ST 004                       ;REPORT FPS INCORRECT.
 6294                                                                    ;TO 305 INTO                    ;
 6295 034726                            6$:                             ;315
 6296
 6297                                   ;EXP=40 (OCT)    FL=1   FIC=0
 6298 034726                            WWC6:
      034726  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6299 034730  004737  035654                    JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6300 034734  050000  000000  000000  1$:     .WORD   50000,0,0,0     ;AC0 OPERAND.
      034742  000000
 6301 034744  000000  000000            2$:     .WORD   0,0             ;EXPECTED RESULT.
 6302 034750  177777  177777            3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
 6303 034754  040312                    4$:     40312                   ;FPS BEFORE EXECUTION.
 6304 034756  040305                            40305                   ;FPS AFTER EXECUTION.
 6305 034760  140305                            140305                  ;ANTICIPATED ERRONEOUS FPS.
 6306 034762  177777                            -1              ;EXPECTED FEC.
 6307 034764  104322                    5$:     ERROR   +322            ;REPORT RESULT INCORRECT.
 6308 034766  000401                            BR      6$
 6309 034770  104331                            ERROR   +331            ;(BUT FIC) ST 004 TO
 6310 034772                            6$:                             ;315 INTO 305
 6311
```

```
                                        ;EXP=30 (OCT)   FL=1    FIC=1
                                        WWC7:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                                JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
                                        1$:     .WORD   46000,1,0,0     ;AC0 OPERAND.
                                        2$:     .WORD   200,1           ;EXPECTED RESULT.
                                        3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
                                        4$:     40700                   ;FPS BEFORE EXECUTION.
                                                40700                   ;FPS AFTER EXECUTION.
                                                -1                      ;ANTICIPATED ERRONEOUS FPS.
                                                -1              ;EXPECTED FEC.
                                        5$:     ERROR   +322            ;REPORT RESULT INCORRECT.
                                                BR      6$
                                                ERROR   +323            ;REPORT FPS INCORRECT.
                                        6$:

                                        ;EXP=27 (OCT)   FL=1    FIC=1
                                        WWC8:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                                JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
                                        1$:     .WORD   45600,1,0,0     ;AC0 OPERAND.
                                        2$:     .WORD   100,0           ;EXPECTED RESULT.
                                        3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
                                        4$:     40707                   ;FPS BEFORE EXECUTION.
                                                40700                   ;FPS AFTER EXECUTION.
                                                -1                      ;ANTICIPATED ERRONEOUS FPS.
                                                -1              ;EXPECTED FEC.
                                        5$:     ERROR   +322            ;REPORT RESULT INCORRECT.
                                                BR      6$
                                                ERROR   +323            ;REPORT FPS INCORRECT.
                                        6$:

                                        ;EXP=17 (OCT)   FL=0    FIC=1
                                        WWC9:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                                JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
                                        1$:     .WORD   43600,0,0,0     ;AC0 OPERAND.
                                        2$:     .WORD   40000,-1                ;EXPECTED RESULT.
                                        3$:     .WORD   0,-1            ;ANTICIPATED ERRONEOUS RESULT.
                                        4$:     40600                   ;FPS BEFORE EXECUTION.
                                                40600                   ;FPS AFTER EXECUTION.
                                                140604                  ;ANTICIPATED ERRONEOUS FPS.
                                                -1              ;EXPECTED FEC.
                                        5$:     ERROR   +332            ;BAD CONSTANT ST 066
                                                BR      6$
                                                ERROR   +333            ;REPORT FPS INCORRECT.
                                        6$:

                                        ;EXP=20 (OCT)   FL=0    FIC=1
                                        WWC10:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                                JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
                                        1$:     .WORD   44000,0,0,0     ;AC0 OPERAND.
```

```
6361 035164  000000  177777          2$:     .WORD   0,-1                    ;EXPECTED RESULT.
6362 035170  177777  177777          3$:     .WORD   -1,-1                   ;ANTICIPATED ERRONEOUS RESULT.
6363 035174  040600                  4$:     40600                           ;FPS BEFORE EXECUTION.
6364 035176  140605                          140605                          ;FPS AFTER EXECUTION.
6365 035200  040600                          40600                           ;ANTICIPATED ERRONEOUS FPS.
6366 035202  000006                          6                       ;EXPECTED FEC.
6367 035204  104322                  5$:     ERROR   +322                    ;REPORT RESULT INCORRECT.
6368 035206  000401                          BR      6$
6369 035210  104334                          ERROR   +334                    ;BAD CONSTANT ST 066
6370 035212                          6$:
6371
6372                                 ;EXP=10 (OCT), AC NEGATIVE, FL=0, FIC=1
6373 035212                          WWC11:
     035212  104413                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
6374 035214  004737  035654                  JSR     PC,@#STCSUB             ;GO EXECUTE THE INSTRUCTION.
6375 035220  142000  000000  000000  1$:     .WORD   142000,0,0,0            ;AC0 OPERAND.
     035226  000000
6376 035230  177600  177777          2$:     .WORD   177600,-1                       ;EXPECTED RESULT.
6377 035234  000200  000000          3$:     .WORD   200,0                   ;ANTICIPATED ERRONEOUS RESULT.
6378 035240  040600                  4$:     40600                           ;FPS BEFORE EXECUTION.
6379 035242  040610                          40610                           ;FPS AFTER EXECUTION.
6380 035244  040600                          40600                           ;ANTICIPATED ERRONEOUS FPS.
6381 035246  177777                          -1                      ;EXPECTED FEC.
6382 035250  104335                  5$:     ERROR   +335            ;(BUT ENBT) ST 632
6383 035252  000401                          BR      6$
6384 035254  104336                          ERROR   +336            ;(SET FN) ST 473
6385 035256                          6$:
6386
6387                                 ;EXP=37 (OCT), FL=1, FIC=1, AC NEG.
6388 035256                          WWC12:
     035256  104413                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
6389 035260  004737  035654                  JSR     PC,@#STCSUB             ;GO EXECUTE THE INSTRUCTION.
6390 035264  147600  000000  000000  1$:     .WORD   147600,0,0,0            ;AC0 OPERAND.
     035272  000000
6391 035274  140000  000000          2$:     .WORD   140000,0                        ;EXPECTED RESULT.
6392 035300  137777  000000          3$:     .WORD   137777,0                        ;ANTICIPATED ERRONEOUS RESULT.
6393 035304  040700                  4$:     40700                           ;FPS BEFORE EXECUTION.
6394 035306  040710                          40710                           ;FPS AFTER EXECUTION.
6395 035310  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
6396 035312  177777                          -1                      ;EXPECTED FEC.
6397 035314  104337                  5$:     ERROR   +337                    ;(BUT COUT) ST 375
6398 035316  000401                          BR      6$                      ;ST 275 TO 074
6399 035320  104323                          ERROR   +323                    ;INTO 274
6400 035322                          6$:
6401
6402                                 ;EXP=37  (OCT),  FL=1,  FIC=1,  AC  NEG
6403 035322                          WWC13:
     035322  104413                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
6404 035324  004737  035654                  JSR     PC,@#STCSUB             ;GO EXECUTE THE INSTRUCTION.
6405 035330  147600  000000  001000  1$:     .WORD   147600,0,1000,0 ;AC0 OPERAND.
     035336  000000
6406 035340  137777  177777          2$:     .WORD   137777,177777           ;EXPECTED RESULT.
6407 035344  140000  177777          3$:     .WORD   140000,177777           ;ANTICIPATED ERRONEOUS RESULT.
6408 035350  040707                  4$:     40707                           ;FPS BEFORE EXECUTION.
6409 035352  040710                          40710                           ;FPS AFTER EXECUTION.
6410 035354  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
6411 035356  177777                          -1                      ;EXPECTED FEC.
```

```
 6412 035360  104340                5$:     ERROR   +340            ;(BUT COUT) ST 375
 6413 035362  000401                        BR      6$              ;TO 274 INTO 074
 6414 035364  104323                        ERROR   +323            ;REPORT FPS INCORRECT.
 6415 035366                        6$:
 6416
 6417                               ;EXP=41 (OCT),  AC NEG, FL=1, FIC=1
 6418 035366                        WWC14:
      035366  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6419 035370  004737  035654                JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6420 035374  150200  000000  000000  1$:   .WORD   150200,0,0,0     ;AC0 OPERAND.
      035402  000000
 6421 035404  000000  000000        2$:   .WORD   0,0               ;EXPECTED RESULT.
 6422 035410  177777  177777        3$:   .WORD   -1,-1             ;ANTICIPATED ERRONEOUS RESULT.
 6423 035414  040700                4$:   40700                     ;FPS BEFORE EXECUTION.
 6424 035416  140705                      140705                    ;FPS AFTER EXECUTION.
 6425 035420  177777                      -1                        ;ANTICIPATED ERRONEOUS FPS.
 6426 035422  000006                      6             ;EXPECTED FEC.
 6427 035424  104322                5$:   ERROR   +322              ;REPORT RESULT INCORRECT.
 6428 035426  000401                        BR      6$
 6429 035430  104341                        ERROR   +341            ;(BUT EZBT) ST 377
 6430 035432                        6$:
 6431                               ;EXP=40 (OCT),  AC NEG,  FL=1,  FIC=1
 6432 035432                        WWC15:
      035432  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6433 035434  004737  035654                JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6434 035440  150000  000001  000000  1$:   .WORD   150000,1,0,0     ;AC0 OPERAND.
      035446  000000
 6435 035450  000000  000000        2$:   .WORD   0,0               ;EXPECTED RESULT.
 6436 035454  100000  177600        3$:   .WORD   100000,-200               ;ANTICIPATED ERRONEOUS RESULT.
 6437 035460  040700                4$:   40700                     ;FPS BEFORE EXECUTION.
 6438 035462  140705                      140705                    ;FPS AFTER EXECUTION.
 6439 035464  040700                      40700                     ;ANTICIPATED ERRONEOUS FPS.
 6440 035466  000006                      6             ;EXPECTED FEC.
 6441 035470  104342                5$:   ERROR   +342              ;(BUT COUT) ST 360
 6442 035472  000401                        BR      6$              ;TO 654 INTO 454
 6443 035474  104323                        ERROR   +323            ;REPORT FPS INCORRECT.
 6444 035476                        6$:
 6445
 6446                               ;EXP=40, AC NEGATIVE, FL=1, FIC=1
 6447 035476                        WWC16:
      035476  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6448 035500  004737  035654                JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6449 035504  150001  000000  000000  1$:   .WORD   150001,0,0,0     ;AC0 OPERAND.
      035512  000000
 6450 035514  000000  000000        2$:   .WORD   0,0               ;EXPECTED RESULT.
 6451 035520  077400  000000        3$:   .WORD   77400,0           ;ANTICIPATED ERRONEOUS RESULT.
 6452 035524  040700                4$:   40700                     ;FPS BEFORE EXECUTION.
 6453 035526  140705                      140705                    ;FPS AFTER EXECUTION.
 6454 035530  177777                      -1                        ;ANTICIPATED ERRONEOUS FPS.
 6455 035532  000006                      6             ;EXPECTED FEC.
 6456 035534  104343                5$:   ERROR   +343              ;REPORT RESULT INCORRECT.
 6457 035536  000401                        BR      6$
 6458 035540  104323                        ERROR   +323            ;REPORT FPS INCORRECT.
 6459 035542                        6$:
 6460
 6461
 6462                               ;EXP 40    (OCT), AC MOST NEG LONG INT, FL=1
```

```
                                        ;FIC=1
035542                                  WWC17:
035542  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
035544  004737  035654                          JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
035550  150000  000000  000000  1$:     .WORD   150000,0,0,0    ;AC0 OPERAND.
035556  000000
035560  100000  000000          2$:     .WORD   100000,0                ;EXPECTED RESULT.
035564  000000  000000          3$:     .WORD   0,0             ;ANTICIPATED ERRONEOUS RESULT.
035570  040700                  4$:     40700                   ;FPS BEFORE EXECUTION.
035572  040710                          40710                   ;FPS AFTER EXECUTION.
035574  140705                          140705                  ;ANTICIPATED ERRONEOUS FPS.
035576  177777                          -1              ;EXPECTED FEC.
035600  104344                  5$:     ERROR   +344                    ;(BUT NBIT) ST 654
035602  000401                          BR      6$                      ;OR (BUT COUT) ST 454
035604  104323                          ERROR   +323            ;REPORT FPS INCORRECT.
035606                          6$:
                                        ;EXP=20, AC = MOST NEG INTEGER, FL=0, FIC=1
035606                                  WWC18:
035606  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
035610  004737  035654                          JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
035614  144000  000001  000000  1$:     .WORD   144000,1,0,0    ;AC0 OPERAND.
035622  000000
035624  100000  177777          2$:     .WORD   100000,-1               ;EXPECTED RESULT.
035630  100000  177400          3$:     .WORD   100000,177400           ;ANTICIPATED ERRONEOUS RESULT.
035634  040600                  4$:     40600                   ;FPS BEFORE EXECUTION.
035636  040610                          40610                   ;FPS AFTER EXECUTION.
035640  140605                          140605                  ;ANTICIPATED ERRONEOUS FPS.
035642  177777                          -1              ;EXPECTED FEC.
035644  104345                  5$:     ERROR   +345                    ;(BUT FL) ST 633
035646  000401                          BR      6$                      ;TO 655 INTO 654
035650  104323                          ERROR   +323            ;REPORT FPS INCORRECT.

035652  000534                  6$:     BR      WWCDONE

                                ;THIS SUBROUTINE, STCSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
                                ;THE STCDI OR STCDL INSTRUCTION AND CHECK THE RESULTS. A CALL
                                ;TO IT IS MADE THUS:
                                ;
                                ;               JSR     PC,@#STCSUB
                                ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                ;               RES:    .WORD   X,X             ;EXPECTED RESULT
                                ;               ERRES:  .WORD   X,X             ;ERROR RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               ERFPS:  .WORD   X               ;ERROR FPS.
                                ;               FEC:    .WORD   X               ;EXPECTED FEC
                                ;               ERR1:   ERROR   +X              ;DATA ERROR.
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   +X              ;FPS ERROR.
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                ;THE STCDI OR STCDL INSTRUCTION IS EXECUTED.
                                ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCSUB RETURNS CONTROL
```

```
                                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCSUB
                                        ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCSUB WILL RETURN
                                        ;TO THE ERROR CALL AT ERR2, OTHERWISE STCSUB ITSELF
                                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                        ;STCDI OR STCDL IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                        ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                        ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCSUB
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCSUB WILL
                                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

035654  012601                  STCSUB: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
035656  012700  000200                  MOV     #200,R0         ;SET UP THE AC0 OPERAND.
035662  170100                          LDFPS   R0
035664  010100                          MOV     R1,R0
035666  172410                          LDD     (R0),AC0
035670  012702  036134                  MOV     #STCIBF,R2      ;INITIALIZE THE OUT PUT BUFFER.
035674  012700  000004                  MOV     #4,R0
035700  012722  177777          1$:     MOV     #-1,(R2)+
035704  077003                          SOB     R0,1$
035706  016100  000020                  MOV     20(R1),R0       ;SET THE FPS.
035712  170100                          LDFPS   R0
035714  012737  035726  001236          MOV     #2$,@#$TMP2
035722  012700  036134                  MOV     #STCIBF,R0
035726  175410                  2$:     STCDL   AC0,(R0)        ;TEST INSTRUCTION.

035730  170204                          STFPS   R4              ;GET THE FPS.
035732  170305                          STST    R5              ;GET THE FEC.
035734  010102                          MOV     R1,R2
035736  010237  001240                  MOV     R2,@#$TMP3
035742  062702  000010                  ADD     #10,R2
035746  010237  001244                  MOV     R2,@#$TMP5
035752  012737  036134  001242          MOV     #STCIBF,@#$TMP4
035760  010437  001250                  MOV     R4,@#$TMP7
035764  016137  000022  001252          MOV     22(R1),@#$TMP10
035772  010102                          MOV     R1,R2
035774  062702  000010                  ADD     #10,R2
036000  012700  036134                  MOV     #STCIBF,R0      ;SEE IF THE RESULT IS CORRECT.
036004  012703  000002                  MOV     #2,R3
036010  022022                  3$:     CMP     (R0)+,(R2)+
036012  001014                          BNE     15$
036014  077303                          SOB     R3,3$
036016  016102  000022                  MOV     22(R1),R2
036022  020204                          CMP     R2,R4           ;SEE IF THE FPS IS CORRECT.
036024  001025                          BNE     20$             ;BRANCH IF INCORRECT.
036026  005702                          TST     R2
036030  100003                          BPL     4$
036032  026105  000026                  CMP     26(R1),R5       ;SEE IF THE FEC IS CORRECT.
036036  001027                          BNE     25$             ;BRANCH IF INCORRECT.

036040  000161  000036          4$:     JMP     36(R1)          ;RETURN.
                                ;DATA ERROR:
                                ;SEE IF THE FAILURE WAS ANTICIPATED.
036044  010102                  15$:    MOV     R1,R2
036046  062702  000014                  ADD #14,R2
036052  012700  036134                  MOV     #STCIBF,R0
036056  012703  000002                  MOV     #2,R3
```

```
 6573 036062  022022                    16$:    CMP     (R0)+,(R2)+
 6574 036064  001003                            BNE     17$
 6575 036066  077303                            SOB     R3,16$
 6576 036070  000161  000030                    JMP     30(R1)
 6577 036074                            17$:
 6578                                   ;FAILURE WAS NOT ANTICIPATED SO REPORT INCORRECT RESULT HERE.
 6579 036074  104322                    18$:    ERROR   +322                    ;DATA BAD
 6580 036076  000760                            BR      4$
 6581
 6582                                   ;FPS INCORRECT, SO SEE IF FAILURE WAS ANTICIPATED.
 6583 036100  020461  000024            20$:    CMP     R4,24(R1)
 6584 036104  001002                            BNE     21$
 6585 036106  000161  000034                    JMP     34(R1)
 6586 036112                            21$:
 6587                                   ;NOT ANTICIPATED SO REPORT BAD FPS HERE.
 6588 036112  104323                    22$:    ERROR   +323                    ;FPS BAD
 6589 036114  000751                            BR      4$
 6590
 6591                                   ;REPORT INCORRECT FEC.
 6592 036116  016137  000026  001256    25$:    MOV     26(R1),@#$TMP12
 6593 036124  010537  001254                    MOV     R5,@#$TMP11
 6594 036130  104324                    26$:    ERROR   +324
 6595 036132  000742                            BR      4$
 6596
 6597                                   ;DATA BUFFER:
 6598 036134  177777  177777  177777    STCIBF: .WORD   -1,-1,-1,-1
      036142  177777
 6599
 6600 036144                            WWCDONE:
      036144  104412                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                          ;SEE IF THE USER HAS EXPRESSED
                                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                          ;THE USER TYPED CONTROL G?).
 6601
 6602
 6610                                   ;;*********************************************************************
                                        ;*TEST 73       STCFL AND STCFI TEST
                                        ;*
                                        ;* THIS IS A TEST OF STCFL AND STCFI.  IT
                                        ;* MAKES USE OF THE SAME SUBROUTINE, STCSUB,
                                        ;* WHICH WAS USED TO TEST STCDL AND STCDI.
                                        ;*
                                        ;;*********************************************************************
      036146  000004                    TST73:  SCOPE
 6611
 6612
 6613                                   ;EXPONENT=37, FL=1
 6614 036150                            XXC1:
      036150  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6615 036152  004737  035654                    JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
 6616 036156  047777  177777  177777  1$:     .WORD   47777,-1,-1,-1  ;AC0 OPERAND.
      036164  177777
 6617 036166  077777  177600            2$:     .WORD   77777,177600            ;EXPECTED RESULT.
 6618 036172  077777  177777            3$:     .WORD   77777,177777            ;ANTICIPATED ERRONEOUS RESULT.
 6619 036176  040100                    4$:     40100                   ;FPS BEFORE EXECUTION.
 6620 036200  040100                            40100                   ;FPS AFTER EXECUTION.
```

```
                                        -1                      ;ANTICIPATED ERRONEOUS FPS.
                                        -1                  ;EXPECTED FEC.
                                5$:     ERROR   +346                        ;X11(1,0)+0 ST 773X
                                        BR      6$
                                        ERROR   +323            ;REPORT FPS INCORRECT.
                                6$:

                                XXCDONE:
036214  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).


                                ;;*****************************************************************
                                ;*TEST 74       STEXP TEST
                                ;*
                                ;* THIS IS A TEST OF THE STEXP
                                ;* INSTRUCTION
                                ;*
                                ;;*****************************************************************
036216  000004                  TST74:  SCOPE

                                ; EXP = 100 (EXCESS 200)
                                YYC1:
036220  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
036222  004737  036506                   JSR     PC,@#STXSUB
036226  020000  000000  000000  1$:     .WORD   20000,0,0,0                     ;AC
036234  000000
036236  177700                  2$:     -100                                    ;EXP RES
036240  052525                  3$:     52525                                   ;ERROR EXP.
036242  040000                  4$:     40000                                   ;FPSB
036244  040010                          40010                                   ;FPSA
036246  040000                          40000                                   ;ERROR FPS
036250  104347                  5$:     ERROR   +347                            ;BAD EXP
036252  000401                          BR      6$
036254  104352                          ERROR   +352                    ;+(BUT ENBT) ST 376
036256                          6$:

                                ; EXP = 200 (EXCESS 200)
036256                          YYC2:
036256  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
036260  004737  036506                   JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
036264  040000  000000  000000  1$:     .WORD   40000,0,0,0     ;AC0 OPERAND.
036272  000000
036274  000000                  2$:     0                       ;EXPECTED EXPONENT RESULT.
036276  052525                  3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
036300  040000                  4$:     40000                   ;FPS BEFORE EXECUTION.
036302  040004                          40004                   ;FPS AFTER EXECUTION.
036304  040000                          40000                   ;ANTICIPATED ERRONEOUS FPS.
036306  104347                  5$:     ERROR   +347            ;REPORT RESULT INCORRECT.
036310  000401                          BR      6$
036312  104353                          ERROR   +353                    ;(BUT EZBT) ST 071
                                                                        ;TO 072 INT 272
036314                          6$:
```

```
                                        ; EXP = 201  (EXCESS 200)
                                        ;
036314                                  YYC3:
036314  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
036316  004737  036506                        JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
036322  040200  000000  000000        1$:     .WORD   40200,0,0,0     ;AC0 OPERAND.
036330  000000
036332  000001                        2$:     1                       ;EXPECTED EXPONENT RESULT.
036334  052525                        3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
036336  040000                        4$:     40000                   ;FPS BEFORE EXECUTION.
036340  040000                                40000                   ;FPS AFTER EXECUTION.
036342  040004                                40004                   ;ANTICIPATED ERRONEOUS FPS.
036344  104347                        5$:     ERROR   +347            ;REPORT RESULT INCORRECT.
036346  000401                                BR      6$
036350  104354                                ERROR   +354                    ;(BUT EZBT) ST 071
036352                                6$:                             ;TO 272 INTO 072
                                        ;
                                        ; EXP = 375  (EXCESS 200)
                                        ;
036352                                  YYC4:
036352  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
036354  004737  036506                        JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
036360  077200  000000  000000        1$:     .WORD   77200,0,0,0     ;AC0 OPERAND.
036366  000000
036370  000175                        2$:     175                     ;EXPECTED EXPONENT RESULT.
036372  052525                        3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
036374  040000                        4$:     40000                   ;FPS BEFORE EXECUTION.
036376  040000                                40000                   ;FPS AFTER EXECUTION.
036400  040010                                40010                   ;ANTICIPATED ERRONEOUS FPS.
```

```
 6694 036402  104347                  5$:     ERROR   +347            ;REPORT RESULT INCORRECT.
 6695 036404  000401                          BR      6$
 6696 036406  104355                          ERROR   +355                    ;(BUT ENBT) ST 376
 6697 036410                          6$:                                     ;TO 471 INTO 071
 6698
 6699                                 ; EXP = 1  (EXCESS 200)
 6700
 6701 036410                          YYC5:
      036410  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6702 036412  004737  036506                  JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
 6703 036416  000200  000000  000000  1$:     .WORD   200,0,0,0       ;AC0 OPERAND.
      036424  000000
 6704 036426  177601                  2$:     -177                    ;EXPECTED EXPONENT RESULT.
 6705 036430  052525                  3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
 6706 036432  040000                  4$:     40000                   ;FPS BEFORE EXECUTION.
 6707 036434  040010                          40010                   ;FPS AFTER EXECUTION.
 6708 036436  040000                          40000                   ;ANTICIPATED ERRONEOUS FPS.
 6709 036440  104347                  5$:     ERROR   +347            ;REPORT RESULT INCORRECT.
 6710 036442  000401                          BR      6$
 6711 036444  104352                          ERROR   +352            ;REPORT FPS INCORRECT.
 6712 036446                          6$:
 6713
 6714                                 ; EXP = 156  (EXCESS 200)
 6715
 6716 036446                          YYC6:
      036446  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6717 036450  004737  036506                  JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
 6718 036454  033400  000000  000000  1$:     .WORD   33400,0,0,0     ;AC0 OPERAND.
      036462  000000
 6719 036464  177756                  2$:     -22                     ;EXPECTED EXPONENT RESULT.
 6720 036466  052525                  3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
 6721 036470  047707                  4$:     47707                   ;FPS BEFORE EXECUTION.
 6722 036472  047710                          47710                   ;FPS AFTER EXECUTION.
 6723 036474  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
 6724 036476  104347                  5$:     ERROR   +347            ;REPORT RESULT INCORRECT.
 6725 036500  000401                          BR      6$
 6726 036502  104350                          ERROR   +350            ;REPORT FPS INCORRECT.
 6727
 6728 036504  000510                  6$:     BR      YYCDONE
 6729
 6730                                 ;THIS SUBROUTINE, STXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
 6731                                 ;THE STEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
 6732                                 ;TO IT IS MADE THUS:
 6733                                 ;
 6734                                 ;               JSR     PC,@#STXSUB
 6735                                 ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
 6736                                 ;               RES:    .WORD   X               ;EXPECTED RESULT
 6737                                 ;               ERRES:  .WORD   X               ;ERROR RESULT
 6738                                 ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
 6739                                 ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
 6740                                 ;               ERFPS:  .WORD   X               ;ERROR FPS.
 6741                                 ;               ERR1:   ERROR   +X              ;DATA ERROR.
 6742                                 ;                       BR      CONT
 6743                                 ;               ERR2:   ERROR   +X              ;FPS ERROR.
 6744                                 ;               CONT:                           ;RETURN ADDRESS
 6745                                 ;
 6746                                 ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
```

```
                                ;THE STEXP INSTRUCTION IS EXECUTED.
                                ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STXSUB RETURNS CONTROL
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STXSUB
                                ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STXSUB WILL RETURN
                                ;TO THE ERROR CALL AT ERR2, OTHERWISE STXSUB ITSELF
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                ;STEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STXSUB
                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STXSUB WILL
                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

036506  012601                  STXSUB: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
036510  010102                          MOV     R1,R2
036512  010237  001240                  MOV     R2,@#$TMP3
036516  062702  000010                  ADD     #10,R2
036522  012237  001244                  MOV     (R2)+,@#$TMP5
036526  012737  036574  001236          MOV     #1$,@#$TMP2
036534  012737  123456  036714          MOV     #123456,@#STXBF
036542  012737  076543  036716          MOV     #76543,@#STXBF+2
036550  012700  000200                  MOV     #200,R0
036554  170100                          LDFPS   R0
036556  010100                          MOV     R1,R0           ;SET UP THE AC0 OPERAND.
036560  172410                          LDD     (R0),AC0
036562  016100  000016                  MOV     16(R1),R0       ;SET THE FPS.
036566  170100                          LDFPS   R0
036570  012700  036714                  MOV     #STXBF,R0
036574  175010                  1$:     STEXP   AC0,(R0)        ;TEST INSTRUCTION.
036576  170204                          STFPS   R4              ;GET FPS.
036600  010437  001250                  MOV     R4,@#$TMP7
036604  016137  000016  001252          MOV     16(R1),@#$TMP10
036612  013737  036714  001242          MOV     @#STXBF,@#$TMP4
036620  026137  000010  036714          CMP     10(R1),@#STXBF  ;WAS RESULT CORRECT?
036626  001411                          BEQ     5$              ;BRANCH IF CORRECT.
036630  026137  000012  036714          CMP     12(R1),@#STXBF  ;OTHERWISE SEE IF THE FAILURE WAS ANTICIPATED.
036636  001002                          BNE     2$
036640  000161  000022                  JMP     22(R1)

                                ;IF NOT ANTICIPATED REPORT ERROR HERE.
036644                          2$:
036644  104347                  3$:     ERROR   +347                            ;EXP BAD
036646  000161  000030          4$:     JMP     30(R1)

036652  020461  000016          5$:     CMP     R4,16(R1)       ;SEE IF THE FPS IS CORRECT.
036656  001407                          BEQ     10$             ;BRANCH IF CORRECT.
036660  020461  000020                  CMP     R4,20(R1)       ;SEE IF THE FAILURE WAS ANTICIPATED.
036664  001002                          BNE     6$
036666  000161  000026                  JMP     26(R1)

                                ;FPS ERROR WAS NOT ANTICIPATED SO REPORT ERROR HERE.
036672                          6$:
036672  104350                  7$:     ERROR   +350                            ;FPS BAD
036674  000764                          BR      4$

                                ;SEE IF MORE THAN ONE WORD WAS WRITTEN IN THE OUTPUT BUFFER.
```

```
 6804 036676  022737  076543  036716  10$:    CMP     #76543,@#STXBF+2
 6805 036704  001760                   BEQ     4$
 6806 036706  104351                   11$:    ERROR   +351                    ;FDFL+0 ST 347X
 6807 036710  000756                   BR      4$
 6808
 6809 036712  177777                   -1
 6810 036714  177777  177777  177777  STXBF:  .WORD   -1,-1,-1,-1,-1
      036722  177777  177777
 6811
 6812 036726                           YYCDONE:
      036726  104412                           RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                      ;SEE IF THE USER HAS EXPRESSED
                                                                      ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                      ;THE USER TYPED CONTROL G?).
 6813
 6824                                 ;;*****************************************************************
                                      ;*TEST 75       STST TEST
                                      ;*
                                      ;* THIS IS A TEST OF THE STST
                                      ;* INSTRUCTION.  FIRST AN ILLEGAL FPS OP CODE
                                      ;* (INSTRUCTION) IS USED TO ENTER AN
                                      ;* ERROR CONDITION IN THE FEC AND
                                      ;* FEA.  THE STST IS EXECUTED AND
                                      ;* THE FEC AND FEA ARE CHECKED
                                      ;*
                                      ;;*****************************************************************
      036730  000004                  TST75:  SCOPE
 6825
 6826 036732                          ZZC1:
      036732  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6827 036734  012700  040000                  MOV     #40000,R0       ;SET FPS. FID=1.
 6828 036740  170100                          LDFPS   R0
 6829
 6830 036742  170003                  ZZC2:   .WORD   170003                  ;ILLEGAL FPP
 6831                                                                         ;OP CODE
 6832 036744  012700  037120                  MOV     #ZZCBF,R0       ;SET UP THE OUTPUT BUFFER.
 6833 036750  012710  177777                  MOV     #-1,(R0)
 6834 036754  012760  177777  000002         MOV     #-1,2(R0)
 6835 036762  012737  036770  001236         MOV     #ZZC3,@#$TMP2
 6836 036770  170310                  ZZC3:   STST    (R0)                    ;GET FEC AND
 6837                                                                         ;FEA
 6838 036772  170204                          STFPS   R4              ;GET FPS.
 6839 036774  012700  037120                  MOV     #ZZCBF,R0
 6840 037000  011037  001240                  MOV     (R0),@#$TMP3
 6841 037004  016037  000002  001242         MOV     2(R0),@#$TMP4
 6842 037012  012737  000002  001244         MOV     #2,@#$TMP5
 6843 037020  012737  036742  001246         MOV     #ZZC2,@#$TMP6
 6844 037026  010437  001250                  MOV     R4,@#$TMP7
 6845 037032  012737  140000  001252         MOV     #140000,@#$TMP10
 6846
 6847 037040  022710  000002                  CMP     #2,(R0) ;SEE IF FEC IS CORRECT.
 6848 037044  001010                          BNE     ZZC5            ;BRANCH IF INCORRECT.
 6849 037046  022760  036742  000002         CMP     #ZZC2,2(R0)     ;SEE IF FEA, ADDRESS, IS CORRECT.
 6850 037054  001006                          BNE     ZZC10           ;BRANCH IF INCORRECT.
 6851 037056  022704  140000                  CMP     #140000,R4      ;SEE IF FPS IS CORRECT.
 6852 037062  001013                          BNE     ZZC15           ;BRANCH IF INCORRECT.
```

```
 6853 037064  000422                          BR      ZZCDONE
 6854
 6855                                 ;REPORT FEC INCORRECT
 6856 037066                          ZZC5:
 6857 037066  104356                  1$:     ERROR   +356                    ;STST BAD
 6858 037070  000420                          BR      ZZCDONE          ;FECX
 6859
 6860                                 ;REPORT FEA INCORRECT
 6861 037072  022760  177777  000002 ZZC10:  CMP     #-1,2(R0)
 6862 037100  001402                          BEQ     ZZC12
 6863 037102  104357                  1$:     ERROR   +357                    ;STST BAD FEA
 6864 037104  000412                          BR      ZZCDONE
 6865 037106                          ZZC12:
 6866 037106  104360                  1$:     ERROR   +360                    ;SET FD FL ST 636
 6867 037110  000410                          BR      ZZCDONE
 6868
 6869                                 ;REPORT FPS INCORRECT
 6870 037112                          ZZC15:
 6871 037112  104361                  1$:     ERROR   +361                    ;FPS X AFTER ST ST
 6872 037114  000406                          BR      ZZCDONE
 6873
 6874                                 ;DATA BUFFER:
 6875 037116  177777                          -1
 6876 037120  177777  177777  177777 ZZCBF:  .WORD   -1,-1,-1,-1
      037126  177777
 6877 037130  177777                          -1
 6878
 6879 037132                          ZZCDONE:
      037132  104412                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                               ;SEE IF THE USER HAS EXPRESSED
                                               ;THE DESIRE TO CHANGE THE SOFTWARE
                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                               ;THE USER TYPED CONTROL G?).
```

```
 6888                                   ;;*****************************************************************
                                        ;*TEST 76       ENABLE D-SPACE AND SEE I-SPACE IS FORCED
                                        ;*
                                        ;*THIS IS A TEST THAT WILL ENABLE D-SPACE BUT MAKE IT NON-RESIDENT
                                        ;*SO THAT AN INSTRUCTION THAT ACCESSES D-SPACE WHEN IT NORMALLY
                                        ;*SHOULDN'T WILL CAUSE A TRAP/ABORT.
                                        ;*
                                        ;;*****************************************************************
     037134  000004                     TST76:  SCOPE
 6889 037136                            ZZF1:
     037136  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6890 037140  005067  140426                    CLR     MMR0            ;MAKE SURE MEMORY MANAGEMENT IS OFF.
 6891 037144  170127  040000                    LDFPS   #40000          ;LOAD FPS STATUS.
 6892
 6893 037150  012767  077406  133142           MOV     #77406,KDPDR0   ;MAKE KDPDR0 RESIDENT.
 6894 037156  012767  077400  133136           MOV     #77400,KDPDR1   ;MAKE KDPDR1 NON-RESIDENT.
 6895 037164  012767  077400  133132           MOV     #77400,KDPDR2   ;MAKE KDPDR2 NON-RESIDENT.
 6896 037172  012767  077000  133126           MOV     #77000,KDPDR3   ;MAKE KDPDR3 NON-RESIDENT FOR ADDRESSES 60000-77756.
 6897 037200  012767  077406  133122           MOV     #77406,KDPDR4   ;MAKE KDPDR4 RESIDENT FOR ADDRESSES 77760-77776.
 6898 037206  012767  077406  133122           MOV     #77406,KDPDR7   ;MAKE KDPDR7 RESIDENT (I/O PAGE).
 6899
 6900 037214  005067  133140                    CLR     KDPAR0          ;MAP D-PAGE 0 FOR 0-4K.
 6901 037220  012767  000200  133134           MOV     #200,KDPAR1     ;MAP D-PAGE 1 FOR 4-8K.
 6902 037226  012767  000400  133130           MOV     #400,KDPAR2     ;MAP D-PAGE 2 FOR 8-12K.
 6903 037234  012767  000600  133124           MOV     #600,KDPAR3     ;MAP D-PAGE 3 FOR ACCESSING ADDRESSES 60000-77756.
 6904 037242  012767  000600  133120           MOV     #600,KDPAR4     ;MAP D-PAGE 4 FOR ACCESSING ADDRESSES 77760-77776.
 6905 037250  012767  177600  133120           MOV     #177600,KDPAR7  ;MAP D-PAGE 7 FOR I/O PAGE.
 6906
 6907 037256  012767  077406  133014           MOV     #77406,KIPDR0   ;MAKE KIPDR0 RESIDENT.
 6908 037264  012767  077406  133010           MOV     #77406,KIPDR1   ;MAKE KIPDR1 RESIDENT.
 6909 037272  012767  077406  133004           MOV     #77406,KIPDR2   ;MAKE KIPDR2 RESIDENT.
 6910 037300  012767  077006  133000           MOV     #77006,KIPDR3   ;MAKE KIPDR3 RESIDENT FOR USING ADDRESSES 60000-77756.
 6911 037306  012767  077400  132774           MOV     #77400,KIPDR4   ;MAKE KIPDR4 NON-RESIDENT FOR USING ADDRESSES 77760-77776.
 6912 037314  012767  077406  132774           MOV     #77406,KIPDR7   ;MAKE KIPDR7 RESIDENT (I/O PAGE).
 6913
 6914 037322  005067  133012                    CLR     KIPAR0          ;MAP I-PAGE 0 FOR 0-4K.
 6915 037326  012767  000200  133006           MOV     #200,KIPAR1     ;MAP I-PAGE 1 FOR 4-8K.
 6916 037334  012767  000400  133002           MOV     #400,KIPAR2     ;MAP I-PAGE 2 FOR 8-12K.
 6917 037342  012767  000600  132776           MOV     #600,KIPAR3     ;MAP I-PAGE 3 FOR ACCESSING ADDRESSES 60000-77756.
 6918 037350  012767  000600  132772           MOV     #600,KIPAR4     ;MAP I-PAGE 4 FOR ACCESSING ADDRESSES 77760-77776.
 6919 037356  012767  177600  132772           MOV     #177600,KIPAR7  ;MAP I-PAGE 7 FOR I/O PAGE.
 6920
 6921 037364  016767  140660  141670           MOV     MMVECT,$TMP14   ;MOVE MM TRAP VECTOR TO $TMP14 FOR TEMP STORAGE.
 6922 037372  012701  117760                    MOV     #DATA,R1        ;SET UP R1.
 6923 037376  012702  117770                    MOV     #DATA+10,R2     ;SET UP R2.
 6924 037402  012703  117772                    MOV     #DATA+12,R3     ;SET UP R3.
 6925 037406  012737  037716  000250           MOV     #TRAPV,@#MMVECT ;SET UP FOR FP TRAPS FOR THIS TEST.
 6926 037414  012737  000340  000252           MOV     #340,@#MMVECT+2
 6927 037422  012737  000024  172516           MOV     #24,@#MMR3      ;TURN ON 22-BIT KERNEL D-SPACE.
 6928 037430  005237  177572                    INC     @#MMR0          ;TURN ON MEMORY MANAGEMENT.
 6929 037434  012737  117760  117770           MOV     #DATA,@#DATA+10 ;SET UP ADDRESS POINTER.
 6930
 6931 037442  170000                           CFCC                    ;* TEST INSTRUCTION WHICH SHOULD NEVER INVOKE D-SPACE.
 6932                                   ;* THIS INSTRUCTION WILL TEST FOR A WORST-CASE HARDWARE PROBLEM.
 6933
 6934                                   ;*****ALL REFERENCES TO MICRO-FLOWS REFER TO *FP11-F-2 REV A* FLOWS*****
 6935                                   ;THE COMMENTS FOR EACH TEST LINE ARE WRITTEN SO YOU CAN GO TO THE MICRO FLOWS
```

```
                                        ;AND PINPOINT THE PROBLEM AREA.  FROM THERE, HARDWARE ANALYSIS SHOULD BE EASIER.
                                        ;* INSTRUCTION GROUPS ISOLATED BY BLANK LINES ARE TO BE EXECUTED TOGETHER
                                        ;* DUE TO PROPER SETUP PURPOSES.  I.E. THE LOCATION OR ADDRESS HAS TO BE
                                        ;* INITIALIZED PROPERLY BEFORE THE INSTRUCTION CAN BE ACCOMPLISHED.

                                        ;* TESTING MODE 1 REG 0.

037444  010100                          MOV     R1,R0           ;SETTING UP R0.
037446  170410                          CLRF    (R0)            ;TESTING BLOCKS 27-K AND 27-R.
037450  177010                          LDCIF   (R0),AC0        ;TESTING BLOCKS 28-F AND 28-P.
037452  172410                          LDF     (R0),AC0        ;TESTING BLOCKS 4-J, 4-X, 4-Z AND 4-BB.
037454  170310                          STST    (R0)            ;TESTING BLOCKS 33-E AND 33-P.

                                        ;* TESTING MODE 2 REG 0 AND 7.

037456  010100                          MOV     R1,R0           ;SETTING UP R0.
037460  170520                          TSTF    (R0)+           ;TESTING BLOCK 21-AA.
037462  170527  001000                  TSTF    #1000

037466  010100                          MOV     R1,R0           ;CORRECTING R0.
037470  170420                          CLRF    (R0)+           ;TESTING BLOCKS 27-K AND 27-R.
                                        ;**NOTE**: THE LOCATION AFTER THE CLRF, AND STST MODE 2 REG 7 INSTRUCTIONS
                                        ;*WILL* BE CHANGED ON SUBSEQUENT PASSES, BUT IS **NOT** INCORRECT.  THE
                                        ;ACTUAL CONTENTS OF THOSE LOCATIONS IS IMMATERIAL, AS THIS TEST INSURES
                                        ;THAT THE INSTRUCTION DOES EXECUTE WITHOUT ACCESSING THAT LOCATION AS
                                        ;A D-SPACE ACCESS.
037472  170427  001000                  CLRF    #1000

037476  010100                          MOV     R1,R0           ;CORRECTING R0.
037500  177020                          LDCIF   (R0)+,AC0       ;TESTING BLOCKS 28-F AND 28-P.
037502  177027  001000                  LDCIF   #1000,AC0

037506  010100                          MOV     R1,R0           ;CORRECTING R0.B
037510  172420                          LDF     (R0)+,AC0       ;TESTING BLOCKS 4-NN, 4-X, 4-Z AND 4-BB.
037512  172427  042572                  LDF     #1000,AC0

037516  010100                          MOV     R1,R0           ;CORRECTING R0.
037520  170320                          STST    (R0)+           ;TESTING BLOCKS 33-J AND 33-P.
037522  170327  001000                  STST    #1000

                                        ;* TESTING MODE 3 REG 0 AND 7.

037526  010200                          MOV     R2,R0           ;SETTING UP R0.
037530  170530                          TSTF    @(R0)+          ;TESTING BLOCK 21-N.
037532  170537  117760                  TSTF    @#DATA

037536  010200                          MOV     R2,R0           ;CORRECTING R0.
037540  170430                          CLRF    @(R0)+          ;TESTING BLOCKS 27-U, 27-T AND 27-R.
037542  170437  117760                  CLRF    @#DATA

037546  010200                          MOV     R2,R0           ;CORRECTING R0.
037550  177030                          LDCIF   @(R0)+,AC0      ;TESTING BLOCKS 28-L, 28-N AND 28-P.
037552  177037  117760                  LDCIF   @#DATA,AC0

037556  010200                          MOV     R2,R0           ;CORRECTING R0.
037560  172430                          LDF     @(R0)+,AC0      ;TESTING BLOCKS 4-R, 4-T, 4-X, 4-Z AND 4-BB.
037562  172437  117760                  LDF     @#DATA,AC0
```

```
 6993
 6994 037566  010200                          MOV     R2,R0           ;CORRECTING R0.
 6995 037570  170330                          STST    @(R0)+          ;TESTING BLOCKS 33-L, 33-N AND 33-P.
 6996 037572  170337  117760                  STST    @#DATA
 6997
 6998                                         ;* TESTING MODE 4 REG 0.
 6999
 7000 037576  010200                          MOV     R2,R0           ;SETTING UP R0.
 7001 037600  172440                          LDF     -(R0),AC0       ;TESTING BLOCKS 4-J, 4-X, 4-Z AND 4-BB.
 7002
 7003                                         ;* TESTING MODE 5 REG 0.
 7004
 7005 037602  010300                          MOV     R3,R0           ;SETTING UP R0.
 7006 037604  170550                          TSTF    @-(R0)          ;TESTING BLOCK 21-U.
 7007
 7008 037606  010300                          MOV     R3,R0           ;CORRECTING R0.
 7009 037610  170450                          CLRF    @-(R0)          ;TESTING BLOCKS 27-X, 27-T AND 27-R.
 7010
 7011 037612  010300                          MOV     R3,R0           ;CORRECTING R0.
 7012 037614  177050                          LDCIF   @-(R0),AC0      ;TESTING BLOCKS 28-S, 28-N AND 28-P.
 7013
 7014 037616  010300                          MOV     R3,R0           ;CORRECTING R0.
 7015 037620  172450                          LDF     @-(R0),AC0      ;TESTING BLOCKS 4-U, 4-T, 4-X, 4-Z AND 4-BB.
 7016
 7017 037622  010300                          MOV     R3,R0           ;CORRECTING R0.
 7018 037624  170350                          STST    @-(R0)          ;TESTING BLOCKS 33-S, 33-N ADN 33-P.
 7019
 7020                                         ;* TESTING MODE 6 REG 7.
 7021
 7022 037626  170567  060126                  TSTF    DATA            ;TESTING BLOCK 21-O.
 7023 037632  170467  060122                  CLRF    DATA            ;TESTING BLOCKS 27-DD, 27-T AND 27-R.
 7024 037636  177067  060116                  LDCIF   DATA,AC0        ;TESTING BLOCKS 28-T, 28-N ADN 28-P.
 7025 037642  172467  060112                  LDF     DATA,AC0        ;TESTING BLOCKS 4-DD, 4-T, 4-X, 4-Z AND 4-BB.
 7026 037646  170367  060106                  STST    DATA            ;TESTING BLOCKS 33-T, 33-N AND 33-P.
 7027
 7028                                         ;* TESTING MODE 7 REG 0 AND 7.
 7029
 7030 037652  010200                          MOV     R2,R0           ;SETTING UP R0.
 7031 037654  170470  000000                  CLRF    @0(R0)          ;TESTING BLOCKS 27-GG, 27-JJ, 27-T AND 27-R.
 7032 037660  170477  060104                  CLRF    @DATA+10
 7033 037664  177070  000000                  LDCIF   @0(R0),AC0      ;TESTING BLOCKS 28-W, 28-Z, 28-N AND 28-P.
 7034 037670  177077  060074                  LDCIF   @DATA+10,AC0
 7035 037674  172470  000000                  LDF     @0(R0),AC0      ;TESTING BLOCKS 4-GG, 4-JJ, 4-T, 4-X 4-Z AND 4-BB.
 7036 037700  172477  060064                  LDF     @DATA+10,AC0
 7037 037704  170370  000000                  STST    @0(R0)          ;TESTING BLOCKS 33-W, 33-Z, 33-N AN 33-P.
 7038 037710  170377  060054                  STST    @DATA+10
 7039
 7040 037714  000431                          BR      ENDTST          ;BRANCH TO END OF TEST ROUTINE.
 7041
 7042 037716  042767  000001  137646  TRAPV:  BIC     #1,MMR0         ;TURN OFF MEMORY MANAGEMENT.
 7043 037724  016767  137642  141306          MOV     MMR0,$TMP3      ;TRANSFER MMR0 TO $TMP3 FOR ERROR PRINTING.
 7044 037732  005267  141302                  INC     $TMP3           ;REPLACE BIT CLEARED TURNING OFF MEMORY MANAGEMENT.
 7045 037736  016767  137634  141272          MOV     MMR2,$TMP2      ;MOVE THE TRAP INSTRUCTION ADDRESS TO $TMP13.
 7046 037744  005067  137622                  CLR     MMR0            ;CLEAR ERROR BITS.
 7047 037750  012667  141312                  MOV     (SP)+,$TMP16    ;POP STACK AND SAVE 1ST CONTENTS.
 7048 037754  012667  141310                  MOV     (SP)+,$TMP17    ;POP STACK AGAIN AND SAVE 2ND CONTENTS.
 7049 037760  104362                          ERROR   +362            ;FPP TRAP/ABORT ERROR CALL.
```

```
 7050 037762  016746  141302                MOV     $TMP17,-(SP)    ;PUSH 2ND SAVED CONTENTS BACK ON STACK.
 7051 037766  016746  141274                MOV     $TMP16,-(SP)    ;PUSH 1ST SAVED CONTENTS BACK ON STACK.
 7052 037772  005267  137574                INC     MMR0            ;TURN ON MEMORY MANAGEMENT.
 7053 037776  000002                        RTI                     ;RETURN FROM INTERRUPT.
 7054
 7055 040000  005067  137566        ENDTST: CLR     MMR0            ;TURN OFF MEMORY MANAGEMENT.
 7056 040004  016767  141252  140236        MOV     $TMP14,MMVECT   ;RESTORE MMVECT TO ITS ORIGINAL CONTENTS.
 7057 040012                        IDONE:
      040012  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                             ;SEE IF THE USER HAS EXPRESSED
                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                             ;THE USER TYPED CONTROL G?).
 7058
 7067                                ;;*****************************************************************
                                     ;*TEST 77       AUTO INCREMENT/DECREMENT TEST
                                     ;*
                                     ;* THIS TEST INSURES THAT AUTO INCREMENT/DECREMENT WORKS PROPERLY AND
                                     ;* *ONLY* WHEN IT IS SUPPOSED TO. THIS IS DONE BY ENABLING 22-BIT KERNEL
                                     ;* D-SPACE, BUT MAKING IT NON-RESIDENT, FORCING A MEMORY MANAGEMENT TRAP
                                     ;* CONDITION.  THIS ENABLES EXAMINING OF SR1 FOR PROPER CONTENTS.
                                     ;*
                                     ;;*****************************************************************
      040014  000004                 TST77:  SCOPE
 7068 040016                         INCDCT:
      040016  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 7069 040020  005067  137546                CLR     MMR0            ;MAKE SURE MEMORY MANAGEMENT IS OFF.
 7070 040024  170127  040000                LDFPS   #40000          ;LOAD FLOATING POINT STATUS.
 7071 040030  005067  141236                CLR     $TMP20          ;CLEAR THE TEMPORARY LOCATION.
 7072 040034  012767  000252  002360        MOV     #252,STORE      ;CLEAR UPPER BYTE - ALTERNATING BITS IN LOWER BYTE.
 7073 040042  012767  125252  002354        MOV     #125252,STORE+2 ;MOVE ALTERNATING BITS TO 2ND WORD.
 7074 040050  012767  125252  002350        MOV     #125252,STORE+4 ;MOVE ALTERNATING BITS TO 3RD WORD.
 7075 040056  012767  125252  002344        MOV     #125252,STORE+6 ;MOVE ALTERNATING BITS TO 4TH WORD.
 7076 040064  172467  002332                LDF     STORE,AC0       ;LOAD AC0.
 7077 040070  172567  002326                LDF     STORE,AC1       ;LOAD AC1.
 7078 040074  172667  002322                LDF     STORE,AC2       ;LOAD AC2.
 7079 040100  172767  002316                LDF     STORE,AC3       ;LOAD AC3.
 7080 040104  012700  042422                MOV     #STORE,R0       ;MOVE ADDRESS OF STORE TO R0.
 7081 040110  012701  000030                MOV     #30,R1          ;MOVE LOOP COUNTER (CLEARING 30 WORDS) TO R1.
 7082 040114  005020                1$:     CLR     (R0)+           ;CLEAR THE WORD.
 7083 040116  077102                        SOB     R1,1$           ;SUBTRACT 1 FROM R1 AND BRANCH IF NOT 0.
 7084 040120  174067  002276                STF     AC0,STORE       ;STORE AC0.
 7085 040124  174167  002302                STF     AC1,STORE+10    ;STORE AC1.
 7086 040130  174267  002306                STF     AC2,STORE+20    ;STORE AC2.
 7087 040134  174367  002312                STF     AC3,STORE+30    ;STORE AC3.
 7088
 7089 040140  012767  077406  132152        MOV     #77406,KDPDR0   ;MAKE KDPDR0 RESIDENT.
 7090 040146  012767  077406  132146        MOV     #77406,KDPDR1   ;MAKE KDPDR1 RESIDENT.
 7091 040154  012767  077400  132142        MOV     #77400,KDPDR2   ;MAKE KDPDR2 NON-RESIDENT.
 7092 040162  012767  077406  132136        MOV     #77406,KDPDR3   ;MAKE KDPDR3 RESIDENT.
 7093 040170  012767  077406  132140        MOV     #77406,KDPDR7   ;MAKE KDPDR7 RESIDENT.
 7094
 7095 040176  012767  077406  132074        MOV     #77406,KIPDR0   ;MAKE KIPDR0 RESIDENT.
 7096 040204  012767  077406  132070        MOV     #77406,KIPDR1   ;MAKE KIPDR1 RESIDENT.
 7097 040212  012767  077406  132064        MOV     #77406,KIPDR2   ;MAKE KIPDR2 RESIDENT.
 7098 040220  012767  077406  132060        MOV     #77406,KIPDR3   ;MAKE KIPDR3 RESIDENT.
 7099 040226  012767  077406  132062        MOV     #77406,KIPDR7   ;MAKE KIPDR7 RESIDENT.
```

```
7100
7101 040234  005067  132120              CLR     KDPAR0          ;MAP D-PAGE 0 FOR 0-4K.
7102 040240  012767  000200  132114      MOV     #200,KDPAR1     ;MAP D-PAGE 1 FOR 4-8K.
7103 040246  012767  000400  132110      MOV     #400,KDPAR2     ;MAP D-PAGE 2 FOR 8-12K.
7104 040254  012767  000600  132104      MOV     #600,KDPAR3     ;MAP D-PAGE 3 FOR 12-16K.
7105 040262  012767  177600  132106      MOV     #177600,KDPAR7  ;MAP D-PAGE 7 FOR I/O PAGE.
7106
7107 040270  005067  132044              CLR     KIPAR0          ;MAP I-PAGE 0 FOR 0-4K.
7108 040274  012767  000200  132040      MOV     #200,KIPAR1     ;MAP I-PAGE 1 FOR 4-8K.
7109 040302  012767  000400  132034      MOV     #400,KIPAR2     ;MAP I-PAGE 2 FOR 8-12K.
7110 040310  012767  000600  132030      MOV     #600,KIPAR3     ;MAP I-PAGE 3 FOR 12-16K.
7111 040316  012767  177600  132032      MOV     #177600,KIPAR7  ;MAP I-PAGE 7 FOR I/O PAGE.
7112
7113 040324  016767  137720  140730      MOV     MMVECT,$TMP14   ;TEMPORARILY STORE THE MMVECT VALUE.
7114 040332  012737  041242  000250      MOV     #TRPV,@#MMVECT  ;SET UP FOR FP TRAPS FOR THIS TEST.
7115 040340  012737  000340  000252      MOV     #340,@#MMVECT+2
7116 040346  016767  137446  140710      MOV     IOTRAP,$TMP15   ;TEMPORARILY STORE THE IOTRAP VALUE.
7117 040354  012737  041156  000020      MOV     #FALTRP,@#IOTRAP;SET UP FOR FAILURE OF TRAPS FOR THIS TEST.
7118 040362  012737  000340  000022      MOV     #340,@#IOTRAP+2
7119
7120 040370  012767  000024  132120      MOV     #24,MMR3        ;TURN ON 22-BIT KERNEL D-SPACE.
7121 040376  012767  042406  002012      MOV     #NODAT,NODAT+10 ;SET UP ADDRESS POINTER.
7122 040404  012700  042406              MOV     #NODAT,R0       ;SET UP R0.
7123 040410  012702  042416              MOV     #NODAT+10,R2    ;SET UP R2.
7124 040414  012703  042420              MOV     #NODAT+12,R3    ;SET UP R3.
7125 040420  010067  002036              MOV     R0,STORE+40     ;STORE R0.
7126 040424  010267  002034              MOV     R2,STORE+42     ;STORE R2.
7127 040430  010367  002032              MOV     R3,STORE+44     ;STORE R3.
7128 040434  005267  137132              INC     MMR0            ;TURN ON MEMORY MANAGEMENT.
7129
7130                                     ;*****ALL REFERENCES TO MICRO-FLOWS REFER TO *FP11-F-2 REV A* FLOWS*****
7131                                     ;THE COMMENTS FOR EACH TEST LINE ARE WRITTEN SO YOU CAN GO TO THE MICRO
7132                                     ;FLOW AND PINPOINT THE PROBLEM AREA.  FROM THERE, HARDWARE ANALYSIS SHOULD
7133                                     ;BE EASIER.
7134
7135                                     ;* THE FOLLOWING TESTS ARE FOR MODE 0 REG 1 (THESE SHOULD *NOT* ABORT).
7136
7137 040440  170501                      TSTF    R1              ;FDST-NOTCLR PAGE 21.
7138 040442  170401                      CLRF    R1              ;FDST MODES PAGE 27.
7139 040444  177001                      LDCIF   R1,AC0          ;SOURCE MODES PAGE 28.
7140 040446  172401                      LDF     R1,AC0          ;FSRC MODES PAGE 4.
7141 040450  170301                      STST    R1              ;DEST MODES PAGE 33.
7142 040452  005067  137114              CLR     MMR0            ;TURN OFF MEMORY MANAGEMENT.
7143 040456  172467  001740              LDF     STORE,AC0       ;RESTORE AC0.
7144 040462  172567  001734              LDF     STORE,AC1       ;RESTORE AC1.
7145 040466  005267  137100              INC     MMR0            ;TURN ON MEMORY MANAGEMENT.
7146
7147                                     ;* THE FOLLOWING TESTS ARE FOR MODE 1 REG 1.
7148
7149 040472  010001          MODE1:      MOV     R0,R1           ;SET UP R1.
7150 040474  010004                      MOV     R0,R4           ;MOVE 'START' VALUE INTO R4.
7151 040476  170511                      TSTF    (R1)            ;FDST-NOTCLR PAGE 21.
7152 040500  000004                      IOT                     ;FORCE A TRAP.
7153 040502  170411                      CLRF    (R1)            ;FDST MODES PAGE 27.
7154 040504  000004                      IOT                     ;FORCE A TRAP.
7155 040506  177011                      LDCIF   (R1),AC0        ;SOURCE MODES PAGE 28.
7156 040510  000004                      IOT                     ;FORCE A TRAP.
```

```
7157 040512  172411                        LDF     (R1),AC0        ;FSRC MODES PAGE 4.
7158 040514  000004                        IOT                     ;FORCE A TRAP.
7159 040516  170311                        STST    (R1)            ;DEST MODES PAGE 33.
7160 040520  000004                        IOT                     ;FORCE A TRAP.
7161
7162                                       ;* THE FOLLOWING TESTS ARE FOR MODE 2 REG 1.
7163
7164 040522  170521                        TSTF    (R1)+           ;FDST-NOTCLR PAGE 21.
7165 040524  000004                LABEL1: IOT                     ;FORCE A TRAP.
7166
7167 040526  010001                        MOV     R0,R1           ;CORRECT R1.
7168 040530  170421                        CLRF    (R1)+           ;FDST MODES PAGE 27.
7169 040532  000004                        IOT                     ;FORCE A TRAP.
7170
7171 040534  010001                        MOV     R0,R1           ;CORRECT R1.
7172 040536  177021                        LDCIF   (R1)+,AC0       ;SOURCE MODES PAGE 28.
7173 040540  000004                        IOT                     ;FORCE A TRAP.
7174
7175 040542  010001                        MOV     R0,R1           ;CORRECT R1.
7176 040544  172421                        LDF     (R1)+,AC0       ;FSRC MODES PAGE 4.
7177 040546  000004                        IOT                     ;FORCE A TRAP.
7178
7179 040550  010001                        MOV     R0,R1           ;CORRECT R1.
7180 040552  170321                        STST    (R1)+           ;DEST MODES PAGE 33.
7181 040554  000004                        IOT                     ;FORCE A TRAP.
7182
7183                                       ;* THE FOLLOWING TESTS ARE FOR MODE 3 REG 1 AND 7.
7184
7185 040556  010201                        MOV     R2,R1           ;SET UP R1 FOR MODE 3.
7186 040560  010204                        MOV     R2,R4           ;MOVE 'START' VALUE INTO R4.
7187 040562  170531                        TSTF    @(R1)+          ;FDST-NOTCLR PAGE 21.
7188 040564  000004                        IOT                     ;FORCE A TRAP.
7189 040566  170537  042406                TSTF    @#NODAT
7190 040572  000004                        IOT                     ;FORCE A TRAP.
7191
7192 040574  010201                        MOV     R2,R1           ;CORRECT R1.
7193 040576  170431                        CLRF    @(R1)+          ;FDST MODES PAGE 27.
7194 040600  000004                        IOT                     ;FORCE A TRAP.
7195 040602  170437  042406                CLRF    @#NODAT
7196 040606  000004                        IOT                     ;FORCE A TRAP.
7197
7198 040610  010201                        MOV     R2,R1           ;CORRECT R1.
7199 040612  177031                        LDCIF   @(R1)+,AC0      ;SOURCE MODES PAGE 28.
7200 040614  000004                        IOT                     ;FORCE A TRAP.
7201 040616  177037  042406                LDCIF   @#NODAT,AC0
7202 040622  000004                        IOT                     ;FORCE A TRAP.
7203
7204 040624  010201                        MOV     R2,R1           ;CORRECT R1.
7205 040626  172431                        LDF     @(R1)+,AC0      ;FSRC MODES PAGE 4.
7206 040630  000004                        IOT                     ;FORCE A TRAP.
7207 040632  172437  042406                LDF     @#NODAT,AC0
7208 040636  000004                        IOT                     ;FORCE A TRAP.
7209
7210 040640  010201                        MOV     R2,R1           ;CORRECT R1.
7211 040642  170331                        STST    @(R1)+          ;DEST MODES PAGE 33.
7212 040644  000004                        IOT                     ;FORCE A TRAP.
7213 040646  170337  042406                STST    @#NODAT
```

```
7214 040652  000004                    IOT                     ;FORCE A TRAP.
7215
7216                                    ;* THE FOLLOWING TESTS ARE FOR MODE 4 REG 1.
7217
7218 040654  010201                    MOV     R2,R1           ;SET UP R1 FOR MODE 4.
7219 040656  170541                    TSTF    -(R1)           ;FDST-NOTCLR PAGE 21.
7220 040660  000004                    IOT                     ;FORCE A TRAP.
7221
7222 040662  010201                    MOV     R2,R1           ;CORRECT R1.
7223 040664  170441                    CLRF    -(R1)           ;FDST MODES PAGE 27.
7224 040666  000004                    IOT                     ;FORCE A TRAP.
7225
7226 040670  010201                    MOV     R2,R1           ;CORRECT R1.
7227 040672  177041                    LDCIF   -(R1),AC0       ;SOURCE MODES PAGE 28.
7228 040674  000004                    IOT                     ;FORCE A TRAP.
7229
7230 040676  010201                    MOV     R2,R1           ;CORRECT R1.
7231 040700  172441                    LDF     -(R1),AC0       ;FSRC MODES PAGE 4.
7232 040702  000004                    IOT                     ;FORCE A TRAP.
7233
7234 040704  010201                    MOV     R2,R1           ;CORRECT R1.
7235 040706  170341                    STST    -(R1)           ;DEST MODES PAGE 33.
7236 040710  000004                    IOT                     ;FORCE A TRAP.
7237
7238                                    ;* THE FOLLOWING TESTS ARE FOR MODE 5 REG 1.
7239
7240 040712  010301                    MOV     R3,R1           ;SET UP R1 FOR MODE 5.
7241 040714  010304                    MOV     R3,R4           ;MOVE 'START' VALUE INTO R4.
7242 040716  170551                    TSTF    @-(R1)          ;FDST-NOTCLR PAGE 21.
7243 040720  000004                    IOT                     ;FORCE A TRAP.
7244
7245 040722  010301                    MOV     R3,R1           ;CORRECT R1.
7246 040724  170451                    CLRF    @-(R1)          ;FDST MODES PAGE 27.
7247 040726  000004                    IOT                     ;FORCE A TRAP.
7248
7249 040730  010301                    MOV     R3,R1           ;CORRECT R1.
7250 040732  177051                    LDCIF   @-(R1),AC0      ;SOURCE MODES PAGE 28.
7251 040734  000004                    IOT                     ;FORCE A TRAP.
7252
7253 040736  010301                    MOV     R3,R1           ;CORRECT R1.
7254 040740  172451                    LDF     @-(R1),AC0      ;FSRC MODES PAGE 4.
7255 040742  000004                    IOT                     ;FORCE A TRAP.
7256
7257 040744  010301                    MOV     R3,R1           ;CORRECT R1.
7258 040746  170351                    STST    @-(R1)          ;DEST MODES PAGE 33.
7259 040750  000004                    IOT                     ;FORCE A TRAP.
7260
7261                                    ;* THE FOLLOWING TESTS ARE FOR MODE 6 REG 1 AND 7.
7262
7263 040752  010001                    MOV     R0,R1           ;SET UP R1 FOR MODE 6.
7264 040754  010004                    MOV     R0,R4           ;MOVE 'START' VALUE INTO R4.
7265 040756  170561  000000            TSTF    0(R1)           ;FDST-NOTCLR PAGE 21.
7266 040762  000004                    IOT                     ;FORCE A TRAP.
7267 040764  170567  001416            TSTF    NODAT
7268 040770  000004                    IOT                     ;FORCE A TRAP.
7269 040772  170461  000000            CLRF    0(R1)           ;FDST MODES PAGE 27.
7270 040776  000004                    IOT                     ;FORCE A TRAP.
```

```
7271 041000  170467  001402                CLRF    NODAT
7272 041004  000004                        IOT                     ;FORCE A TRAP.
7273 041006  177061  000000                LDCIF   0(R1),AC0        ;SOURCE MODES PAGE 28.
7274 041012  000004                        IOT                     ;FORCE A TRAP.
7275 041014  177067  001366                LDCIF   NODAT,AC0
7276 041020  000004                        IOT                     ;FORCE A TRAP.
7277 041022  172461  000000                LDF     0(R1),AC0       ;FSRC MODES PAGE 4.
7278 041026  000004                        IOT                     ;FORCE A TRAP.
7279 041030  172467  001352                LDF     NODAT,AC0
7280 041034  000004                        IOT                     ;FORCE A TRAP.
7281 041036  170361  000000                STST    0(R1)           ;DEST MODES PAGE 33.
7282 041042  000004                        IOT                     ;FORCE A TRAP.
7283 041044  170367  001336                STST    NODAT
7284 041050  000004                        IOT                     ;FORCE A TRAP.
7285
7286                                       ;* THE FOLLOWING TESTS ARE FOR MODE 7 REG 1 AND 7.
7287
7288 041052  010201                        MOV     R2,R1           ;SET UP R1 FOR MODE 7.
7289 041054  010204                        MOV     R2,R4           ;MOVE 'START' VALUE TO R4.
7290 041056  170571  000000                TSTF    @0(R1)          ;FDST-NOTCLR PAGE 21.
7291 041062  000004                        IOT                     ;FORCE A TRAP.
7292 041064  170577  001326                TSTF    @NODAT+10
7293 041070  000004                        IOT                     ;FORCE A TRAP.
7294 041072  170471  000000                CLRF    @0(R1)          ;FDST MODES PAGE 27.
7295 041076  000004                        IOT                     ;FORCE A TRAP.
7296 041100  170477  001312                CLRF    @NODAT+10
7297 041104  000004                        IOT                     ;FORCE A TRAP.
7298 041106  177071  000000                LDCIF   @0(R1),AC0      ;SOURCE MODES PAGE 28.
7299 041112  000004                        IOT                     ;FORCE A TRAP.
7300 041114  177077  001276                LDCIF   @NODAT+10,AC0
7301 041120  000004                        IOT                     ;FORCE A TRAP.
7302 041122  172471  000000                LDF     @0(R1),AC0      ;FSRC MODES PAGE 4.
7303 041126  000004                        IOT                     ;FORCE A TRAP.
7304 041130  172477  001262                LDF     @NODAT+10,AC0
7305 041134  000004                        IOT                     ;FORCE A TRAP.
7306 041136  170371  000000                STST    @0(R1)          ;DEST MODES PAGE 33.
7307 041142  000004                        IOT                     ;FORCE A TRAP.
7308 041144  170377  001246                STST    @NODAT+10
7309 041150  000004                        IOT                     ;FORCE A TRAP.
7310 041152  000167  001324                JMP     ENDTES          ;BRANCH TO END TEST.
7311
7312 041156  005067  136410        FALTRP: CLR     MMR0            ;TURN OFF MEMORY MANAGEMENT.
7313 041162  011667  140072                MOV     (SP),$TMP13     ;MOVE NEXT INSTRUCTION ADDRESS TO $TMP13.
7314
7315                                       ;THIS NEXT SECTION NOW CORRECTS THE CONTENTS OF $TMP13 SO THAT IT POINTS
7316                                       ;TO THE PREVIOUS FPP INSTRUCTION.  IT DOES THIS BY SUBTRACTING 2 FROM THE
7317                                       ;ADDRESS IN $TMP13, REPLACING THE 170000 THAT THE BIC INSTRUCTION USES,
7318                                       ;AND BIT CLEARING THE INSTRUCTION WITH 170000.  IF THE INSTRUCTION THAT
7319                                       ;$TMP13 IS POINTING TO IS NOT AN FPP INSTRUCTION, THE 170000 WILL NOT
7320                                       ;CLEAR, SATISFYING THE NEXT BRANCH.  THE ADDRESS IS AGAIN CORRECTED,
7321                                       ;AND THE TESTING PROCESS STARTS OVER.  THIS CONTINUES UNTIL $TMP13 IS
7322                                       ;POITING TO AN FPP INSTRUCTION, AND NORMALLY WILL NOT BE EXECUTED MORE
7323                                       ;THAN THREE TIMES BEFORE FINDING THE INSTRUCTION.
7324 041166  162767  000002  140064  1$:   SUB     #2,$TMP13       ;SUBTRACT 2 FROM $TMP13.
7325 041174  012767  170000  000004        MOV     #170000,2$+4    ;SET UP BIC DATA LOCATION.
7326 041202  047727  140052  170000  2$:   BIC     @$TMP13,#170000 ;TEST TO SEE IF FPP INSTRUCTION.
7327 041210  001366                        BNE     1$              ;BRANCH BACK FOR ANOTHER TRY IF NOT.
```

```
7328 041212  012667  140050                    MOV     (SP)+,$TMP16    ;SAVE 1ST CONTENTS OF STACK AND POP IT ONCE.
7329 041216  012667  140046                    MOV     (SP)+,$TMP17    ;SAVE 2ND CONTENTS OF STACK AND POP IT AGAIN.
7330 041222  104364                           ERROR   +364            ;FAILURE TO ABORT ON INSTRUCTION ERROR.
7331 041224  016746  140040                    MOV     $TMP17,-(SP)    ;PUSH 2ND SAVED CONTENTS BACK ON STACK.
7332 041230  016746  140032                    MOV     $TMP16,-(SP)    ;PUSH 1ST SAVED CONTENTS BACK ON STACK.
7333 041234  005267  136332                    INC     MMR0            ;TURN MEMORY MANAGEMENT BACK ON.
7334 041240  000002                           RTI                     ;RETURN FROM INTERRUPT.
7335
7336 041242  016767  136326  137766  TRPV:   MOV     SR1,$TMP2       ;MOVE SR1 TO $TMP2 FOR TESTING.
7337 041250  016767  136322  140002          MOV     MMR2,$TMP13     ;TRANSFER ADDRESS OF INST. CAUSING TRAP TO $TMP13.
7338 041256  005067  136310                    CLR     MMR0            ;TURN OFF MEMORY MANAGEMENT.
7339 041262  112767  177776  140002          MOVB    #-2,$TMP20      ;MOVE -2 TO LOWER BYTE IN ERROR POINTER.
7340 041270  022767  040472  137762          CMP     #MODE1,$TMP13   ;SEE IF INSTRUCTION CAUSING TRAP IS BEFORE MODE 1 (MODE 0).
7341 041276  002402                           BLT     1$              ;BRANCH AROUND MODE 0 ERROR JUMP IF NOT.
7342 041300  000167  000754                    JMP     73$             ;JUMP TO ERROR NEST.
7343 041304  017767  137750  137754  1$:     MOV     @$TMP13,$TMP16  ;MOVE INSTRUCTION CAUSING TRAP TO $TMP16.
7344 041312  112767  177777  137752          MOVB    #-1,$TMP20      ;MOVE -1 TO LOWER BYTE IN ERROR POINTER.
7345 041320  005067  137714                    CLR     $TMP3           ;CLEAR CALCULATED LOCATION.
7346 041324  012767  041516  137712          MOV     #65$,$TMP5      ;MOVE NEXT CHECK ADDRESS TO $TMP5.
7347 041332  022767  040524  137720          CMP     #LABEL1,$TMP13  ;SEE IF TRAP IS BEFORE MODE 2 REG 1 CLRF INST.
7348 041340  100007                           BPL     21$             ;BRANCH TO SR1=0 TEST IF SO.
7349 041342  012767  000060  000004          MOV     #60,2$+4        ;SET UP BIC DATA POSITION.
7350 041350  046727  137712  000060  2$:     BIC     $TMP16,#60      ;TEST TO SEE IF MODE 6 OR 7 INSTRUCTION.
7351 041356  001005                           BNE     4$              ;BRANCH TO FURTHER TESTS IF NOT.
7352 041360  005767  137652          21$:    TST     $TMP2           ;TEST TO SEE IF SR1=0.
7353 041364  001454                           BEQ     65$             ;BRANCH TO NEXT CHECK IF OK.
7354 041366  000167  000666                    JMP     73$             ;JUMP TO ERROR NEST IF NOT.
7355                                          ;THIS NEXT ROUTINE DETERMINES WHICH REGISTER WAS IN THE INSTRUCTION, AND
7356                                          ;LOADS THE START AND END VALUES OF EITHER R1 OR R7 (PROGRAM COUNTER) INTO
7357                                          ;$TMP17 AND $TMP3 RESPECTIVELY.  THEY ARE THEN SUBTRACTED TO FIND THE
7358                                          ;DIFFERENCE THAT ACTUALLY OCCURED.  IF NO DIFFERENCE WAS FOUND, THE TEST
7359                                          ;FOR ZERO IN SR1 IS ACCOMPLISHED.  IF A DIFFERENCE IS FOUND, THE DIFFERENCE
7360                                          ;IS SHIFTED LEFT 3 PLACES, THE TOP BYTE IS CLEARED, AND THE REGISTER
7361                                          ;OF THE INSTRUCTION IS ADDED.  $TMP3 NOW CONTAINS WHAT SHOULD APPEAR
7362                                          ;IN SR1, ACCORDING TO WHAT ACTUALLY HAPPENED TO THE REGISTER.
7363 041372  042767  177770  137666  4$:     BIC     #177770,$TMP16  ;BIT CLEAR THE INSTRUCTION, LEAVING THE REG EXPOSED.
7364 041400  026727  137662  000002          CMP     $TMP16,#2       ;COMPARE REGISTER TO DETERMINE IF IT IS REG 7.
7365 041406  003005                           BGT     5$              ;BRANCH TO THE REG 7 SETUP IF EQUAL TO REG 7.
7366 041410  010467  137654                    MOV     R4,$TMP17       ;MOVE THE START VALUE TO $TMP17.
7367 041414  010167  137620                    MOV     R1,$TMP3        ;MOVE THE END VALUE TO $TMP3.
7368 041420  000410                           BR      6$              ;BRANCH TO CONTINUE.
7369 041422  016767  137632  137640  5$:     MOV     $TMP13,$TMP17   ;MOVE THE START VALUE TO $TMP17.
7370 041430  062767  000002  137632          ADD     #2,$TMP17       ;ADD 2 TO START VALUE FOR NORMAL INCREMENTING.
7371 041436  011667  137576                    MOV     (SP),$TMP3      ;MOVE THE END VALUE TO $TMP3.
7372 041442  166767  137622  137570  6$:     SUB     $TMP17,$TMP3    ;FIND THE DIFFERENCE THAT OCCURED.
7373 041450  001743                           BEQ     21$             ;BRANCH TO TEST FOR SR1=0 IF NO DIFFERENCE.
7374 041452  006367  137562                    ASL     $TMP3           ;ARITHMETIC SHIFT LEFT $TMP3 3
7375 041456  006367  137556                    ASL     $TMP3           ;PLACES TO PUT DIFFERENCE FOUND
7376 041462  006367  137552                    ASL     $TMP3           ;IN BITS 3 THROUGH 7.
7377 041466  042767  177400  137544          BIC     #177400,$TMP3   ;BIT CLEAR UPPER BYTE OF $TMP3.
7378 041474  066767  137566  137536          ADD     $TMP16,$TMP3    ;ADD THE REGISTER THAT WAS CHANGED, AND
7379 041502  026767  137530  137530          CMP     $TMP2,$TMP3     ;COMPARE SR1 WITH CALCULATED DATA.
7380 041510  001402                           BEQ     65$             ;BRANCH AROUND ERROR JUMP IF OK.
7381 041512  000167  000446                    JMP     7$              ;JUMP TO ERROR REPORT IF INCORRECT.
7382 041516  032767  000400  137546  65$:     BIT     #400,$TMP20     ;TEST TO SEE IF BIT 8 IS SET.
7383 041524  001402                           BEQ     66$             ;BRANCH AROUND AC SKIP JUMP IF NOT.
7384 041526  000167  000624                    JMP     8$              ;JUMP TO RETURN - AC TESTS ARE TO BE SKIPPED.
```

```
 7385 041532  105067  137534          66$:   CLRB    $TMP20          ;CLEAR LOWER BYTE OF $TMP20 FOR ERROR CALL POINTING.
 7386 041536  010067  000736                 MOV     R0,STORE+56     ;STORE R0 FOR USE LATER IN THIS ROUTINE.
 7387 041542  005067  137470                 CLR     $TMP2           ;MOVE A '0' IN 'AC CHANGED' LOCATION.
 7388 041546  012767  041612  137470         MOV     #101$,$TMP5     ;MOVE RETURN TO $TMP5.
 7389 041554  173467  000642                 CMPF    STORE,AC0       ;SEE IF AC0 WAS CHANGED.
 7390 041560  170000                         CFCC                    ;COPY FPP CONDITION CODES TO CPU CODES.
 7391 041562  001413                         BEQ     101$            ;BRANCH TO NEXT TEST IF OK.
 7392 041564  174067  000700                 STF     AC0,STORE+46    ;STORE ACTUAL AC0 FOR ERROR PRINTING.
 7393 041570  012700  042422                 MOV     #STORE,R0       ;MOVE ADDRESS OF EXPECTED AC0 TO R0.
 7394                              ;THE NEXT TWO INSTRUCTIONS TRY TO RESTORE THE ACCUMULATOR AND CHECK THE ACCUMULATOR
 7395                              ;TO MAKE SURE IT WAS RESTORED PROPERLY FOR THE NEXT RUN THROUGH THIS TRAP HANDLER.
 7396                              ;IT IS *IMPORTANT* TO REALIZE THAT IF THE ''CMPF'' FINDS A DIFFERENCE, THAT THE
 7397                              ;*FLOATING*POINT*STATUS* IS BEING CHANGED MISTAKENLY.  AN ERROR IN THE MICROCODE
 7398                              ;HAS BEEN FOUND TO CAUSE THIS, SO CHECK THE REVISION OF THE ROM/PROM SET IN THE
 7399                              ;FPP YOU HAVE.  IF YOU DO HAVE WHAT *SEEMS* TO BE THE LATEST REV, A NEW REV WILL
 7400                              ;BE COMMING OUT TO CORRECT THIS PROBLEM.  THIS SAME ''LDF/CMPF'' SET OF RESTORE/
 7401                              ;CHECK INSTRUCTIONS IS ACCOMPLISHED FOR EACH ACCUMULATOR CHECK. IT IS ALSO
 7402                              ;IMPORTANT TO NOTE THAT IF AN ACCUMULATOR FAILS TO RESTORE PROPERLY, SUBSEQUENT
 7403                              ;PASSES THROUGH THE TRAP HANDLER WILL SKIP THE ACCUMULATOR CHECKS DUE TO THE
 7404                              ;BIT TEST #400 ABOVE.  FOR EXAMPLE, IF AC0 FAILS TO LOAD PROPERLY, AC1 THROUGH
 7405                              ;AC3 WILL STILL BE CHECKED.  AS SOON AS ANOTHER FPP INSTRUCTION TRAPS IN THE
 7406                              ;MAIN TEST, ALL *FURTHER* AC0-AC3 CHECKS WILL BE SKIPPED.
 7407 041574  172467  000622                 LDF     STORE,AC0       ;RESTORE AC0.
 7408 041600  173467  000616                 CMPF    STORE,AC0       ;SEE IF IT WAS RESTORED PROPERLY.
 7409 041604  170000                         CFCC                    ;COPY FPP CONDITION CODES TO CPU CODES.
 7410 041606  001566                         BEQ     7$              ;BRANCH TO ERROR CALL IF OK.
 7411 041610  000476                         BR      113$            ;BRANCH TO ERROR SETUP ROUTINE.
 7412 041612  012767  000001  137416  101$:  MOV     #1,$TMP2        ;PUT A '1' IN 'AC CHANGED' LOCATION.
 7413 041620  012767  041664  137416         MOV     #102$,$TMP5     ;MOVE RETURN TO $TMP5.
 7414 041626  173567  000600                 CMPF    STORE+10,AC1    ;SEE IF AC1 WAS CHANGED.
 7415 041632  170000                         CFCC                    ;COPY FPP CONDITION CODES TO CPU CODES.
 7416 041634  001413                         BEQ     102$            ;BRANCH TO NEXT TEST IF OK.
 7417 041636  174167  000626                 STF     AC1,STORE+46    ;STORE ACTUAL AC1 FOR ERROR PRINTING.
 7418 041642  012700  042432                 MOV     #STORE+10,R0    ;MOVE ADDRESS OF EXPECTED AC1 TO R0.
 7419 041646  172567  000560                 LDF     STORE+10,AC1    ;RESTORE AC1.
 7420 041652  173567  000554                 CMPF    STORE+10,AC1    ;SEE IF IT WAS RESTORED PROPERLY.
 7421 041656  170000                         CFCC                    ;COPY FPP CONDITION CODES TO CPU CODES.
 7422 041660  001541                         BEQ     7$              ;BRANCH TO ERROR CALL IF OK.
 7423 041662  000451                         BR      113$            ;BRANCH TO ERROR SETUP ROUTINE.
 7424 041664  012767  000002  137344  102$:  MOV     #2,$TMP2        ;PUT A '2' IN 'AC CHANGED' LOCATION.
 7425 041672  012767  041736  137344         MOV     #103$,$TMP5     ;MOVE RETURN TO $TMP5.
 7426 041700  173667  000536                 CMPF    STORE+20,AC2    ;SEE IF AC2 WAS CHANGED.
 7427 041704  170000                         CFCC                    ;COPY FPP CONDITION CODES TO CPU CODES.
 7428 041706  001413                         BEQ     103$            ;BRANCH TO NEXT TEST IF OK.
 7429 041710  174267  000554                 STF     AC2,STORE+46    ;STORE ACTUAL AC2 FOR ERROR PRINTING.
 7430 041714  012700  042442                 MOV     #STORE+20,R0    ;MOVE ADDRESS OF EXPECTED AC2 TO R0.
 7431 041720  172667  000516                 LDF     STORE+20,AC2    ;RESTORE AC2.
 7432 041724  173667  000512                 CMPF    STORE+20,AC2    ;SEE IF IT WAS RESTORED PROPERLY.
 7433 041730  170000                         CFCC                    ;COPY FPP CONDITION CODES TO CPU CODES.
 7434 041732  001514                         BEQ     7$              ;BRANCH TO ERROR CALL IF OK.
 7435 041734  000424                         BR      113$            ;BRANCH TO ERROR SETUP ROUTINE.
 7436 041736  012767  000003  137272  103$:  MOV     #3,$TMP2        ;PUT A '3' IN 'AC CHANGED' LOCATION.
 7437 041744  012767  042016  137272         MOV     #100$,$TMP5     ;MOVE RETURN TO $TMP5.
 7438 041752  173767  000474                 CMPF    STORE+30,AC3    ;SEE IF AC3 WAS CHANGED.
 7439 041756  170000                         CFCC                    ;COPY FPP CONDITION CODES TO CPU CODES.
 7440 041760  001416                         BEQ     100$            ;BRANCH TO NEXT TEST IF OK.
 7441 041762  174367  000502                 STF     AC3,STORE+46    ;STORE ACTUAL AC3 FOR ERROR PRINTING.
```

```
7442 041766 012700 042452                MOV   #STORE+30,R0    ;MOVE ADDRESS OF EXPECTED AC3 TO R0.
7443 041772 172767 000454                LDF   STORE+30,AC3    ;RESTORE AC3.
7444 041776 173767 000450                CMPF  STORE+30,AC3    ;SEE IF IT WAS RESTORED PROPERLY.
7445 042002 170000                       CFCC                  ;COPY FPP CONDITION CODES TO CPU CODES.
7446 042004 001467                       BEQ   7$              ;BRANCH TO ERROR CALL IF OK.
7447 042006 012767 000402 137256 113$:   MOV   #402,$TMP20     ;MOVE 402 TO ERROR POINTER.
7448 042014 000463                       BR    7$              ;BRANCH TO ERROR CALL.
7449 042016 005067 137214        100$:   CLR   $TMP2           ;CLEAR 'REGISTER CHANGED' LOCATION.
7450 042022 105267 137244                INCB  $TMP20          ;SET ERROR POINTER BYTE.
7451 042026 012767 042062 137210         MOV   #120$,$TMP5     ;MOVE RETURN TO $TMP5.
7452 042034 026700 000422                CMP   STORE+40,R0     ;SEE IF R0 WAS CHANGED.
7453 042040 001410                       BEQ   120$            ;BRANCH TO NEXT TEST IF OK.
7454 042042 010067 137200                MOV   R0,$TMP6        ;MOVE ACTUAL R0 TO LOCATION FOR ERROR PRINTING.
7455 042046 016767 000410 137164         MOV   STORE+40,$TMP3  ;MOVE EXPECTED TO LOCATION FOR ERROR PRINTING.
7456 042054 016700 000402                MOV   STORE+40,R0     ;RESTORE R0.
7457 042060 000441                       BR    7$              ;BRANCH TO ERROR CALL.
7458 042062 012767 000002 137146 120$:   MOV   #2,$TMP2        ;PUT A '2' IN 'REGISTER CHANGED' LOCATION.
7459 042070 012767 042124 137146         MOV   #130$,$TMP5     ;MOVE RETURN TO $TMP5.
7460 042076 026702 000362                CMP   STORE+42,R2     ;SEE IF R2 WAS CHANGED.
7461 042102 001410                       BEQ   130$            ;BRANCH TO NEXT TEST IF OK.
7462 042104 010267 137136                MOV   R2,$TMP6        ;MOVE ACTUAL R2 TO LOCATION FOR ERROR PRINTING.
7463 042110 016767 000350 137122         MOV   STORE+42,$TMP3  ;MOVE EXPECTED TO LOCATION FOR ERROR PRINTING.
7464 042116 016702 000342                MOV   STORE+42,R2     ;RESTORE R2.
7465 042122 000420                       BR    7$              ;BRANCH TO ERROR CALL.
7466 042124 012767 000003 137126 130$:   MOV   #3,$TMP13       ;PUT A '3' IN 'REGISTER CHANGED' LOCATION.
7467 042132 012767 042356 137104         MOV   #8$,$TMP5       ;MOVE RETURN TO $TMP5.
7468 042140 026703 000322                CMP   STORE+44,R3     ;SEE IF R3 WAS CHANGED.
7469 042144 001504                       BEQ   8$              ;BRANCH TO RETURN IF OK.
7470 042146 010367 137074                MOV   R3,$TMP6        ;MOVE ACTUAL R3 TO LOCATION FOR ERROR PRINTING.
7471 042152 016767 000310 137060         MOV   STORE+44,$TMP3  ;MOVE EXPECTED TO LOCATION FOR ERROR PRINTING.
7472 042160 016703 000302                MOV   STORE+44,R3     ;RESTORE R3.
7473 042164 105767 137102        7$:     TSTB  $TMP20          ;TEST TO SEE WHICH ERROR IS BEING PRINTED.
7474 042170 001027                       BNE   72$             ;BRANCH AROUND ERROR DATA SETUPS IF SR1 OR REG  ERROR.
7475 042172 012067 137042        71$:    MOV   (R0)+,$TMP3     ;MOVE 1ST WORD OF ACTUAL AC DATA TO $TMP3.
7476 042176 012067 137040                MOV   (R0)+,$TMP4     ;MOVE 2ND WORD OF ACTUAL AC DATA TO $TMP4.
7477 042202 012067 137040                MOV   (R0)+,$TMP6     ;MOVE 3RD WORD OF ACTUAL AC DATA TO $TMP6.
7478 042206 012067 137036                MOV   (R0)+,$TMP7     ;MOVE 4TH WORD OF ACTUAL AC DATA TO $TMP7.
7479 042212 016700 000262                MOV   STORE+56,R0     ;RESTORE R0 TO WHAT IT HAD AT BEGINNING OF TRAP.
7480 042216 016767 000246 137026         MOV   STORE+46,$TMP10 ;MOVE 1ST WORD OF EXPECTED AC DATA TO $TMP10.
7481 042224 016767 000242 137022         MOV   STORE+50,$TMP11 ;MOVE 2ND WORD OF EXPECTED AC DATA TO $TMP11.
7482 042232 016767 000236 137016         MOV   STORE+52,$TMP12 ;MOVE 3RD WORD OF EXPECTED AC DATA TO $TMP12.
7483 042240 016767 000232 137026         MOV   STORE+54,$TMP21 ;MOVE 4TH WORD OF EXPECTED AC DATA TO $TMP21.
7484 042246 000404                       BR    73$             ;BRANCH TO ERROR CALL NEST.
7485 042250 122767 000002 137014 72$:    CMPB  #2,$TMP20       ;TEST TO SEE IF AC LOAD ERROR.
7486 042256 001745                       BEQ   71$             ;BRANCH TO DATA PREPARE ROUTINE IF SO.
7487 042260 012667 137002        73$:    MOV   (SP)+,$TMP16    ;SAVE 1ST CONTENTS OF STACK AND POP IT ONCE.
7488 042264 012667 137000                MOV   (SP)+,$TMP17    ;SAVE 2ND CONTENTS OF STACK AND POP IT AGAIN.
7489 042270 122767 000002 136774         CMPB  #2,$TMP20       ;TEST TO SEE IF AC LOAD ERROR.
7490 042276 001002                       BNE   735$            ;BRANCH TO NEXT CHECK IF NOT.
7491 042300 104370                       ERROR +370            ;AC LOAD ERROR.
7492 042302 000417                       BR    77$             ;BRANCH TO STACK RESTORE.
7493 042304 122767 177776 136760 735$:   CMPB  #-2,$TMP20      ;SEE IF MODE 0 ERROR REPORT.
7494 042312 001002                       BNE   74$             ;BRANCH TO NEXT TEST IF NOT.
7495 042314 104365                       ERROR +365            ;MODE 0 TRAP ERROR.
7496 042316 000411                       BR    77$             ;BRANCH TO STACK RESTORE.
7497 042320 105767 136746        74$:    TSTB  $TMP20          ;TEST ERROR CALL FLAG.
7498 042324 100002                       BPL   75$             ;BRANCH TO 2ND ERROR IF NOT NEGATIVE.
```

```
7499 042326  104363                        ERROR   +363            ;ERROR FOR DIFFERENCE APPEARING BETWEEN SR1 & CALC'D.
7500 042330  000404                        BR      77$             ;BRANCH TO STACK RESTORE.
7501 042332  003002                75$:    BGT     76$             ;BRANCH TO 3RD ERROR IF $TMP20 EQUALS 1.
7502 042334  104366                        ERROR   +366            ;FPP ACCUMULATOR WAS CHANGED IN ABORT ERROR.
7503 042336  000401                        BR      77$             ;BRANCH TO STACK RESTORE.
7504 042340  104367                76$:    ERROR   +367            ;GENERAL REGISTER WAS CHANGED IN ABORT ERROR.
7505 042342  016746  136722        77$:    MOV     $TMP17,-(SP)    ;PUSH 2ND CONTENTS BACK ON THE STACK.
7506 042346  016746  136714                MOV     $TMP16,-(SP)    ;PUSH 1ST CONTENTS BACK ON THE STACK.
7507 042352  000177  136666                JMP     @$TMP5          ;JUMP TO CONTINUE CHECKING.
7508 042356  022776  000004  000000  8$:   CMP     #4,@0(SP)       ;SEE IF INSTRUCTION IS THE IOT.
7509 042364  001403                        BEQ     9$              ;BRANCH IF THE IOT HAS BEEN FOUND.
7510 042366  062716  000002                ADD     #2,(SP)         ;CORRECT PC RETURN.
7511 042372  000771                        BR      8$              ;BRANCH BACK FOR ANOTHER TRY.
7512 042374  062716  000002          9$:   ADD     #2,(SP)         ;CORRECT PC RETURN TO POINT AFTER IOT FOUND.
7513 042400  005267  135166                INC     MMR0            ;TURN ON MEMORY MANAGEMENT, AND
7514 042404  000002                        RTI                     ;RETURN FROM INTERRUPT.
7515
7516 042406                        NODAT:  .BLKW   6               ;LOCATION IN NON-RES. D-SPACE USED TO FORCE A TRAP.
7517                               ;THE 'STORE' LOCATION BELOW IS PARTITIONED TO RESERVE *4* WORDS FOR EACH FP
7518                               ;ACCUMULATOR, EVEN THOUGH ONLY 2 ARE REQUIRED FOR STORING A FLOATING NUMBER.
7519                               ;THIS IS BECAUSE *IF* THE FPS IS CHANGED BY A PROBLEM IN THE FPP, SO THAT A
7520                               ;*DOUBLE* IS STORED, *4* WORDS RESERVED WILL GUARANTEE THAT THE NEXT DATA BLOCK
7521                               ;WILL NOT BE DISTURBED.  PARTITIONING IS AS FOLLOWS:
7522                               ;       WORD(S)         USE
7523                               ;       -------         -------------
7524                               ;       1 - 4           STORE AC0
7525                               ;       5 -10           STORE AC1
7526                               ;       11-14           STORE AC2
7527                               ;       15-20           STORE AC3
7528                               ;        21             STORE R0
7529                               ;        22             STORE R2
7530                               ;        23             STORE R3
7531                               ;       24-27           STORE ACTUAL AC
7532                               ;        30             STORE ACTUAL R0 SO R0 CAN BE USED IN AC ERROR CALLS
7533 042422                        STORE:  .BLKW   30      ;STORAGE LOCATIONS FOR THE FLOATING ACCUMULATORS & DATA.
7534
7535 042502  005067  135064        ENDTES: CLR     MMR0            ;TURN OFF MEMORY MANAGEMENT.
7536 042506  016767  136550  135534        MOV     $TMP14,MMVECT   ;RESTORE MMVECT CONTENTS.
7537 042514  016767  136544  135276        MOV     $TMP15,IOTRAP   ;RESTORE IOTRAP CONTENTS.
7538 042522                        DIDONE:
     042522  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
7539
7540 042524                        TST100:
7541
7542
7543
7544                               .SBTTL  END OF PASS ROUTINE
                                   ;;*****************************************************************
                                   ;*INCREMENT THE PASS NUMBER ($PASS)
                                   ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                   ;*IF SW12=1 INHIBIT TRACE TRAP
                                   ;*IF THERES A MONITOR GO TO IT
                                   ;*IF THERE ISN'T JUMP TO LOOP
```

```
042524                          $EOP:
042524  000004                          SCOPE
042526  005067  136350                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
042532  005067  136544                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
042536  005267  136562                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
042542  042767  100000  136554        BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
042550  005327                          DEC     (PC)+           ;;LOOP?
042552  000001                  $EOPCT: .WORD   1
042554  003074                          BGT     $DOAGN          ;;YES
042556  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
042560  000001                  $ENDCT: .WORD   1
042562  042552                          $EOPCT
042564  104401  042572                  TYPE    ,65$            ;;TYPE ASCIZ STRING
042570  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
042610                          64$:
042610  016746  136510                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER IN OCTAL
042614  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
042616     006                          .BYTE   6               ;;TYPE 6 DIGITS
042617     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
042620  104401  042626                  TYPE    ,67$            ;;TYPE ASCIZ STRING
042624  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
042670                          66$:
042670  016746  136216                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS IN OCTAL
042674  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
042676     006                          .BYTE   6               ;;TYPE 6 DIGITS
042677     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
042700  104401  001313                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
042704  005067  136202                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
042710  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
042714  001414                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
042716  005046                          CLR     -(SP)           ;;INSURE THE 'T' BIT IS CLEAR
042720  012746  042726                  MOV     #$CLR.T,-(SP)   ;;SETUP FOR AN RTI OR RTT
042724  000426                          BR      $RTRN           ;;GO DO AN RTI OR RTT TO LOAD THE PSW
                                                                ;;WITH A CLEARED 'T' BIT
042726                          $CLR.T:
042726  013700  000042                  MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
042732  001405                          BEQ     $DOAGN          ;;RETURN ADDRESS
042734  000005                          RESET                   ;;CLEAR THE WORLD
042736  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
042740  000240                          NOP                     ;;SAVE ROOM
042742  000240                          NOP                     ;;FOR
042744  000240                          NOP                     ;;ACT11
042746                          $DOAGN:
042746  104400                          TRAP                    ;;PUSH OLD PSW AND PC ON STACK
042750  042716  000020                  BIC     #20,(SP)        ;;CLEAR THE 'T' BIT
042754  032777  010000  136156          BIT     #BIT12,@SWR     ;;RUN WITH TRACE TRAP?
042762  001005                          BNE     1$              ;;BR IF NO
042764  005167  000020                  COM     $TBIT           ;;IS IT TIME FOR TRACE TRAP
042770  100402                          BMI     1$              ;;BR IF NO
042772  052716  000020                  BIS     #20,(SP)        ;;SET TRACE TRAP
042776  012746  043004          1$:     MOV     #$LOOP,-(SP)    ;;JUMP TO START OF TEST
043002  000002                  $RTRN:  RTI                     ;;RETURN--THIS IS CHANGED TO
                                                                ;;AN 'RTT' IF 'RTT' IS A LEGAL
```

```
                                                        ;;INSTRUCTION
043004                          $LOOP:
043004  000137                          JMP     @(PC)+          ;;RETURN
043006  006570                  $RTNAD: .WORD   LOOP
043010  000000                  $TBIT:  .WORD   0               ;;''T'' BIT STATE INDICATOR
043012  377     377     000     $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                        .EVEN
7545
7546                            .SBTTL  SCOPE HANDLER ROUTINE
                                ;;**************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*SW09=1         LOOP ON ERROR
                                ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT
043016                          $SCOPE:
043016  104406                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
043020  032777  040000  136112  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
043026  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
043030  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A ''NOP'' (NOP=240)
043032  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
043036  012737  043056  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
043044  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
043050  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
043054  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
043056  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
043060  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
043064  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
043066                          6$:;#####END OF CODE FOR THE XOR TESTER#####
043066  032777  000400  136044          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
043074  001404                          BEQ     2$              ;;BR IF NO
043076  127767  136036  135776          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
043104  001465                          BEQ     $OVER           ;;BR IF YES
043106  105767  135771          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
043112  001421                          BEQ     3$              ;;BR IF NO
043114  126767  135775  135761          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
043122  101015                          BHI     3$              ;;BR IF NO
043124  032777  001000  136006          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
043132  001404                          BEQ     4$              ;;BR IF NO
043134  016767  135750  135744  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
043142  000446                          BR      $OVER
043144  105067  135733          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
043150  005067  136126                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
043154  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
043156  032777  004000  135754  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
043164  001011                          BNE     1$              ;;BR IF YES
043166  005767  136132                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
043172  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
043174  005267  135704                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
043200  026767  136076  135676          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
```

```
043206  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
043210  012767  000001  135666  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
043216  016767  000052  136056          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
043224  105267  135652          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
043230  116767  135646  136064          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
043236  011667  135644                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
043242  011667  135642                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
043246  005067  136032                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
043252  112767  000001  135635          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
043260  016777  135616  135654  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
043266  016716  135614                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
043272  000002                          RTI                     ;;FIXES PS
043274  000001                  $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
7547
7548                                    .SBTTL  ERROR HANDLER ROUTINE
                                        ;;**********************************************************************
                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO ERTYPE ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1         BELL ON ERROR
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
043276                          $ERROR:
043276  104406                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
043300  105267  135577          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
043304  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
043306  016777  135570  135626          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
043314  032777  002000  135616          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
043322  001402                          BEQ     1$              ;;NO - SKIP
043324  104401  001306                  TYPE    ,$BELL          ;;RING BELL
043330  005267  135556          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
043334  011667  135556                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
043340  162767  000002  135550          SUB     #2,$ERRPC
043346  117767  135544  135540          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
043354  032777  020000  135556          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
043362  001004                          BNE     20$             ;;SKIP TYPEOUTS
043364  004767  002174                  JSR     PC,ERTYPE       ;;GO TO USER ERROR ROUTINE
043370  104401  001313                  TYPE    ,$CRLF
043374                          20$:
043374  122767  000001  135734          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
043402  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
043404  116767  135504  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
043412  004767  001010                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
043416     000                  21$:     .BYTE   0
043417     000                          .BYTE   0
043420  000777                  22$:    BR      22$             ;;APT ERROR LOOP
043422  005777  135512          2$:     TST     @SWR            ;;HALT ON ERROR
043426  100002                          BPL     3$              ;;SKIP IF CONTINUE
043430  000000                          HALT                    ;;HALT ON ERROR!
043432  104406                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
043434  032777  001000  135476  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
043442  001402                          BEQ     4$              ;;BR IF NO
043444  016716  135440                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
```

```
043450  005767  135630          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
043454  001402                          BEQ     5$              ;;BR IF NONE
043456  016716  135622                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
043462                          5$:
043462  022737  042736  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
043470  001001                          BNE     6$              ;;BRANCH IF NO
043472  000000                          HALT                    ;;YES
043474                          6$:
043474  032777  001000  135436          BIT     #BIT09,@SWR
043502  001013                          BNE     ERM10
043504  011637  001162                  MOV     (SP),@#$REG0            ;SEE IF ERROR #377
043510  062737  177776  001162          ADD     #-2,@#$REG0
043516  122777  000377  135436          CMPB    #377,@$REG0
043524  001002                          BNE     ERM10
043526  062716  000002                  ADD     #2,(SP)
043532  000002                  ERM10:  RTI
7549
7550                                    .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
                                        ;;*****************************************************************
                                        ;*SAVE R0-R5
                                        ;*CALL:
                                        ;*      SAVREG
                                        ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                        ;*
                                        ;*TOP---(+16)
                                        ;* +2---(+18)
                                        ;* +4---R5
                                        ;* +6---R4
                                        ;* +8---R3
                                        ;*+10---R2
                                        ;*+12---R1
                                        ;*+14---R0
043534                                  $SAVREG:
043534  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043536  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043540  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
043542  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
043544  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
043546  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
043550  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
043554  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
043560  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
043564  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
043570  000002                          RTI
                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
043572                                  $RESREG:
043572  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
043576  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
043602  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
043606  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
043612  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
043614  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
043616  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
043620  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
043622  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
043624  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
043626  000002                          RTI
 7551
 7552                           .SBTTL  TYPE ROUTINE
                                ;;******************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*
043630  105767  135323          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
043634  100002                          BPL     1$              ;;BR IF YES
043636  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
043640  000430                          BR      3$              ;;LEAVE
043642  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
043644  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
043650  122767  000001  135460          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
043656  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
043660  132767  000100  135451          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
043666  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
043670  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
043674  004767  000516                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
043700  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
043702  132767  000040  135427  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
043710  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
043712  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
043714  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
043716  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
043720  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
043722  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
043726  000002                          RTI                     ;;RETURN
043730  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
043734  001430                          BEQ     8$
043736  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
043742  001006                          BNE     5$
043744  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
043746  104401                          TYPE                    ;;TYPE A CR AND LF
043750  001313                          $CRLF
043752  105067  000200                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
043756  000755                          BR      2$              ;;GET NEXT CHARACTER
043760  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
043764  126726  135166          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
043770  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
043772  016746  135156                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
043776  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
044002  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
044004  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
044010  105367  000142                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
044014  000770                          BR      7$              ;;LOOP
                        ;HORIZONTAL TAB PROCESSOR
044016  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
044022  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
044026  132767  000007  000122          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
044034  001372                          BNE     9$              ;;TAB STOP
044036  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
044040  000724                          BR      2$              ;;GET NEXT CHARACTER
044042  105777  135102          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
044046  100375                          BPL     $TYPEC
044050  116677  000002  135074          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
044056  105777  135062                  TSTB    @$TKS           ;;SEE IF KEYBOARD IS TALKING.
044062  100021                          BPL     2$              ;;BRANCH IF IT ISN'T.
044064  017746  135056                  MOV     @$TKB,-(SP)     ;;PUSH CHARACTER ONTO STACK.
044070  042716  177600                  BIC     #177600,(SP)    ;;BIT CLEAR TOP BYTE AND PARITY BIT.
044074  022726  000023                  CMP     #23,(SP)+       ;;SEE IF THIS IS A ^S.
044100  001012                          BNE     2$              ;;BRANCH TO CONTINUE IF IT ISN'T.
044102  105777  135036          3$:     TSTB    @$TKS           ;;WAIT FOR ANOTHER INPUT.
044106  100375                          BPL     3$              ;;BRANCH BACK IF NOT READY.
044110  017746  135032                  MOV     @$TKB,-(SP)     ;;PUSH NEXT CHARACTER ON STACK.
044114  042716  177600                  BIC     #177600,(SP)    ;;BIT CLEAR TOP BYTE AND PARITY BIT.
044120  022726  000021                  CMP     #21,(SP)+       ;;SEE IF THIS IS A ^Q.
044124  001366                          BNE     3$              ;;BRANCH BACK FOR MORE WAIT IF NOT.
044126  122766  000015  000002  2$:     CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
044134  001003                          BNE     1$              ;;BRANCH IF NO
044136  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
044142  000406                          BR      $TYPEX          ;;EXIT
044144  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
044152  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
044154  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
044156  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
044160  000207                  $TYPEX: RTS     PC
7553
7554                            .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
                                ;;**********************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
044162  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
```

```
044166  116667  000001  000211          MOVB    1(SP),$OFILL      ;;LOAD ZERO FILL SWITCH
044174  112667  000207                  MOVB    (SP)+,$OMODE+1    ;;NUMBER OF DIGITS TO TYPE
044200  062716  000002                  ADD     #2,(SP)           ;;ADJUST RETURN ADDRESS
044204  000406                          BR      $TYPON
044206  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL         ;;SET THE ZERO FILL SWITCH
044214  112767  000006  000165          MOVB    #6,$OMODE+1       ;;SET FOR SIX(6) DIGITS
044222  112767  000005  000154  $TYPON: MOVB    #5,$OCNT          ;;SET THE ITERATION COUNT
044230  010346                          MOV     R3,-(SP)          ;;SAVE R3
044232  010446                          MOV     R4,-(SP)          ;;SAVE R4
044234  010546                          MOV     R5,-(SP)          ;;SAVE R5
044236  116704  000145                  MOVB    $OMODE+1,R4       ;;GET THE NUMBER OF DIGITS TO TYPE
044242  005404                          NEG     R4
044244  062704  000006                  ADD     #6,R4             ;;SUBTRACT IT FOR MAX. ALLOWED
044250  110467  000132                  MOVB    R4,$OMODE         ;;SAVE IT FOR USE
044254  116704  000125                  MOVB    $OFILL,R4         ;;GET THE ZERO FILL SWITCH
044260  016605  000012                  MOV     12(SP),R5         ;;PICKUP THE INPUT NUMBER
044264  005003                          CLR     R3                ;;CLEAR THE OUTPUT WORD
044266  006105                  1$:     ROL     R5                ;;ROTATE MSB INTO "C"
044270  000404                          BR      3$                ;;GO DO MSB
044272  006105                  2$:     ROL     R5                ;;FORM THIS DIGIT
044274  006105                          ROL     R5
044276  006105                          ROL     R5
044300  010503                          MOV     R5,R3
044302  006103                  3$:     ROL     R3                ;;GET LSB OF THIS DIGIT
044304  105367  000076                  DECB    $OMODE            ;;TYPE THIS DIGIT?
044310  100016                          BPL     7$                ;;BR IF NO
044312  042703  177770                  BIC     #177770,R3        ;;GET RID OF JUNK
044316  001002                          BNE     4$                ;;TEST FOR 0
044320  005704                          TST     R4                ;;SUPPRESS THIS 0?
044322  001403                          BEQ     5$                ;;BR IF YES
044324  005204                  4$:     INC     R4                ;;DON'T SUPPRESS ANYMORE 0'S
044326  052703  000060                  BIS     #'0,R3            ;;MAKE THIS DIGIT ASCII
044332  052703  000040          5$:     BIS     #' ,R3            ;;MAKE ASCII IF NOT ALREADY
044336  110367  000040                  MOVB    R3,8$             ;;SAVE FOR TYPING
044342  104401  044402                  TYPE    ,8$               ;;GO TYPE THIS DIGIT
044346  105367  000032          7$:     DECB    $OCNT             ;;COUNT BY 1
044352  003347                          BGT     2$                ;;BR IF MORE TO DO
044354  002402                          BLT     6$                ;;BR IF DONE
044356  005204                          INC     R4                ;;INSURE LAST DIGIT ISN'T A BLANK
044360  000744                          BR      2$                ;;GO DO THE LAST DIGIT
044362  012605                  6$:     MOV     (SP)+,R5          ;;RESTORE R5
044364  012604                          MOV     (SP)+,R4          ;;RESTORE R4
044366  012603                          MOV     (SP)+,R3          ;;RESTORE R3
044370  016666  000002  000004          MOV     2(SP),4(SP)       ;;SET THE STACK FOR RETURNING
044376  012616                          MOV     (SP)+,(SP)
044400  000002                          RTI                       ;;RETURN
044402     000                  8$:     .BYTE   0                 ;;STORAGE FOR ASCII DIGIT
044403     000                          .BYTE   0                 ;;TERMINATOR FOR TYPE ROUTINE
044404     000          $OCNT:  .BYTE   0                 ;;OCTAL DIGIT COUNTER
044405     000          $OFILL: .BYTE   0                 ;;ZERO FILL SWITCH
044406  000000          $OMODE: .WORD   0                 ;;NUMBER OF DIGITS TO TYPE
7555
7556                            .SBTTL  APT COMMUNICATIONS ROUTINE
                                ;;**************************************************************
044410  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG          ;;TO REPORT FATAL ERROR
044416  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG          ;;TO TYPE A MESSAGE
044424  000403                          BR      $ATYC
```

```
044426  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
044434                          $ATYC:
044434  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044436  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044440  105767  000206          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
044444  001450                  BEQ     5$              ;;IF NOT:  BR
044446  122767  000001  134662  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
044454  001031                  BNE     3$              ;;IF NOT:  BR
044456  132767  000100  134653  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
044464  001425                  BEQ     3$              ;;IF NOT:  BR
044466  017600  000004          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
044472  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
044500  005767  134612  1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
044504  001375                  BNE     1$              ;;IF NOT:  WAIT
044506  010067  134620          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
044512  105720          2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
044514  001376                  BNE     2$
044516  166700  134610          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
044522  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
044524  010067  134604          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
044530  012767  000004  134560  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
044536  000413                  BR      5$
044540  017667  000004  000016  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
044546  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
044554  016746  133216          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
044560  004767  177044          JSR     PC,$TYPE        ;;CALL TYPE MACRO
044564  000000          4$:     .WORD   0
044566                  5$:
044566  105767  000062  10$:     TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
044572  001416                  BEQ     12$             ;;IF NOT:  BR
044574  005767  134536          TST     $ENV            ;;RUNNING UNDER APT?
044600  001413                  BEQ     12$             ;;IF NOT:  BR
044602  005767  134510  11$:     TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
044606  001375                  BNE     11$             ;;IF NOT:  WAIT
044610  017667  000004  134502  MOV     @4(SP),$FATAL   ;;GET ERROR #
044616  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
044624  005267  134466          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
044630  105067  000020  12$:     CLRB    $FFLG           ;;CLEAR FATAL FLAG
044634  105067  000013          CLRB    $LFLG           ;;CLEAR LOG FLAG
044640  105067  000006          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
044644  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
044646  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
044650  000207                  RTS     PC              ;;RETURN
044652     000  $MFLG:  .BYTE   0               ;;MESSG. FLAG
044653     000  $LFLG:  .BYTE   0               ;;LOG FLAG
044654     000  $FFLG:  .BYTE   0               ;;FATAL FLAG
                        .EVEN
        000200          APTSIZE=200
        000001          APTENV=001
        000100          APTSPOOL=100
        000040          APTCSUP=040
7557
7558                    .SBTTL  TTY INPUT ROUTINE
                        ;;*****************************************************************
                        .ENABL  LSB
                        ;;*****************************************************************
                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
```

```
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
044656  022767  000176  134254  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
044664  001074                          BNE     15$             ;;BRANCH IF NO
044666  105777  134252                  TSTB    @$TKS           ;;CHAR THERE?
044672  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
044674  117746  134246                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
044700  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
044704  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
044710  001062                          BNE     15$             ;;NO, RETURN TO USER
044712  126727  134216  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
044720  001456                          BEQ     15$             ;;BRANCH IF YES
044722  104401  045265                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
044726  104401  045272          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
044732  016746  133240                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
044736  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
044740  104401  045303                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
044744  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
044746  005046                          CLR     -(SP)           ;;THE NEW SWR
044750  105777  134170          7$:     TSTB    @$TKS           ;;CHAR THERE?
044754  100375                          BPL     7$              ;;IF NOT TRY AGAIN
044756  117746  134164                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
044762  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
044766  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
044772  001005                          BNE     10$             ;;BRANCH IF NOT
044774  104401  045260                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
045000  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
045004  000757                          BR      19$             ;;LET'S TRY IT AGAIN
045006  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
045012  001022                          BNE     16$             ;;BRANCH IF NO
045014  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
045020  001403                          BEQ     11$             ;;BRANCH IF YES
045022  016677  000002  134110          MOV     2(SP),@SWR      ;;SAVE NEW SWR
045030  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
045034  104401  001313          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
045040  126727  134071  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
045046  001003                          BNE     15$             ;;BRANCH IF NOT
045050  012777  000100  134066          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
045056  000002                  15$:    RTI                     ;;RETURN
045060  004767  176756          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
045064  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
045070  002420                          BLT     18$             ;;BRANCH IF YES
045072  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
045076  003015                          BGT     18$             ;;BRANCH IF YES
045100  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
045104  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
045110  001403                          BEQ     17$             ;;BRANCH IF YES
045112  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
045114  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
045116  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
045120  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
045124  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
045130  000707                          BR      7$              ;;GET THE NEXT ONE
045132  104401  001312          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
045136  000720                          BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB
```

```
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
045140  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
045142  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
045150  105777  133770          1$:     TSTB    @$TKS           ;;WAIT FOR
045154  100375                          BPL     1$              ;;A CHARACTER
045156  117766  133764  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
045164  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
045172  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
045200  001013                          BNE     3$              ;;BRANCH IF NO
045202  105777  133736          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
045206  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
045210  117746  133732                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
045214  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
045220  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
045224  001366                          BNE     2$              ;;IF NOT DISCARD IT
045226  000750                          BR      1$              ;;YES, RESUME
045230  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
045236  002407                          BLT     4$              ;;BRANCH IF YES
045240  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
045246  003003                          BGT     4$              ;;BRANCH IF YES
045250  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
045256  000002                  4$:     RTI                     ;;GO BACK TO USER
045260     136     125     015  $CNTLU: .ASCIZ   /^U/<15><12>    ;;CONTROL 'U'
045263     012     000
045265     136     107     015  $CNTLG: .ASCIZ   /^G/<15><12>    ;;CONTROL 'G'
045270     012     000
045272     015     012     123  $MSWR:  .ASCIZ   <15><12>/SWR = /
045275     127     122     040
045300     075     040     000
045303     040     040     116  $MNEW:  .ASCIZ   /  NEW = /
045306     105     127     040
045311     075     040     000
7559
7560                            .SBTTL  TRAP DECODER
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE ''TRAP'' INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.
045314  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
045316  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
045322  005740                          TST     -(R0)           ;;BACKUP BY 2
045324  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
045326  006300                          ASL     R0              ;;POSITION FOR INDEXING
045330  016000  045350                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
045334  000200                          RTS     R0              ;;GO TO ROUTINE
                                ;;THIS IS USE TO HANDLE THE ''GETPRI'' MACRO
045336  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
045340  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
045346  000002                          RTI                     ;;RESTORE THE PSW
                                .SBTTL  TRAP TABLE
```

```
                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE "TRAP" INSTRUCTION.
                                        ;       ROUTINE
                                        ;       -------
        045350  045336                  $TRPAD: .WORD   $TRAP2
        045352  043630                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
        045354  044206                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        045356  044162                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
        045360  044222                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
        045362  044726                          $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING
        045364  044656                          $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
        045366  045140                          $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
        045370  043534                          $SAVREG ;;CALL=SAVREG   TRAP+10(104410) SAVE R0-R5 ROUTINE
        045372  043572                          $RESREG ;;CALL=RESREG   TRAP+11(104411) RESTORE R0-R5 ROUTINE
7561    045374  046316                          .RSET   ;;CALL=RSETUP   TRAP+12(104412) ROUTINE TO INITIALIZE AT END OF EACH TEST
7562    045376  046310                          .LPER   ;;CALL=LPERR    TRAP+13(104413) ROUTINE TO SET UP LOOP ON ERROR ADDRESS
7563            000030                  $TERM=.-$TRPAD
7564
7565                                    .SBTTL  POWER DOWN AND UP ROUTINES
                                        ;;**********************************************************************
                                        ;POWER DOWN ROUTINE
        045400  012737  045556  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
        045406  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
        045414  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
        045416  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
        045420  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
        045422  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
        045424  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
        045426  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
        045430  017746  133504                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
        045434  010667  000122                  MOV     SP,$SAVR6       ;;SAVE SP
        045440  012737  045452  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
        045446  000000                          HALT
        045450  000776                          BR      .-2             ;;HANG UP
                                        ;;**********************************************************************
                                        ;POWER UP ROUTINE
        045452  012737  045556  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
        045460  016706  000076                  MOV     $SAVR6,SP       ;;GET SP
        045464  005067  000072                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
        045470  005267  000066          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
        045474  001375                          BNE     1$              ;;OF  WORD
        045476  012677  133436                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
        045502  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
        045504  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
        045506  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
        045510  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        045512  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        045514  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
        045516  012737  045400  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
        045524  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
        045532  104401                          TYPE                    ;;REPORT THE POWER FAILURE
        045534  046366          $PWRMG: .WORD   POWERM                  ;;POWER FAIL MESSAGE POINTER
        045536  012716                          MOV     (PC)+,(SP)      ;;RESTART AT START
        045540  006106          $PWRAD: .WORD   START                   ;;RESTART ADDRESS
        045542  042766  000020  000002          BIC     #20,2(SP)       ;;CLEAR "T" BIT
        045550  005067  175234                  CLR     $TBIT           ;;CLEAR THE "T" BIT FLAG
        045554  000002                          RTI
```

```
045556  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
045560  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
045562  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
 7566
 7567
 7568                                           .SBTTL  ERROR TYPE OUT ROUTINE
 7569                                   ;;*************************************************************************
                                        ;;*************************************************************************
 7570                                   ;*THIS ROUTINE IS CALLED TO TYPE AN ERROR MESSAGE WHICH IS INCLUDED
 7571                                   ;*IN THE ERROR MESSAGE DATA TABLE. IT IS CALLED BY THE $ERROR ROUTINE
 7572                                   ;*OR BY FIRST SETTING $ITEMB EQUAL TO THE ERROR TABLE ITEM TO BE PRINTED
 7573                                   ;*OUT AND THEN ECECUTING A:
 7574                                   ;*              JSR     PC,ERTYPE
 7575                                   ;*
 7576 045564  104401            ERTYPE: TYPE                                    ;TYPE A CRLF
 7577 045566  001313                    .WORD   $CRLF
 7578 045570  113737  001102  001232    MOVB    @#$TSTNM,@#$TMP0
 7579 045576  042737  177400  001232    BIC     #177400,@#$TMP0
 7580 045604  013737  001116  001234    MOV     @#$ERRPC,@#$TMP1        ;GET PC OF CALL
 7581 045612  010046                    MOV     R0,-(SP)                ;SAVE R0
 7582
 7583 045614  113700  001114            MOVB    @#$ITEMB,R0             ;GET THE ITEM NUMBER.
 7584 045620  042700  177400            BIC     #177400,R0
 7585 045624  001005                    BNE     1$
 7586
 7587 045626  013746  001116            MOV     @#$ERRPC,-(SP)          ;IF ZERO THEN JUST
 7588 045632  104402                    TYPOC                           ;PRINT THE PC
 7589 045634  000137  046210            JMP     @#ERT5
 7590
 7591 045640  022700  000377    1$:     CMP     #377,R0
 7592 045644  001005                    BNE     20$
 7593 045646  016600  000004            MOV     4(SP),R0
 7594 045652  011000                    MOV     (R0),R0
 7595 045654  062700  000400            ADD     #400,R0
 7596 045660  005300            20$:    DEC     R0                      ;OTHERWISE MAKE R0 AN
 7597 045662  006300                    ASL     R0                      ;INDEX FOR THE TABLE.
 7598 045664  006300                    ASL     R0
 7599 045666  006300                    ASL     R0
 7600 045670  062700  001442            ADD     #$ERRTB,R0
 7601
 7602 045674  012037  045704            MOV     (R0)+,@#2$              ;PICK UP THE ADDRESS
 7603 045700  001404                    BEQ     3$                      ;OF THE EM, ERROR MESSAGE
 7604 045702  104401                    TYPE
 7605 045704  000000            2$:     .WORD   0
 7606 045706  104401                    TYPE
 7607 045710  001313                    .WORD   $CRLF
 7608
 7609 045712  012037  045722    3$:     MOV     (R0)+,@#4$              ;GET THE DH,DATA HEADER
 7610 045716  001404                    BEQ     5$
 7611 045720  104401                    TYPE
 7612 045722  000000            4$:     .WORD   0
 7613 045724  104401                    TYPE
 7614 045726  001313                    .WORD   $CRLF
 7615
 7616 045730  010146            5$:     MOV     R1,-(SP)                ;SAVE R1,R2 AND R3
 7617 045732  010246                    MOV     R2,-(SP)
 7618 045734  010346                    MOV     R3,-(SP)
```

```
 7619
 7620 045736  012001                    MOV     (R0)+,R1        ;GET THE ADDRESS OF THE
 7621                                                            ;DATA TABLE.
 7622 045740  001516                    BEQ     ERT4            ;RETURN IF NO DATA.
 7623
 7624 045742  011000                    MOV     (R0),R0         ;GET A POINTER TO THE DATA
 7625                                                            ;FORMAT TABLE.
 7626 045744  105710            ERT1:   TSTB    (R0)            ;FORMAT ZERO?
```

```
 7628 045746  001003                    BNE     7$
 7629
 7630 045750  013146                    MOV     @(R1)+,-(SP)            ;FORMAT ZERO SO TYPE
 7631 045752  104402                    TYPOC                           ;AN OCTAL NUMBER.
 7632 045754  000502                    BR      ERT2
 7633
 7634 045756                    7$:
 7635 045756  122710  000002    8$:     CMPB    #2,(R0)                 ;FORMAT TWO?
 7636 045762  001010                    BNE     9$
 7637
 7638 045764  013102                    MOV     @(R1)+,R2               ;FORMAT TWO SO TYPE TWO
 7639 045766  012246                    MOV     (R2)+,-(SP)             ;OCTAL NUMBERS.
 7640 045770  104402                    TYPOC
 7641 045772  104401                    TYPE
 7642 045774  046432                    .WORD   SPACE
 7643 045776  011246                    MOV     (R2),-(SP)
 7644 046000  104402                    TYPOC
 7645 046002  000467                    BR      ERT2
 7646
 7647 046004  122710  000003    9$:     CMPB    #3,(R0)                 ;FORMAT THREE?
 7648 046010  001020                    BNE     10$
 7649
 7650 046012  013102                    MOV     @(R1)+,R2               ;FORMAT THREE SO TYPE
 7651 046014  012246                    MOV     (R2)+,-(SP)             ;FOUR OCTAL NUMBERS.
 7652 046016  104402                    TYPOC
 7653 046020  104401                    TYPE
 7654 046022  046432                    .WORD   SPACE
 7655 046024  012246                    MOV     (R2)+,-(SP)
 7656 046026  104402                    TYPOC
 7657 046030  104401                    TYPE
 7658 046032  046432                    .WORD   SPACE
 7659 046034  012246                    MOV     (R2)+,-(SP)
 7660 046036  104402                    TYPOC
 7661 046040  104401                    TYPE
 7662 046042  046432                    .WORD   SPACE
 7663 046044  011246                    MOV     (R2),-(SP)
 7664 046046  104402                    TYPOC
 7665 046050  000444                    BR      ERT2
 7666
 7667 046052  122710  000004   10$:     CMPB    #4,(R0)                 ;FORMAT FOUR?
 7668 046056  001004                    BNE     11$
 7669
 7670 046060  013146                    MOV     @(R1)+,-(SP)            ;FORMAR FOUR SO TYPE
 7671 046062  104403                    TYPOS                           ;AN OCTAL NUMBER
 7672 046064     016                    .BYTE   16                      ;SUPPRESSING LEADING ZEROES.
 7673 046065     000                    .BYTE   0
 7674 046066  000435                    BR      ERT2
 7675
 7676 046070  122710  000005   11$:     CMPB    #5,(R0)                 ;FORMAT FIVE?
 7677 046074  001005                    BNE     13$
 7678
 7679 046076  012137  046104            MOV     (R1)+,@#12$             ;FORMAT FIVE SO TYPE AN
 7680 046102  104401                    TYPE                            ;ASCIZ STRING.
 7681 046104  000000   12$:     .WORD   0
 7682 046106  000427                    BR      ERT3
 7683
 7684 046110  122710  000011   13$:     CMPB    #11,(R0)                ;FORMAT ELEVEN?
```

```
7685 046114  001005                    BNE     15$
7686
7687 046116  013137  046124            MOV     @(R1)+,@#14$            ;FORMAT ELEVEN SO PICK
7688 046122  104401                    TYPE                            ;A POINTER TO AN ASCIZ
7689 046124  000000              14$:  .WORD   0                       ;STRING.
7690 046126  000417                    BR      ERT3
7691
7692 046130  122710  000012      15$:  CMPB    #12,(R0)                ;FORMAT TWELVE?
7693 046134  001011                    BNE     17$
7694
7695 046136  013102                    MOV     @(R1)+,R2               ;FORMAT TWELVE SO TYPE
7696 046140  012703  000006            MOV     #6,R3                   ;TYPE SIX OCTAL NUMBERS
7697 046144  012246              16$:  MOV     (R2)+,-(SP)
7698 046146  104402                    TYPOC
7699 046150  104401                    TYPE
7700 046152  046432                    .WORD   SPACE
7701 046154  077305                    SOB     R3,16$
7702 046156  000401                    BR      ERT2
7703
7704 046160  000000              17$:  HALT                            ;UNDEFINED FORMAT FOR DATA?????
7705
7706 046162  104401              ERT2: TYPE                            ;PRINT A TAB AFTER TYPING
7707 046164  046435                    .WORD   $TAB                    ;AN DATA TABLE ENTRY
7708                                                                   ;OF ALL FORMATS EXCEPT
7709                                                                   ;ASCIZ, FORMATS 5 OR 11
7710
7711 046166  005200              ERT3: INC     R0                      ;POINT TO THE NEXT FORMAT
7712 046170  005711                    TST     (R1)                    ;END OF DATA TABLE.
7713 046172  001401                    BEQ     ERT4
7714 046174  000663                    BR      ERT1
7715
7716 046176  104401              ERT4: TYPE                            ;DONE.
7717 046200  001313                    .WORD   $CRLF
7718 046202  012603                    MOV     (SP)+,R3                ;RESTORE R1,R2 AND R3
7719 046204  012602                    MOV     (SP)+,R2
7720 046206  012601                    MOV     (SP)+,R1
7721 046210  012600              ERT5: MOV     (SP)+,R0                ;RESTORE R0.
7722 046212  000207                    RTS     PC                      ;AND RETURN.
7723
7724
7725
7726
7727                                    .SBTTL  FPP SPURIOUS TRAP TO 244 HANDLER
7728                            ;;**********************************************************************
                                ;;**********************************************************************
7729                            ;*THIS ROUTINE HANDLES UNEXPECTED TRAPS TO THE FPP TRAP VECTOR AT 244.
7730                            ;*THE LAST FPP INSTRUCTION EXECUTED AND ITS ADDRESS HAS BEEN RECORDED
7731                            ;*THESE ALONG WITH THE FEC, FPS AND PC OF TRAP ARE REPORTED.
7732                            ;*
7733 046214  011637  001236  FPSPUR: MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
7734 046220  022626                    CMP     (SP)+,(SP)+             ;RESTORE SP.
7735 046222  170200                    STFPS   R0                      ;GET FPS
7736 046224  010037  001240            MOV     R0,@#$TMP3
7737 046230  170300                    STST    R0                      ;GET FEC
7738 046232  010037  001242            MOV     R0,@#$TMP4
7739 046236  104377              1$:   ERROR   +377
7740 046240  000441                    .WORD   441
```

```
7741 046242  104412                    RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                               ;SEE IF THE USER HAS EXPRESSED
                                                               ;THE DESIRE TO CHANGE THE SOFTWARE
                                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                               ;THE USER TYPED CONTROL G?).
7742 046244  000137  042524             JMP     @#$EOP
7743
7744
7745                                    .SBTTL  CPU SPURIOUS TRAP TO 4 HANDLER
7746                                ;;**********************************************************************
                                    ;;**********************************************************************
7747                                ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 4.
7748                                ;*
7749 046250  011637  001236        CPSPUR: MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
7750 046254  022626                        CMP     (SP)+,(SP)+
7751 046256  104377                1$:     ERROR   +377
7752 046260  000442                        .WORD   442
7753 046262  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                   ;SEE IF THE USER HAS EXPRESSED
                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                   ;THE USER TYPED CONTROL G?).
7754 046264  000137  042524                JMP     @#$EOP
7755
7756
7757                                    .SBTTL  CPU SPURIOUS TRAP TO 10 HANDLER
7758                                ;;**********************************************************************
                                    ;;**********************************************************************
7759                                ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 10.
7760                                ;*
7761 046270  011637  001236        CPTWO:  MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
7762 046274  022626                        CMP     (SP)+,(SP)+
7763 046276  104377                1$:     ERROR   +377
7764 046300  000443                        .WORD   443
7765 046302  104412                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                   ;SEE IF THE USER HAS EXPRESSED
                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                   ;THE USER TYPED CONTROL G?).
7766 046304  000137  042524                JMP     @#$EOP
7767
7768
7769
7770
7771
7772                                    .SBTTL  SET LOOP ON ERROR ADDRESS ROUTINE
7773                                ;;**********************************************************************
                                    ;;**********************************************************************
7774                                ;*
7775 046310  011637  001110        .LPER:  MOV     (SP),@#$LPERR
7776 046314  000002                        RTI
7777
7778                                    .SBTTL  FLAG RESET AND CONSOLE TEST ROUTINE
7779                                ;;**********************************************************************
                                    ;;**********************************************************************
7780                                ;*THIS ROUTINE WILL BE CALLED AT THE END OF EACH TEST TO
7781                                ;*RESET THE STACK, CLEAR THE FPS AND SEE IF THE USER HAS TYPED
```

```
                                        ;* CONTROL G ON THE TERMINAL. IF THE USER HAS TYPED CONTROL G AND
                                        ;*THERE IS NO PHYSICAL CONSOLE SWITCH REGISTER THEN THE CONTENTS
                                        ;*OF THE SOFTWARE SWITCH REGISTER WILL BE TYPED IN OCTAL ON THE
                                        ;*TELETYPE AND THE USER CAN MODIFY IT.
                                        ;*
046316  023727  001140  177570  .RSET:  CMP     @#SWR,#177570 ,         ;SEE IF THERE IS A PHYSICAL
                                                                        ;CONSOLE SWITCH REGISTER.
046324  001001                          BNE     1$                      ;BRANCH IF NO.
046326  104406                          CKSWR                           ;OTHERWISE TYPE THE CONTENTS
                                                                        ;OF THE PROGRAM VIRTUAL SWITCH REGISTER
                                                                        ;AND GIVE THE USER A CHANCE TO
                                                                        ;MODIFY IT.
046330  012737  046214  000244  1$:     MOV     #FPSPUR,@#FPVECT
046336  012737  046250  000004          MOV     #CPSPUR,@#ERRVECT
046344  012737  046270  000010          MOV     #CPTWO,@#10
046352  011600                          MOV     (SP),R0                 ;SAVE RETURN ADDRESS.
046354  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK POINTER.
046360  005004                          CLR     R4                      ;CLEAR THE FPS.
046362  170104                          LDFPS   R4
046364  000110                          JMP     (R0)                    ;RETURN.


                                        .NLIST  BEX

                                        ;THESE ARE SPECIAL MESSAGES:

046366      200     120     117  POWERM: .ASCIZ  <CRLF>'POWER FAILURE. PROGRAM RESTARTING.'
046432      040     040     000  SPACE:  .ASCIZ  '  '
046435      011     000          $TAB:   .ASCIZ  <TAB>

046437      107     117     124  MS1:    .ASCIZ  'GOT RESULT:'<TAB><TAB>
046455      105     130     120  MS2:    .ASCIZ  'EXPECTED RESULT:'<TAB>
046477      101     103     040  MS3:    .ASCIZ  'AC OPERAND:'<TAB><TAB>
046515      123     117     125  MS4:    .ASCIZ  'SOURCE OPERAND: '<TAB>
            046477               MS10=MS3
046537      105     130     120  MS11:   .ASCIZ  'EXPONENT OPERAND:'<TAB>
046562      114     117     101  MS20:   .ASCIZ  'LOADED:'<TAB><TAB>
046574      124     122     111  MS21:   .ASCIZ  'TRIED TO LOAD:'<TAB>

                                        ;THESE ARE ERROR MESSAGES:

046614      123     124     106  EM1:    .ASCIZ  'STF A,AC7 DID NOT TRAP. FID=0.'
046653      123     124     106  EM2:    .ASCIZ  'STF A,AC7. FPS BAD. FID=0.'
046706      123     124     106  EM3:    .ASCIZ  'STF A,AC7. FEC BAD. FID=0.'
046741                           EM4:
046741      123     124     106          .ASCIZ  \STF A,(R). R0 BAD. FDST FAILED.\
047001                           EM5:
047001      123     124     106          .ASCII  \STF A,(R) FAILED.\
047022      000                  .BYTE   0
047023                           EM6:
047023      123     124     106          .ASCII  \STF A,(R). FDST FAILED.\
047052      200     050     102          .ASCIZ  <CRLF>\(BUT FD) ST 707 WENT TO 245 INSTEAD OF 244.\
047127                           EM7:
047127      123     124     106          .ASCIZ  \STF A,(R)+. R0 BAD. FDST FAILED.\
047170                           EM10:
047170      123     124     106          .ASCII  \STF A,(R)+ FAILED.\
047212      000                  .BYTE   0
```

```
 7846 047213                                EM11:
      047213      123      124      104             .ASCIZ  \STD A,(R)+. R0 BAD. FDST FAILED.\
 7847 047254                                EM12:
      047254      123      124      104             .ASCII  \STD A,(R)+ FAILED.\
 7848 047276      000                               .BYTE   0
 7849 047277      123      124      104     EM13:   .ASCIZ  'STD A,#N TRAP TO 4 IN FDST.'
 7850          047277                       EM14=EM13
 7851 047333                                EM15:
      047333      123      124      104             .ASCII  \STD A,#N FAILED.\
 7852 047353      000                               .BYTE   0
 7853 047354      120      103      040     EM16:   .ASCIZ  'PC BAD AFTER STD A,#N.'
 7857 047403                                EM17:
      047403      123      124      104             .ASCIZ  \STD A,-(R) TRAP TO 4 IN FDST.\
 7858 047441                                EM20:
      047441      123      124      104             .ASCIZ  \STD A,-(R). R0 BAD. FDST FAILED.\
 7859 047502                                EM21:
      047502      123      124      104             .ASCII  \STD A,-(R) FAILED.\
 7860 047524      000                               .BYTE   0
 7861          047502                       EM22=EM21
 7862 047525                                EM23:
      047525      123      124      104             .ASCIZ  \STD A,@(R)+ TRAP TO 4 IN FDST.\
 7863 047564                                EM24:
      047564      123      124      104             .ASCIZ  \STD A,@(R)+. R0 BAD. FDST FAILED.\
 7864 047626                                EM25:
      047626      123      124      104             .ASCII  \STD A,@(R)+ FAILED.\
 7865 047651      000                               .BYTE   0
 7866 047652                                EM26:
      047652      123      124      104             .ASCIZ  \STD A,@-(R) TRAP TO 4 IN FDST.\
 7867 047711                                EM27:
      047711      123      124      104             .ASCIZ  \STD A,@-(R). R0 BAD. FDST FAILED.\
 7868 047753                                EM30:
      047753      123      124      104             .ASCII  \STD A,@-(R) FAILED.\
 7869 047776      000                               .BYTE   0
 7870 047777                                EM31:
      047777      123      124      104             .ASCIZ  \STD A,N(R) TRAP TO 4 IN FDST.\
 7871 050035                                EM32:
      050035      123      124      104             .ASCIZ  \STD A,N(R). R0 BAD. FDST FAILED.\
 7872 050076                                EM33:
      050076      123      124      104             .ASCII  \STD A,N(R) FAILED.\
 7873 050120      000                               .BYTE   0
 7874 050121                                EM34:
      050121      123      124      104             .ASCIZ  \STD A,@N(R) TRAP TO 4 IN FDST.\
 7875 050160                                EM35:
      050160      123      124      104             .ASCIZ  \STD A,@N(R). R0 BAD. FDST FAILED.\
 7876 050222                                EM36:
      050222      123      124      104             .ASCII  \STD A,@N(R) FAILED.\
 7877 050245      000                               .BYTE   0
 7884 050246                                EM37:
      050246      123      124      103             .ASCII  'STCFD A,(R) FAILED.'
 7885 050271      000                               .BYTE   0
 7886 050272                                EM40:
      050272      123      124      103             .ASCII  \STCFD A,(R). FPS BAD.\
 7887 050317      000                               .BYTE   0
 7888 050320                                EM41:
      050320      123      124      103             .ASCII  \STCFD A,(R). FEC BAD.\
 7889 050345      000                               .BYTE   0
 7890 050346                                EM42:
```

```
                050346  123  124  103          .ASCII  'STCFD A,(R) FAILED.'
7891            050371  200  111  116          .ASCIZ  <CRLF>'INVERT FDFL ST 767 FAILED.'
7892            050425                  EM43:
                050425  123  124  103          .ASCII  \STCFD A,(R). FPS BAD.\
7893            050452  200  050  102          .ASCIZ  <CRLF>\(BUT EZBT) ST 560 WENT TO 061 INSTEAD OF 261.\
7894            050531                  EM44:
                050531  123  124  103          .ASCII  'STCFD A,(R) FAILED.'
7895            050554  200  114  117          .ASCIZ  <CRLF>'LOW ORDER BITS OF X11 DID NOT GET 0 ST 766.'
7896            050631                  EM45:
                050631  123  124  103          .ASCII  'STCFD A,(R) FAILED.'
7897            050654  200  050  102          .ASCIZ  <CRLF>'(BUT OP1C) ST 251 FAILED.'
7898            050707                  EM46:
                050707  123  124  103          .ASCII  \STCFD A,(R). FPS BAD.\
7899            050734  200  050  102          .ASCIZ  <CRLF>\(BUT EZBT) ST 421 WENT TO 262 INSTEAD OF 062.\
7900            051013                  EM47:
                051013  123  124  103          .ASCII  'STCFD A,(R) FAILED.'
7901            051036  200  050  102          .ASCIZ  <CRLF>\(BUT FD) ST 113 WENT TO 415 INSTEAD OF 414.\
7902            051113                  EM50:
                051113  123  124  103          .ASCII  'STCFD A,(R) FAILED.'
7903            051136  040  123  111          .ASCII  ' SIGN BAD.'
7904            051150  200  050  102          .ASCIZ  <CRLF>\(BUT ENBT) ST 567 WENT TO 060 INSTEAD OF 460.\
7911            051227                  EM51:
                051227  123  124  103          .ASCII  'STCDF A,(R) FAILED.'
7912            051252  000                   .BYTE   0
7913            051253                  EM52:
                051253  123  124  104          .ASCII  \STD A,(R). FPS BAD.\
7914            051276  000                   .BYTE   0
7915            051277                  EM53:
                051277  123  124  104          .ASCII  \STD A,(R). FEC BAD.\
7916            051322  000                   .BYTE   0
7917            051323                  EM54:
                051323  123  124  103          .ASCII  'STCDF A,(R) FAILED.'
7918            051346  200  111  116          .ASCIZ  <CRLF>'INVERT FDFL ST 767 FAILED.'
7919            051402                  EM55:
                051402  123  124  103          .ASCII  'STCDF A,(R) FAILED.'
7920            051425  200  122  117          .ASCII  <CRLF>'ROUND ERROR, OR'
7921            051445  200  050  102          .ASCIZ  <CRLF>\(BUT BREAKOUT) ST 400 WENT TO 766 INSTEAD OF 767.\
7922            051530                  EM56:
                051530  123  124  104          .ASCII  \STD A,(R). FPS BAD.\
7923            051553  200  050  102          .ASCIZ  <CRLF>\(BUT EZBT) ST 421 WENT TO 062 INSTEAD OF 262.\
7924            051632                  EM57:
                051632  123  124  104          .ASCII  \STD A,(R). FPS BAD.\
7925            051655  040  106  111          .ASCII  ' FIV=0.'
7926            051664  200  050  102          .ASCIZ  <CRLF>\(BUT FIV) ST 262 WENT TO 123 INSTEAD OF 103.\
7927            051742                  EM60:
                051742  123  124  103          .ASCII  'STCDF A,(R) FAILED.'
7928            051765  040  106  111          .ASCII  ' FIV=1.'
7929            051774  200  050  102          .ASCIZ  <CRLF>\(BUT FIV) ST 262 WENT TO 103 INSTEAD OF 123.\
7930            052052                  EM61:
                052052  123  124  104          .ASCII  \STD A,(R). FPS BAD.\
7931            052075  200  050  102          .ASCIZ  <CRLF>\(BUT FLAG) ST 147 WENT TO 361 INSTEAD OF 365.\
7932            052154  123  124  103  EM62:   .ASCII  'STCFD A,AC6. FPS BAD.'
7933            052201  200  050  102          .ASCIZ  <CRLF>\(BUT FDST) ST 767 WENT TO 567 INSTEAD OF 577.\
7934            052260  123  124  103  EM63:   .ASCIZ  'STCFD A,AC6. FEC BAD.'
7935            052306                  EM64:
                052306  103  114  122          .ASCII  \CLRD (R) FAILED.\
7936            052326  200  132  105          .ASCIZ  <CRLF>'ZERO X11 AT ST 770 FAILED.'
```

```
052362                          EM65:
052362   103   114   122                .ASCII  \CLRD (R). FPS BAD.\
052404   000                            .BYTE   0
052405                          EM66:
052405   103   114   122                .ASCIZ  \CLRD (R). R0 BAD. FDST FAILED.\
052444                          EM67:
052444   103   114   122                .ASCII  \CLRD AC7. FPS BAD.\
052466   200   050   102                .ASCIZ  <CRLF>\(BUT FDST) ST 770 WENT TO 607 INSTEAD OF 617.\
052545                          EM70:
052545   103   114   122                .ASCII  \CLRD AC7. FEC BAD.\
052567   000                            .BYTE   0
052570   116   105   107  EM176:        .ASCIZ  'NEGF AC7. FPS BAD.'
052613   116   105   107  EM177:        .ASCIZ  'NEGF AC7. FEC BAD.'
052636                          EM71:
052636   116   105   107                .ASCIZ  \NEGF A FAILED.\
052655                          EM72:
052655   116   105   107                .ASCIZ  \NEGF A. FPS BAD.\
052676                          EM107:
052676   116   105   107                .ASCIZ  \NEGD (R) TRAP TO 4 IN SRC MODE.\
052736                          EM73:
052736   116   105   107                .ASCIZ  \NEGD (R) FAILED.\
052757                          EM74:
052757   116   105   107                .ASCIZ  \NEGD (R). R0 BAD.\
053001                          EM75:
053001   116   105   107                .ASCIZ  \NEGD (R). FPS BAD.\
053024                          EM76:
053024   101   102   123                .ASCIZ  \ABSD (R)+ TRAP TO 4 IN SRC MODE.\
053065                          EM77:
053065   101   102   123                .ASCIZ  \ABSD (R)+ FAILED.\
053107                          EM100:
053107   101   102   123                .ASCIZ  \ABSD (R)+. R0 BAD.\
053132                          EM101:
053132   101   102   123                .ASCIZ  \ABSD (R)+. FPS BAD.\
053156                          EM102:
053156   101   102   123                .ASCIZ  \ABSD -(R) TRAP TO 4 IN SRC MODE.\
053217                          EM103:
053217   101   102   123                .ASCIZ  \ABSD -(R) FAILED.\
053241                          EM104:
053241   101   102   123                .ASCIZ  \ABSD -(R). R0 BAD.\
053264                          EM105:
053264   101   102   123                .ASCIZ  \ABSD -(R). FPS BAD.\
053310                          EM106:
053310   101   102   123                .ASCIZ  \ABSD @(R)+ TRAP TO 4 IN SRC MODE.\
053352                          EM110:
053352   101   102   123                .ASCIZ  \ABSD @(R)+ FAILED.\
053375                          EM111:
053375   101   102   123                .ASCIZ  \ABSD @(R)+. R0 BAD.\
053421                          EM112:
053421   101   102   123                .ASCIZ  \ABSD @(R)+. FPS BAD.\
053446                          EM113:
053446   116   105   107                .ASCIZ  \NEGD @-(R) TRAP TO 4 IN SRC MODE.\
053510                          EM114:
053510   116   105   107                .ASCIZ  \NEGD @-(R) FAILED.\
053533                          EM115:
053533   116   105   107                .ASCIZ  \NEGD @-(R). R0 BAD.\
053557                          EM116:
053557   116   105   107                .ASCIZ  \NEGD @-(R). FPS BAD.\
```

```
7983 053604                             EM117:
     053604  101  102  123                       .ASCIZ  \ABSD N(R) TRAP TO 4 IN SRC MODE.\
7984 053645                             EM120:
     053645  101  102  123                       .ASCIZ  \ABSD N(R) FAILED.\
7985 053667                             EM121:
     053667  101  102  123                       .ASCIZ  \ABSD N(R). R0 BAD.\
7986 053712                             EM122:
     053712  101  102  123                       .ASCIZ  \ABSD N(R). FPS BAD.\
7987 053736                             EM123:
     053736  116  105  107                       .ASCIZ  \NEGD @N(R) TRAP TO 4 IN SRC MODE.\
7988 054000                             EM124:
     054000  116  105  107                       .ASCIZ  \NEGD @N(R) FAILED.\
7989 054023                             EM125:
     054023  116  105  107                       .ASCIZ  \NEGD @N(R). R0 BAD.\
7990 054047                             EM126:
     054047  116  105  107                       .ASCIZ  \NEGD @N(R). FPS BAD.\
7991 054074                             EM127:
     054074  116  105  107                       .ASCIZ  \NEGD N(R7) TRAP TO 4 IN SRC MODE.\
7992 054136                             EM130:
     054136  116  105  107                       .ASCIZ  \NEGD N(R7) FAILED.\
7993 054161                             EM131:
     054161  116  105  107                       .ASCIZ  \NEGD N(R7). FPS BAD.\
7994 054206                             EM132:
     054206  101  102  123                       .ASCIZ  \ABSD @N(R7) TRAP TO 4 IN SRC MODE.\
7995 054251                             EM133:
     054251  101  102  123                       .ASCIZ  \ABSD @N(R7) FAILED.\
7996 054275                             EM134:
     054275  101  102  123                       .ASCIZ  \ABSD @N(R7). FPS BAD.\
8006 054323  116  105  107             EM135:    .ASCII  'NEGD A FAILED.'
8007 054341  200  130  117                       .ASCIZ  <CRLF>'XOR SIGN BIT ST 336 FAILED.'
8008 054376                             EM136:
     054376  116  105  107                       .ASCIZ  \NEGD A FAILED.\
8009 054415                             EM137:
     054415  116  105  107                       .ASCIZ  \NEGD A. FPS BAD.\
8010 054436                             EM140:
     054436  116  105  107                       .ASCIZ  \NEGD (R) FAILED.\
8011 054457                             EM141:
     054457  116  105  107                       .ASCIZ  \NEGD (R). R0 BAD. SPECIAL DEST FAILED.\
8012 054526                             EM142:
     054526  116  105  107                       .ASCIZ  \NEGD (R). FPS BAD.\
8013 054551                             EM143:
     054551  116  105  107                       .ASCIZ  \NEGD (R)+ FAILED.\
8014 054573                             EM144:
     054573  116  105  107                       .ASCIZ  \NEGD (R)+. R0 BAD. SPECIAL DEST FAILED.\
8015 054643                             EM145:
     054643  116  105  107                       .ASCIZ  \NEGD (R)+. FPS BAD.\
8016 054667                             EM146:
     054667  116  105  107                       .ASCIZ  \NEGD -(R) FAILED.\
8017 054711                             EM147:
     054711  116  105  107                       .ASCIZ  \NEGD -(R). R0 BAD. SPECIAL DEST FAILED.\
8018 054761                             EM150:
     054761  116  105  107                       .ASCIZ  \NEGD -(R). FPS BAD.\
8019 055005                             EM151:
     055005  116  105  107                       .ASCIZ  \NEGD @(R)+ FAILED.\
8020 055030                             EM152:
     055030  116  105  107                       .ASCIZ  \NEGD @(R)+. R0 BAD. SPECIAL DEST FAILED.\
8021 055101                             EM153:
```

```
        055101  116  105  107                .ASCIZ  \NEGD @(R)+. FPS BAD.\
8022    055126                      EM154:
        055126  116  105  107                .ASCIZ  \NEGD @-(R) FAILED.\
8023    055151                      EM155:
        055151  116  105  107                .ASCIZ  \NEGD @-(R). R0 BAD. SPECIAL DEST FAILED.\
8024    055222                      EM156:
        055222  116  105  107                .ASCIZ  \NEGD @-(R). FPS BAD.\
8025    055247                      EM157:
        055247  116  105  107                .ASCIZ  \NEGF (R)+ FAILED.\
8026    055271  116  105  107      EM160:    .ASCII  'NEGF (R)+. R0 BAD.'
8027    055313  102  101  104                .ASCIZ  'BAD CONSTANT USED. SPECIAL DEST FAILED.'
8028    055363                      EM161:
        055363  116  105  107                .ASCIZ  \NEGF (R)+. FPS BAD.\
8029    055407                      EM162:
        055407  116  105  107                .ASCIZ  \NEGD (R7)+ FAILED.\
8030    055432                      EM163:
        055432  116  105  107                .ASCIZ  \NEGD (R7)+. FPS BAD.\
8031    055457  120  103  040      EM164:    .ASCIZ  'PC BAD AFTER NEGD (R7)+. BAD CONSTANT USED.'
8036    055533                      EM215:
        055533  120  103  040                .ASCII  \PC BAD AFTER NEGD N(R). BAD CONSTANT USED 746 746.\
        055615  200  117  122                .ASCIZ  <CRLF>'OR (BUT FDST) IN SPECIAL DEST FAILED.'
8037    055664                      EM216:
        055664  116  105  107                .ASCIZ  \NEGD N(R) FAILED.\
8038    055706                      EM217:
        055706  116  105  107                .ASCIZ  \NEGD N(R). R0 BAD. SPECIAL DEST FAILED.\
8039    055756                      EM220:
        055756  116  105  107                .ASCIZ  \NEGD N(R). FPS BAD.\
8040    056002                      EM221:
        056002  120  103  040                .ASCII  \PC BAD AFTER NEGD @N(R). BAD CONSTANT USED 747 747.\
        056065  200  117  122                .ASCIZ  <CRLF>'OR (BUT FDST) IN SPECIAL DEST FAILED.'
8041    056134                      EM222:
        056134  116  105  107                .ASCIZ  \NEGD @N(R) FAILED.\
8042    056157                      EM223:
        056157  116  105  107                .ASCIZ  \NEGD @N(R). R0 BAD. SPECIAL DEST FAILED.\
8043    056230                      EM224:
        056230  116  105  107                .ASCIZ  \NEGD @N(R). FPS BAD.\
8059    056255                      EM165:
        056255  116  105  107                .ASCIZ  \NEGD (R) FAILED.\
8060    056276                      EM166:
        056276  101  102  123                .ASCIZ  \ABSD (R) FAILED.\
8061    056317                      EM167:
        056317  124  123  124                .ASCIZ  \TSTD (R) FAILED.\
8062    056340                      EM170:
        056340  116  105  107                .ASCIZ  \NEGD (R). FPS BAD.\
8063    056363                      EM171:
        056363  101  102  123                .ASCIZ  \ABSD (R). FPS BAD.\
8064    056406                      EM172:
        056406  124  123  124                .ASCIZ  \TSTD (R). FPS BAD.\
8065    056431                      EM173:
        056431  116  105  107                .ASCIZ  \NEGD (R). FEC BAD.\
8066    056454                      EM174:
        056454  101  102  123                .ASCIZ  \ABSD (R). FEC BAD.\
8067    056477                      EM175:
        056477  124  123  124                .ASCIZ  \TSTD (R). FEC BAD.\
8068    056522                      EM200:
        056522  116  105  107                .ASCII  \NEGD (R) FAILED.\
8069    056542  200  130  117                .ASCIZ  <CRLF>'XOR SIGN BIT FAILED ST 336.'
```

```
8070 056577                            EM201:
     056577  116  105  107                    .ASCII  \NEGD (R). FPS BAD.\
8071 056621  200  050  102                    .ASCIZ  <CRLF>\(BUT ENBT) ST 336 WENT TO 053 INSTEAD OF 453.\
8072 056700                            EM202:
     056700  116  105  107                    .ASCII  \NEGD (R). FPS BAD.\
8073 056722  200  050  102                    .ASCIZ  <CRLF>\(BUT ENBT) ST 336 WENT TO 453 INSTEAD OF 053.\
8074 057001                            EM203:
     057001  101  102  123                    .ASCII  \ABSD (R) FAILED.\
8075 057021  200  050  102                    .ASCII  <CRLF>'(BUT OP1B) ST 055 WENT TO 336 INSTEAD OF 335, OR'
8076 057102  200  050  102                    .ASCIZ  <CRLF>\(BUT ENBT) ST 335 WENT TO 452 INSTEAD OF 052.\
8077 057161                            EM204:
     057161  101  102  123                    .ASCII  \ABSD (R) FAILED.\
8078 057201  200  130  117                    .ASCIZ  <CRLF>'XOR SIGN BIT FAILED ST 452.'
8079 057236                            EM205:
     057236  124  123  124                    .ASCII  \TSTD (R) FAILED.\
8080 057256  200  050  102                    .ASCIZ  <CRLF>\(BUT OP1B) ST 055 WENT TO 336 INSTEAD OF 334.\
8081 057335                            EM206:
     057335  124  123  124                    .ASCII  \TSTD (R). FPS BAD.\
8082 057357  200  050  102                    .ASCIZ  <CRLF>\(BUT ENBT) ST 334 WENT TO 453 INSTEAD OF 053.\
8083 057436                            EM207:
     057436  124  123  124                    .ASCII  \TSTD (R) FAILED.\
8084 057456  200  050  102                    .ASCIZ  <CRLF>\(BUT OP1B) ST 057 WENT TO 335 INSTEAD OF 334.\
8085 057535                            EM210:
     057535  124  123  124                    .ASCII  \TSTD (R) FAILED.\
8086 057555  200  050  102                    .ASCIZ  <CRLF>\(BUT ENBT) ST 334 WENT TO 053 INSTEAD OF 453.\
8087 057634                            EM211:
     057634  124  123  124                    .ASCII  \TSTD (R) FAILED.\
8088 057654  200  050  102                    .ASCIZ  <CRLF>\(BUT OP1B) ST 255 WENT TO 311 OR 312 INSTEAD OF 310.\
8089 057742                            EM212:
     057742  124  123  124                    .ASCII  \TSTD (R). FPS BAD.\
8090 057764  200  050  102                    .ASCIZ  <CRLF>\(BUT ENBT) ST 310 WENT TO 402 INSTEAD OF 002.\
8091 060043                            EM213:
     060043  124  123  124                    .ASCII  \TSTD (R). FPS BAD.\
8092 060065  040  106  111                    .ASCII  ' FIUV=0, OPERAND=-0.'
8093 060111  200  050  102                    .ASCIZ  <CRLF>\(BUT FIUV) ST 257 WENT TO 355 INSTEAD OF 255.\
8094 060170                            EM214:
     060170  124  123  124                    .ASCII  \TSTD (R). FPS BAD.\
8095 060212  040  106  111                    .ASCII  ' FIUV=1, OPERAND=-0.'
8096 060236  200  050  102                    .ASCIZ  <CRLF>\(BUT FIUV) ST 257 WENT TO 255 INSTEAD OF 355.\
8097
8107
8108 060315                            EM225:
     060315  114  104  106                    .ASCIZ  \LDFPS (R). R0 BAD.\
8109 060340                            EM226:
     060340  114  104  106                    .ASCIZ  \LDFPS (R). FPS BAD.\
8110 060364                            EM227:
     060364  114  104  106                    .ASCIZ  \LDFPS (R) TRAPPED TO 4.\
8111
8112 060414                            EM230:
     060414  114  104  106                    .ASCIZ  \LDFPS (R)+. R0 BAD.\
8113 060440                            EM231:
     060440  114  104  106                    .ASCIZ  \LDFPS (R)+. FPS BAD.\
8114 060465                            EM232:
     060465  114  104  106                    .ASCIZ  \LDFPS (R)+ TRAPPED TO 4.\
8115
8116 060516                            EM233:
     060516  114  104  106                    .ASCIZ  \LDFPS -(R). R0 BAD.\
```

```
8117 060542                         EM234:
     060542   114    104    106            .ASCIZ  \LDFPS -(R). FPS BAD.\
8118 060567                         EM235:
     060567   114    104    106            .ASCIZ  \LDFPS -(R) TRAPPED TO 4.\
8119
8120 060620                         EM236:
     060620   114    104    106            .ASCIZ  \LDFPS @(R)+. R0 BAD.\
8121 060645                         EM237:
     060645   114    104    106            .ASCIZ  \LDFPS @(R)+. FPS BAD.\
8122 060673                         EM240:
     060673   114    104    106            .ASCIZ  \LDFPS @(R)+ TRAPPED TO 4.\
8123
8124 060725                         EM241:
     060725   114    104    106            .ASCIZ  \LDFPS @-(R). R0 BAD.\
8125 060752                         EM242:
     060752   114    104    106            .ASCIZ  \LDFPS @-(R). FPS BAD.\
8126 061000                         EM243:
     061000   114    104    106            .ASCIZ  \LDFPS @-(R) TRAPPED TO 4.\
8127
8131
8132 061032                         EM244:
     061032   114    104    106            .ASCIZ  \LDFPS N(R). R0 BAD.\
8133 061056                         EM245:
     061056   114    104    106            .ASCIZ  \LDFPS N(R). FPS BAD.\
8134 061103                         EM246:
     061103   120    103    040            .ASCIZ  \PC BAD AFTER LDFPS N(R).\
8135 061134                         EM247:
     061134   114    104    106            .ASCIZ  \LDFPS N(R) TRAPPED TO 4.\
8136
8137 061165                         EM250:
     061165   114    104    106            .ASCIZ  \LDFPS @N(R). R0 BAD.\
8138 061212                         EM251:
     061212   114    104    106            .ASCIZ  \LDFPS @N(R). FPS BAD.\
8139 061240                         EM252:
     061240   120    103    040            .ASCIZ  \PC BAD AFTER LDFPS @N(R).\
8140 061272                         EM253:
     061272   114    104    106            .ASCIZ  \LDFPS @N(R) TRAPPED TO 4.\
8141
8142 061324                         EM254:
     061324   120    103    040            .ASCIZ  \PC BAD AFTER LDCLD (R7)+,A.\
8143 061360                         EM255:
     061360   114    104    103            .ASCIZ  \LDCLD (R7)+,A TRAPPED TO 4.\
8144
8145 061414                         EM256:
     061414   114    104    103            .ASCIZ  \LDCLD (R)+,A. R0 BAD.\
8146 061442                         EM257:
     061442   114    104    103            .ASCIZ  \LDCLD (R)+,A. FPS BAD.\
8147
8154
8155 061471                         EM260:
     061471   114    104    103            .ASCII  \LDCIF OR LDCLF (R),A FAILED.\
8156 061525   000                              .BYTE   0
8157 061526                         EM261:
     061526   114    104    103            .ASCII  \LDCIF OR LDCLF (R),A. FPS BAD.\
8158 061564   000                              .BYTE   0
8159
8160 061565                         EM262:
```

```
        061565  114  104  103          .ASCII  \LDCIF (R),A FAILED.\
 8161   061610  200  050  102          .ASCIZ  <CRLF>\(BUT FL) ST 277 WENT TO 300 INSTEAD OF 301.\
 8162
 8163   061665                  EM263:
        061665  114  104  103          .ASCII  \LDCLF (R),A. FPS BAD.\
 8164   061712  000                    .BYTE   0
 8165
 8166   061713                  EM264:
        061713  114  104  103          .ASCII  \LDCIF (R),A FAILED.\
 8167   061736  200  125  123          .ASCIZ  <CRLF>'USED CONSTANT 237 INSTEAD OF 217 ST 107.'
 8168
 8169   062010                  EM265:
        062010  114  104  103          .ASCII  \LDCIF OR LDCLF (R),A FAILED.\
 8170   062044  200  123  105          .ASCIZ  <CRLF>'SET SIGN BIT FAILED ST 146.'
 8171
 8172   062101                  EM266:
        062101  114  104  103          .ASCII  \LDCIF OR LDCLF (R),A FAILED.\
 8173   062135  200  050  102          .ASCIZ  <CRLF>\(BUT XNBT) ST 372 WENT TO 152 INSTEAD OF 112.\
 8174
 8175   062214                  EM267:
        062214  114  104  103          .ASCII  \LDCLF (R),A FAILED.\
 8176   062237  200  125  123          .ASCIZ  <CRLF>'USED CONSTANT 217 INSTEAD OF 237 ST 107.'
 8177
 8178   062311                  EM270:
        062311  114  104  103          .ASCII  \LDCLF (R),A FAILED.\
 8179   062334  040  122  117          .ASCIZ  ' ROUND ERROR.'
 8180
 8181   062352                  EM271:
        062352  114  104  103          .ASCII  \LDCLF (R),A FAILED.\
 8182   062375  040  124  122          .ASCIZ  ' TRUNCATION ERROR.'
 8183
 8184   062420                  EM272:
        062420  114  104  103          .ASCII  \LDCIF OR LDCLF (R),A FAILED.\
 8185   062454  200  122  061          .ASCIZ  <CRLF>'R14 NOT INCREMENTED ST 630.'
 8186
 8187   062511                  EM273:
        062511  114  104  103          .ASCII  \LDCID OR LDCLD (R),A FAILED.\
 8188   062545  000                    .BYTE   0
 8189   062546                  EM274:
        062546  114  104  103          .ASCII  \LDCID OR LDCLD (R),A. FPS BAD.\
 8190   062604  000                    .BYTE   0
 8191
 8192   062605                  EM275:
        062605  114  104  103          .ASCII  \LDCID (R),A FAILED.\
 8193   062630  200  050  102          .ASCIZ  <CRLF>\(BUT FL) ST 277 WENT TO 300 INSTEAD OF 301.\
 8194
 8195   062705                  EM276:
        062705  114  104  103          .ASCII  \LDCID (R),A FAILED.\
 8196   062730  200  125  123          .ASCIZ  <CRLF>'USED CONSTANT 237 INSTEAD OF 217 ST 107.'
 8197
 8198   063002                  EM277:
        063002  114  104  103          .ASCII  \LDCID (R),A FAILED.\
 8199   063025  200  123  105          .ASCIZ  <CRLF>'SET SIGN FAILED ST 146.'
 8200
 8201   063056                  EM300:
        063056  114  104  103          .ASCII  \LDCLD (R),A FAILED.\
 8202   063101  200  125  123          .ASCIZ  <CRLF>'USED CONSTANT 217 INSTEAD OF 237 ST 107.'
```

```
8203
8204 063153                          EM301:
     063153  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8205 063176  000                         .BYTE   0
8206 063177                          EM302:
     063177  114  104  105                 .ASCII  \LDEXP (R),A. FPS BAD.\
8207 063224  000                         .BYTE   0
8208 063225  114  104  105  EM303:       .ASCIZ  'LDEXP (R),A. FEC BAD.'
8209
8210 063253                          EM304:
     063253  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8211 063276  200  105  130                 .ASCIZ  <CRLF>'EXCESS 200 CALCULATION ST 624 BAD.'
8212
8213 063342                          EM305:
     063342  114  104  105                 .ASCII  \LDEXP (R),A. FPS BAD.\
8214 063367  050  102  125                 .ASCII  '(BUT ENBT,EZBT,XNBT) ST 625 DID NOT GO TO 304.'
8215
8216 063445                          EM306:
     063445  114  104  105                 .ASCII  \LDEXP (R),A. FPS BAD.\
8217 063472  200  050  102                 .ASCII  <CRLF>'(BUT EZBT) ST 544 WENT TO 504 INSTEAD OF 704, OR'
8218 063553  200  050  102                 .ASCIZ  <CRLF>\(BUT EZBT) ST 704 WENT TO 264 INSTEAD OF 064.\
8219
8220 063632                          EM307:
     063632  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8221 063655  200  050  102                 .ASCIZ  <CRLF>\(BUT EZBT) ST 704 WENT TO 064 INSTEAD OF 264.\
8222
8223 063734                          EM310:
     063734  114  104  105                 .ASCII  \LDEXP (R),A. FPS BAD.\
8224 063761  200  050  102                 .ASCIZ  <CRLF>\(BUT FIU) ST 264 WENT TO 115 INSTEAD OF 155.\
8225
8226 064037                          EM311:
     064037  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8227 064062  200  050  102                 .ASCIZ  <CRLF>\(BUT FIU) ST 264 WENT TO 155 INSTEAD OF 115.\
8228
8229 064140                          EM312:
     064140  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8230 064163  200  050  102                 .ASCIZ  <CRLF>\(BUT EZBT) ST 544 WENT TO 704 INSTEAD OF 504.\
8231
8232 064242                          EM313:
     064242  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8233 064265  200  050  102                 .ASCIZ  <CRLF>\(BUT FIU) ST 504 WENT TO 155 INSTEAD OF 115.\
8234
8235 064343                          EM314:
     064343  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8236 064366  200  050  102                 .ASCIZ  <CRLF>\(BUT FIV) ST 104 WENT TO 116 INSTEAD OF 136.\
8237
8238 064444                          EM315:
     064444  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8239 064467  200  050  102                 .ASCIZ  <CRLF>\(BUT FIV) ST 104 WENT TO 136 INSTEAD OF 116.\
8240
8241 064545                          EM316:
     064545  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8242 064570  200  050  102                 .ASCIZ  <CRLF>\(BUT FIV) ST 144 WENT TO 116 INSTEAD OF 136.\
8243
8244 064646                          EM317:
     064646  114  104  105                 .ASCII  \LDEXP (R),A FAILED.\
8245 064671  200  050  102                 .ASCIZ  <CRLF>\(BUT FIV) ST 144 WENT TO 136 INSTEAD OF 116.\
```

```
                                        EM320:
064747                                          .ASCII  \LDEXP (R),A FAILED.\
064747  114     104     105
064772  200     050     102                     .ASCIZ  <CRLF>\(BUT FIV) ST 344 WENT TO 116 INSTEAD OF 136.\

                                        EM321:
065050                                          .ASCII  \LDEXP (R),A FAILED.\
065050  114     104     105
065073  200     050     102                     .ASCIZ  <CRLF>\(BUT FIV) ST 344 WENT TO 136 INSTEAD OF 116.\

                                        EM322:
065151                                          .ASCII  \STCDI OR STCDL (R),A FAILED.\
065151  123     124     103
065205  000                                     .BYTE   0

                                        EM323:
065206                                          .ASCII  \STCDI OR STCDL (R),A. FPS BAD.\
065206  123     124     103
065244  000                                     .BYTE   0

065245  123     124     103     EM324:          .ASCIZ  'STCDI OR STCDL (R),A. FEC BAD.'

                                        EM325:
065304                                          .ASCII  \STCDL (R),A. FPS BAD.\
065304  123     124     103
065331  200     103     114                     .ASCII  <CRLF>'CLEAR FLAG ST 774 FAILED, OR'
065366  200     050     102                     .ASCIZ  <CRLF>\(BUT FLAG) ST 662 WENT TO 365 INSTEAD OF 361.\

        065304                          EM326=EM325

                                        EM327:
065445                                          .ASCII  \STCDL (R),A FAILED.\
065445  123     124     103
065470  200     050     102                     .ASCIZ  <CRLF>\(BUT ENBT) ST 632 WENT TO 473 INSTEAD OF 073.\

                                        EM330:
065547                                          .ASCII  \STCDL (R),A. FPS BAD.\
065547  123     124     103
065574  200     050     102                     .ASCIZ  <CRLF>\(BUT FIC) ST 004 WENT TO 305 INSTEAD OF 315.\

                                        EM331:
065652                                          .ASCII  \STCDL (R),A. FPS BAD.\
065652  123     124     103
065677  200     050     102                     .ASCIZ  <CRLF>\(BUT FIC) ST 004 WENT TO 315 INSTEAD OF 305.\

        065206                          EM333=EM323

                                        EM334:
065755                                          .ASCII  \STCDI (R),A. FPS BAD.\
065755  123     124     103
066002  200     125     123                     .ASCIZ  <CRLF>'USED CONSTANT 37 INSTEAD OF 17 ST 66.'

                                        EM335:
066051                                          .ASCII  \STCDI (R),A FAILED.\
066051  123     124     103
066074  200     050     102                     .ASCIZ  <CRLF>\(BUT ENBT) ST 632 WENT TO 073 INSTEAD OF 473.\

                                        EM336:
066153                                          .ASCII  \STCDI (R),A. FPS BAD.\
066153  123     124     103
```

```
8286 066200  200  123  105          .ASCIZ  <CRLF>'SET FN ST 473 FAILED.'
8287
8288 066227                   EM337:
     066227  123  124  103          .ASCII  \STCDL (R),A FAILED.\
8289 066252  200  050  102          .ASCIZ  <CRLF>\(BUT COUT) ST 275 WENT TO 074 INSTEAD OF 274.\
8290
8291 066331                   EM340:
     066331  123  124  103          .ASCII  \STCDL (R),A FAILED.\
8292 066354  200  050  102          .ASCIZ  <CRLF>\(BUT COUT) ST 275 WENT TO 274 INSTEAD OF 074.\
8293
8294 066433                   EM341:
     066433  123  124  103          .ASCII  \STCDL (R),A. FPS BAD.\
8295 066460  200  050  102          .ASCIZ  <CRLF>\(BUT EZBT) ST 377 WENT TO 633 INSTEAD OF 433.\
3296
8297 066537                   EM342:
     066537  123  124  103          .ASCII  \STCDL (R),A FAILED.\
8298 066562  200  050  102          .ASCIZ  <CRLF>\(BUT COUT) ST 360 WENT TO 654 INSTEAD OF 454.\
8299
8300 066641                   EM343:
     066641  123  124  103          .ASCII  \STCDL (R),A FAILED.\
8301 066664  200  050  102          .ASCIZ  <CRLF>\(BUT NBIT) ST 654 WENT TO 531 INSTEAD OF 431.\
8302
8303 066743                   EM344:
     066743  123  124  103          .ASCII  \STCDL (R),A FAILED.\
8304 066766  200  050  102          .ASCII  <CRLF>'(BUT COUT) ST 360 WENT TO 454 INSTEAD OF 654, OR'
8305 067047  200  050  102          .ASCIZ  <CRLF>\(BUT NBIT) ST 654 WENT TO 431 INSTEAD OF 531.\
8306
8307 067126                   EM332:
     067126  123  124  103          .ASCII  \STCDI (R),A FAILED.\
8308 067151  200  125  123          .ASCIZ  <CRLF>'USED CONSTANT 37 INSTEAD OF 17 ST 66.'
8309
8310 067220                   EM345:
     067220  123  124  103          .ASCII  \STCDI (R),A FAILED.\
8311 067243  200  050  102          .ASCIZ  <CRLF>\(BUT FL) ST 633 WENT TO 655 INSTEAD OF 654.\
8312
8313 067320                   EM346:
     067320  123  124  103          .ASCII  \STCFL (R),A FAILED.\
8314 067343  200  132  105          .ASCIZ  <CRLF>'ZERO LOW ORDER PART OF X11 FAILED ST 773.'
8315
8316
8317 067416                   EM347:
     067416  123  124  105          .ASCII  \STEXP A,(R) FAILED.\
8318 067441  000                    .BYTE   0
8319
8320 067442                   EM350:
     067442  123  124  105          .ASCII  \STEXP A,(R). FPS BAD.\
8321 067467  000                    .BYTE   0
8322
8323 067470  115  117  122  EM351:  .ASCII  'MORE THAN ONE WORD '
8324 067513  127  122  111          .ASCIZ  'WRITTEN BY STEXP A,(R).'<CRLF>'ZERO FDFL ST 347 FAILED.'
8325
8326 067574                   EM352:
     067574  123  124  105          .ASCII  \STEXP A,(R). FPS BAD.\
8327 067621  200  050  102          .ASCIZ  <CRLF>\(BUT ENBT) ST 376 WENT TO 071 INSTEAD OF 471.\
8328
8329 067700                   EM353:
     067700  123  124  105          .ASCII  \STEXP A,(R). FPS BAD.\
```

```
 8330 067725      200      050      102            .ASCIZ  <CRLF>\(BUT EZBT) ST 071 WENT TO 072 INSTEAD OF 272.\
 8331
 8332 070004                                EM354:
      070004      123      124      105            .ASCII  \STEXP A,(R). FPS BAD.\
 8333 070031      200      050      102            .ASCIZ  <CRLF>\(BUT EZBT) ST 071 WENT TO 272 INSTEAD OF 072.\
 8334
 8335 070110                                EM355:
      070110      123      124      105            .ASCII  \STEXP A,(R). FPS BAD.\
 8336 070135      200      050      102            .ASCIZ  <CRLF>\(BUT ENBT) ST 376 WENT TO 471 INSTEAD OF 071.\
 8337
 8338 070214      123      124      123  EM356:    .ASCII  'STST (R) GOT BAD FEC.'<CRLF>
 8339 070242      101      106      124            .ASCIZ  'AFTER EXECUTING AN ILLEGAL FPP OP CODE.'
 8340
 8341 070312      123      124      123  EM357:    .ASCII  'STST (R) GOT BAD FEA.'<CRLF>
 8342 070340      101      106      124            .ASCIZ  'AFTER EXECUTING AN ILLEGAL FPP OP CODE.'
 8343
 8344 070410      117      116      114  EM360:    .ASCII  'ONLY ONE WORD WRITTEN BY STST (R). '
 8345 070453      123      105      124            .ASCIZ  'SET FDFL ST 636 FAILED.'
 8346
```

```
8360
8361 070503                          EM401:
     070503  123  124  106                .ASCIZ  \STFPS (R). R0 BAD.\
8362 070526                          EM402:
     070526  123  124  106                .ASCIZ  \STFPS (R) FAILED.\
8363 070550  115  117  122           EM403:  .ASCII  'MORE THAN ONE WORD WRITTEN BY STFPS (R).'
8364 070620  200  050  102                .ASCIZ  <CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8365 070702                          EM404:
     070702  123  124  106                .ASCIZ  \STFPS (R) TRAPPED TO 4.\
8366
8367 070732                          EM405:
     070732  123  124  106                .ASCIZ  \STFPS (R)+. R0 BAD.\
8368 070756                          EM406:
     070756  123  124  106                .ASCIZ  \STFPS (R)+ FAILED.\
8369 071001  115  117  122           EM407:  .ASCII  'MORE THAN ONE WORD WRITTEN BY STFPS (R)+.'
8370 071052  200  050  102                .ASCIZ  <CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8371 071134                          EM410:
     071134  123  124  106                .ASCIZ  \STFPS (R)+ TRAPPED TO 4.\
8372
8373 071165                          EM411:
     071165  123  124  106                .ASCIZ  \STFPS -(R). R0 BAD.\
8374 071211                          EM412:
     071211  123  124  106                .ASCIZ  \STFPS -(R) FAILED.\
8375 071234  115  117  122           EM413:  .ASCII  'MORE THAN ONE WORD WRITTEN BY STFPS -(R).'
8376 071305  200  050  102                .ASCIZ  <CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8377 071367                          EM414:
     071367  123  124  106                .ASCIZ  \STFPS -(R) TRAPPED TO 4.\
8378
8379 071420                          EM415:
     071420  123  124  106                .ASCIZ  \STFPS @(R)+. R0 BAD.\
8380 071445                          EM416:
     071445  123  124  106                .ASCIZ  \STFPS @(R)+ FAILED.\
8381 071471  123  124  106           EM417:  .ASCIZ  'STFPS @(R)+ DID NOT DEFFER THE WRITE.'
8382 071537                          EM420:
     071537  123  124  106                .ASCIZ  \STFPS @(R)+ TRAPPED TO 4.\
8383
8384 071571                          EM421:
     071571  123  124  106                .ASCIZ  \STFPS @-(R). R0 BAD.\
8385 071616                          EM422:
     071616  123  124  106                .ASCIZ  \STFPS @-(R) FAILED.\
8386 071642  123  124  106           EM423:  .ASCIZ  'STFPS @-(R) DID NOT DEFFER THE WRITE.'
8387 071710                          EM424:
     071710  123  124  106                .ASCIZ  \STFPS @-(R) TRAPPED TO 4.\
8388
8389 071742                          EM425:
     071742  123  124  106                .ASCIZ  \STFPS N(R). R0 BAD.\
8390 071766                          EM426:
     071766  123  124  106                .ASCIZ  \STFPS N(R) FAILED.\
8391 072011  115  117  122           EM427:  .ASCII  'MORE THAN ONE WORD WRITTEN BY STFPS N(R).'
8392 072062  200  050  102                .ASCIZ  <CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8393 072144                          EM430:
     072144  123  124  106                .ASCIZ  \STFPS N(R) TRAPPED TO 4.\
8394 072175  120  103  040           EM431:  .ASCII  'PC BAD AFTER STFPS N(R). BAD CONSTANT USED.'
8395
8396 072250                          EM432:
     072250  123  124  106                .ASCIZ  \STFPS @N(R). R0 BAD.\
8397 072275                          EM433:
```

```
        072275      123     124     106             .ASCIZ  \STFPS @N(R) FAILED.\
 8398   072321      115     117     122     EM434:  .ASCII  'MORE THAN ONE WORD WRITTEN BY STFPS @N(R).'
 8399   072373      200     050     102             .ASCIZ  <CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
 8400   072455                              EM435:
        072455      123     124     106             .ASCIZ  \STFPS @N(R) TRAPPED TO 4.\
 8401   072507      120     103     040     EM436:  .ASCIZ  'PC BAD AFTER STFPS @N(R). BAD CONSTANT USED.'
 8402
 8403   072564                              EM437:
        072564      123     124     103             .ASCIZ  \STCDL A,(R)+. R0 BAD.\
 8404
 8405   072612                              EM440:
        072612      123     124     103             .ASCIZ  \STCDL A,-(R). R0 BAD.\
 8406
 8407   072640      123     124     123     EM361:  .ASCIZ  'STST (R). FPS BAD.'
 8408
 8409   072663      116     117     116     EM362:  .ASCII  'NON-RESIDENT MEMORY MANAGEMENT TRAP - IMPROPER D-SPACE ACCESS'
 8410   072760      040     101     124             .ASCIZ  ' ATTEMPTED'
 8411   072773      104     111     106     EM363:  .ASCIZ  'DIFFERENCE BETWEEN SR1 AND CALCULATED'
 8412   073041      106     120     120     EM364:  .ASCII  'FPP INSTRUCTION FAILED TO ABORT, NOT '
 8413   073106      101     114     114             .ASCIZ  'ALLOWNG EXAMINATION OF SR1'
 8414   073141      115     117     104     EM365:  .ASCIZ  'MODE 0 INSTRUCTION ABORTED WHEN IT SHOULD NOT HAVE'
 8415   073224      106     120     120     EM366:  .ASCIZ  'FPP ACCUMULATOR WAS CHANGED IN THE EXPECTED ABORT.'
 8416   073307      107     105     116     EM367:  .ASCIZ  'GENERAL REGISTER WAS CHANGED IN THE EXPECTED ABORT'
 8417   073372      106     120     120     EM370:  .ASCIZ  'FPP UNABLE TO RESTORE AN AC'
 8418           000000                      EM371=0
 8419           000000                      EM372=0
 8420           000000                      EM373=0
 8421           000000                      EM374=0
 8422           000000                      EM375=0
 8423           000000                      EM376=0
 8424           000000                      EM377=0
 8425           000000                      EM400=0
 8426
 8427   073426      125     116     105     EM441:  .ASCIZ  'UNEXPECTED FPP TRAP TO 244.'
 8428   073462      125     116     105     EM442:  .ASCIZ  'UNEXPECTED CPU TRAP TO 4.'
 8429   073514      125     116     105     EM443:  .ASCIZ  'UNEXPECTED CPU TRAP TO 10.'
 8430
 8431                                        ;THESE ARE DATA TABLE HEADERS:
 8432
 8433   073547      040     040     124     DH1:    .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 8434   073607      011     106     120             .ASCIZ  <TAB>'FPS.'<TAB>'FEC.'
 8438   073622                              DH2:
        073622      040     040     124             .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 8439   073662      011     107     117             .ASCIZ  <TAB>'GOT FPS.'<TAB>'EXPECTED FPS.'
 8440   073712                              DH3:
        073712      040     040     124             .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 8441   073752      011     107     117             .ASCIZ  <TAB>'GOT FEC.'<TAB>'EXPECTED FEC.'
 8442   074002                              DH4:
        074002      040     040     124             .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 8443   074042      011     107     117             .ASCIZ  <TAB>'GOT R0. '<TAB>'EXPECTED R0.'
 8444   074071                              DH5:
        074071      040     040     124             .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 8445   074131      000                             .BYTE   0
 8446           074071                      DH6=DH5
 8447           074002                      DH7=DH4
 8448           074071                      DH10=DH5
 8449           074002                      DH11=DH4
```

```
8450         074071                   DH12=DH5
8451 074132     040      040      124 DH13:    .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
8452         074132                   DH14=DH13
8453         074071                   DH15=DH5
8454 074172                           DH16:
     074172     040      040      124          .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
8455 074232     011      107      117          .ASCIZ  <TAB>'GOT PC. '<TAB>'EXPECTED PC.'
8456         074132                   DH17=DH13
8457         074002                   DH20=DH4
8458         074071                   DH21=DH5
8459         074071                   DH22=DH5
8460         074132                   DH23=DH13
8461         074002                   DH24=DH4
8462         074071                   DH25=DH5
8463         074132                   DH26=DH13
8464         074002                   DH27=DH4
8465         074071                   DH30=DH5
8466         074132                   DH31=DH13
8467         074002                   DH32=DH4
8468         074071                   DH33=DH5
8469         074132                   DH34=DH13
8470         074002                   DH35=DH4
8471         074071                   DH36=DH5
8472 074261     040      040      124 DH37:    .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>'GOT FPS.'<TAB>'EXPECTED
8473         074261                   DH40=DH37
8474 074351     040      040      124 DH41:    .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>'FPS.'<TAB>'GOT FEC. EXPE
8475         074261                   DH42=DH37
8476         074261                   DH43=DH37
8477         074261                   DH44=DH37
8478         074261                   DH45=DH37
8479         074261                   DH46=DH37
8480         074261                   DH47=DH37
8481         074261                   DH50=DH37
8482         074261                   DH51=DH37
8483         074261                   DH52=DH37
8484         074351                   DH53=DH41
8485         074261                   DH54=DH37
8486         074261                   DH55=DH37
8487         074261                   DH56=DH37
8488         074261                   DH57=DH37
8489         074261                   DH60=DH37
8490         074261                   DH61=DH37
8491         073622                   DH62=DH2
8492         073712                   DH63=DH3
8493         074071                   DH64=DH5
8494         073622                   DH65=DH2
8495         074002                   DH66=DH4
8496         073622                   DH67=DH2
8497         073712                   DH70=DH3
8498         073622                   DH176=DH2
8499         073712                   DH177=DH3
8500         074071                   DH71=DH5
8501         073622                   DH72=DH2
8502         074132                   DH107=DH13
8503         074071                   DH73=DH5
8504         074002                   DH74=DH4
8505         073622                   DH75=DH2
```

```
8506    074132          DH76=DH107
8507    074071          DH77=DH5
8508    074002          DH100=DH4
8509    073622          DH101=DH2
8510    074132          DH102=DH107
8511    074071          DH103=DH5
8512    074002          DH104=DH4
8513    073622          DH105=DH2
8514    074132          DH106=DH107
8515    074071          DH110=DH5
8516    074002          DH111=DH4
8517    073622          DH112=DH2
8518    074132          DH113=DH107
8519    074071          DH114=DH5
8520    074002          DH115=DH4
8521    073622          DH116=DH2
8522    074132          DH117=DH107
8523    074071          DH120=DH5
8524    074002          DH121=DH4
8525    073622          DH122=DH2
8526    074132          DH123=DH107
8527    074071          DH124=DH5
8528    074002          DH125=DH4
8529    073622          DH126=DH2
8530    074132          DH127=DH107
8531    074071          DH130=DH5
8532    073622          DH131=DH2
8533    074132          DH132=DH107
8534    074071          DH133=DH5
8535    073622          DH134=DH2
8536    074071          DH135=DH5
8537    074071          DH136=DH5
8538    073622          DH137=DH2
8539    074071          DH140=DH5
8540    074002          DH141=DH4
8541    073622          DH142=DH2
8542    074071          DH143=DH5
8543    074002          DH144=DH4
8544    073622          DH145=DH2
8545    074071          DH146=DH5
8546    074002          DH147=DH4
8547    073622          DH150=DH2
8548    074071          DH151=DH5
8549    074002          DH152=DH4
8550    073622          DH153=DH2
8551    074071          DH154=DH5
8552    074002          DH155=DH4
8553    073622          DH156=DH2
8554    074071          DH157=DH5
8555    074002          DH160=DH4
8556    073622          DH161=DH2
8557    074071          DH162=DH5
8558    073622          DH163=DH2
8559    074172          DH164=DH16
8560    074172          DH215=DH16
8561    074071          DH216=DH5
8562    074002          DH217=DH4
```

```
         073622                       DH220=DH2
         074172                       DH221=DH16
         074071                       DH222=DH5
         074002                       DH223=DH4
         073622                       DH224=DH2
         074261                       DH165=DH37
         074261                       DH166=DH37
         074261                       DH167=DH37
         074261                       DH170=DH37
         074261                       DH171=DH37
         074261                       DH172=DH37
         074351                       DH173=DH41
         074351                       DH174=DH41
         074351                       DH175=DH41
         074261                       DH200=DH37
         074261                       DH201=DH37
         074261                       DH202=DH37
         074261                       DH203=DH37
         074261                       DH204=DH37
         074261                       DH205=DH37
         074261                       DH206=DH37
         074261                       DH207=DH37
         074261                       DH210=DH37
         074261                       DH211=DH37
         074261                       DH212=DH37
         074261                       DH213=DH37
         074261                       DH214=DH37

         074002                       DH225=DH4
         073622                       DH226=DH2
074446      040      040      124     DH227:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
         074002                       DH230=DH4
         073622                       DH231=DH2
         074446                       DH232=DH227
         074002                       DH233=DH4
         073622                       DH234=DH2
         074446                       DH235=DH227
         074002                       DH236=DH4
         073622                       DH237=DH2
         074446                       DH240=DH227
         074002                       DH241=DH4
         073622                       DH242=DH2
         074446                       DH243=DH227
         074002                       DH244=DH4
         073622                       DH245=DH2
         074172                       DH246=DH16
         074446                       DH247=DH227
         074002                       DH250=DH4
         073622                       DH251=DH2
         074172                       DH252=DH16
         074446                       DH253=DH227
         074172                       DH254=DH16
         074446                       DH255=DH227
         074002                       DH256=DH4
         073622                       DH257=DH2
         074261                       DH260=DH37
         074261                       DH261=DH37
```

```
8620    074261          DH262=DH37
8621    074261          DH263=DH37
8622    074261          DH264=DH37
8623    074261          DH265=DH37
8624    074261          DH266=DH37
8625    074261          DH267=DH37
8626    074261          DH270=DH37
8627    074261          DH271=DH37
8628    074261          DH272=DH37
8629    074261          DH273=DH37
8630    074261          DH274=DH37
8631    074261          DH275=DH37
8632    074261          DH276=DH37
8633    074261          DH277=DH37
8634    074261          DH300=DH37
8635    074261          DH301=DH37
8636    074261          DH302=DH37
8637    074351          DH303=DH41
8638    074261          DH304=DH37
8639    074261          DH305=DH37
8640    074261          DH306=DH37
8641    074261          DH307=DH37
8642    074261          DH310=DH37
8643    074261          DH311=DH37
8644    074261          DH312=DH37
8645    074261          DH313=DH37
8646    074261          DH314=DH37
8647    074261          DH315=DH37
8648    074261          DH316=DH37
8649    074261          DH317=DH37
8650    074261          DH320=DH37
8651    074261          DH321=DH37
8652    074261          DH322=DH37
8653    074261          DH323=DH37
8654    074351          DH324=DH41
8655    074261          DH325=DH37
8656    074261          DH326=DH37
8657    074261          DH327=DH37
8658    074261          DH330=DH37
8659    074261          DH331=DH37
8660    074261          DH332=DH37
8661    074261          DH333=DH37
8662    074261          DH334=DH37
8663    074261          DH335=DH37
8664    074261          DH336=DH37
8665    074261          DH337=DH37
8666    074261          DH340=DH37
8667    074261          DH341=DH37
8668    074261          DH342=DH37
8669    074261          DH343=DH37
8670    074261          DH344=DH37
8671    074261          DH345=DH37
8672    074261          DH346=DH37
8673    074261          DH347=DH37
8674    074261          DH350=DH37
8675    074132          DH351=DH13
8676    074261          DH352=DH37
```

```
               074261                        DH353=DH37
               074261                        DH354=DH37
               074261                        DH355=DH37
               074002                        DH356=DH11
        074506    040      040      124      DH357:   .ASCII   '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
        074546    011      107      117               .ASCIZ   <TAB>'GOT FEA.'<TAB>'EXPECTED FEA.'
               074132                        DH360=DH13
               073622                        DH361=DH2

        074576    040      040      124      DH362:   .ASCIZ   '  TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'<TAB>'MMR0'
        074641    040      040      124      DH363:   .ASCII   '  TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'<TAB>
        074677    123      122      061               .ASCIZ   'SR1'<TAB>'CALCULATED'
        074716    040      040      124      DH364:   .ASCII   '  TEST'<TAB>'PC OF CALL'<TAB>'PC OF INSTRUCTION FAILING'
        074771    040      124      117               .ASCIZ   ' TO ABORT'
        075003    040      040      124      DH365:   .ASCIZ   '  TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'
        075041    040      040      124      DH366:   .ASCII   '  TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'<TAB>'AC #'
        075103    040      103      110               .ASCIZ   ' CHANGED'
        075114    040      040      124      DH367:   .ASCII   '  TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'<TAB>'REG #'
        075157    011      122      105               .ASCIZ   <TAB>'RECEIVED'<TAB>'EXPECTED'
        075202    040      040      124      DH370:   .ASCIZ   '  TEST'<TAB>'PC OF CALL'<TAB>'AC #'<TAB>'PC OF ERROR'
               000000                        DH371=0
               000000                        DH372=0
               000000                        DH373=0
               000000                        DH374=0
               000000                        DH375=0
               000000                        DH376=0
               000000                        DH377=0
               000000                        DH400=0
```

```
 8707            074002                  DH401=DH4
 8708            073622                  DH402=DH2
 8709            074132                  DH403=DH13
 8710            074446                  DH404=DH227
 8711            074002                  DH405=DH4
 8712            073622                  DH406=DH2
 8713            074132                  DH407=DH13
 8714            074446                  DH410=DH227
 8715            074002                  DH411=DH4
 8716            073622                  DH412=DH2
 8717            074132                  DH413=DH13
 8718            074446                  DH414=DH227
 8719            074002                  DH415=DH4
 8720            073622                  DH416=DH2
 8721            074132                  DH417=DH13
 8722            074446                  DH420=DH227
 8723            074002                  DH421=DH4
 8724            073622                  DH422=DH2
 8725            074132                  DH423=DH13
 8726            074446                  DH424=DH227
 8727            074002                  DH425=DH4
 8728            073622                  DH426=DH2
 8729            074132                  DH427=DH13
 8730            074446                  DH430=DH227
 8731            074132                  DH431=DH13
 8732            074002                  DH432=DH4
 8733            073622                  DH433=DH2
 8734            074132                  DH434=DH13
 8735            074446                  DH435=DH227
 8736            074132                  DH436=DH13
 8737            074002                  DH437=DH4
 8738            074002                  DH440=DH4
 8739 075245      040      040     124   DH441:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>'FEC.'
 8740 075313      040      040     124   DH442:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 8741            075313                  DH443=DH442
 8742
 8743
 8744
 8745                                    ;THESE ARE FORMAT SPECIFICATIONS FOR THE DATA TABLES:
 8746 075354      004      000     005   DF1:    .BYTE   4,0,5,0,5,0,0
 8747 075363      004      000     005   DF2:    .BYTE   4,0,5,0,5,0,5,0
 8748            075363                  DF3=DF2
 8749            075363                  DF4=DF2
 8750 075373      004      000     005   DF5:    .BYTE   4,0,5,0,5,5,2,5,5,2
 8751 075405      004      000     005   DF6:    .BYTE   4,0,5,0
 8752            075363                  DF7=DF4
 8753            075373                  DF10=DF5
 8754            075363                  DF11=DF4
 8755 075411      004      000     005   DF12:   .BYTE   4,0,5,0,5,5,3,5,5,3
 8756            075405                  DF13=DF6
 8757            075405                  DF14=DF6
 8758            075411                  DF15=DF12
 8759            075363                  DF16=DF2
 8760            075405                  DF17=DF6
 8761            075363                  DF20=DF2
 8762            075411                  DF21=DF12
 8763            075411                  DF22=DF12
```

```
 8764            075405                          DF23=DF6
 8765            075363                          DF24=DF2
 8766            075411                          DF25=DF12
 8767            075405                          DF26=DF6
 8768            075363                          DF27=DF2
 8769            075411                          DF30=DF12
 8770            075405                          DF31=DF6
 8771            075363                          DF32=DF2
 8772            075411                          DF33=DF12
 8773            075405                          DF34=DF6
 8774            075363                          DF35=DF2
 8775            075411                          DF36=DF12
 8776 075423       004       000       005       DF37:    .BYTE   4,0,5,0,5,0,5,0,5,5,3,5,5,3,5,5,3
 8777            075423                          DF40=DF37
 8778 075444       004       000       005       DF41:    .BYTE   4,0,5,0,5,0,0,0,5,5,3,5,5,3,5,5,3
 8779            075423                          DF42=DF37
 8780            075423                          DF43=DF37
 8781            075423                          DF44=DF37
 8782            075423                          DF45=DF37
 8783            075423                          DF46=DF37
 8784            075423                          DF47=DF37
 8785            075423                          DF50=DF37
 8786            075423                          DF51=DF37
 8787            075423                          DF52=DF37
 8788            075423                          DF53=DF37
 8789            075423                          DF54=DF37
 8790            075423                          DF55=DF37
 8791            075423                          DF56=DF37
 8792            075423                          DF57=DF37
 8793            075423                          DF60=DF37
 8794            075423                          DF61=DF37
 8795            075363                          DF62=DF2
 8796            075363                          DF63=DF2
 8797            075373                          DF64=DF5
 8798            075363                          DF65=DF2
 8799            075363                          DF66=DF2
 8800            075363                          DF67=DF2
 8801            075363                          DF70=DF2
 8802            075363                          DF176=DF2
 8803            075363                          DF177=DF2
 8804 075465       004       000       005       DF71:    .BYTE   4,0,5,0,5,5,3,5,5,3,5,5,3
 8805            075363                          DF72=DF2
 8806            075405                          DF107=DF6
 8807            075465                          DF73=DF71
 8808            075363                          DF74=DF2
 8809            075363                          DF75=DF2
 8810            075405                          DF76=DF6
 8811            075465                          DF77=DF71
 8812            075363                          DF100=DF2
 8813            075363                          DF101=DF2
 8814            075405                          DF102=DF6
 8815            075465                          DF103=DF71
 8816            075363                          DF104=DF2
 8817            075363                          DF105=DF2
 8818            075405                          DF106=DF6
 8819            075465                          DF110=DF71
 8820            075363                          DF111=DF2
```

```
8821    075363                  DF112=DF2
8822    075405                  DF113=DF6
8823    075465                  DF114=DF71
8824    075363                  DF115=DF2
8825    075363                  DF116=DF2
8826    075405                  DF117=DF6
8827    075465                  DF120=DF71
8828    075363                  DF121=DF2
8829    075363                  DF122=DF2
8830    075405                  DF123=DF6
8831    075465                  DF124=DF71
8832    075363                  DF125=DF2
8833    075363                  DF126=DF2
8834    075405                  DF127=DF6
8835    075465                  DF130=DF71
8836    075363                  DF131=DF2
8837    075405                  DF132=DF6
8838    075465                  DF133=DF71
8839    075363                  DF134=DF2
8840    075411                  DF135=DF12
8841    075411                  DF136=DF12
8842    075363                  DF137=DF2
8843    075411                  DF140=DF12
8844    075363                  DF141=DF2
8845    075363                  DF142=DF2
8846    075411                  DF143=DF12
8847    075363                  DF144=DF2
8848    075363                  DF145=DF2
8849    075411                  DF146=DF12
8850    075363                  DF147=DF2
8851    075363                  DF150=DF2
8852    075411                  DF151=DF12
8853    075363                  DF152=DF2
8854    075363                  DF153=DF2
8855    075411                  DF154=DF12
8856    075363                  DF155=DF2
8857    075363                  DF156=DF2
8858    075411                  DF157=DF12
8859    075363                  DF160=DF2
8860    075363                  DF161=DF2
8861    075411                  DF162=DF12
8862    075363                  DF163=DF2
8863    075363                  DF164=DF2
8864    075363                  DF215=DF2
8865    075411                  DF216=DF12
8866    075363                  DF217=DF2
8867    075363                  DF220=DF2
8868    075363                  DF221=DF2
8869    075411                  DF222=DF12
8870    075363                  DF223=DF2
8871    075363                  DF224=DF2
8872    075423                  DF165=DF37
8873    075423                  DF166=DF37
8874    075423                  DF167=DF37
8875    075423                  DF170=DF37
8876    075423                  DF171=DF37
8877    075423                  DF172=DF37
```

```
                 075444                      DF173=DF41
                 075444                      DF174=DF41
                 075444                      DF175=DF41
                 075423                      DF200=DF37
                 075423                      DF201=DF37
                 075423                      DF202=DF37
                 075423                      DF203=DF37
                 075423                      DF204=DF37
                 075423                      DF205=DF37
                 075423                      DF206=DF37
                 075423                      DF207=DF37
                 075423                      DF210=DF37
                 075423                      DF211=DF37
                 075423                      DF212=DF37
                 075423                      DF213=DF37
                 075423                      DF214=DF37
        075502      004      000      005    DF225:    .BYTE    4,0,5,0,5,0,5,0
                 075502                      DF226=DF225
        075512      004      000      005    DF227:    .BYTE    4,0,5,0
                 075502                      DF230=DF225
                 075502                      DF231=DF225
                 075512                      DF232=DF227
                 075502                      DF233=DF225
                 075502                      DF234=DF225
                 075512                      DF235=DF227
                 075502                      DF236=DF225
                 075502                      DF237=DF225
                 075512                      DF240=DF227
                 075502                      DF241=DF225
                 075502                      DF242=DF225
                 075512                      DF243=DF227
                 075502                      DF244=DF225
                 075502                      DF245=DF225
                 075502                      DF246=DF225
                 075512                      DF247=DF227
                 075502                      DF250=DF225
                 075502                      DF251=DF225
                 075502                      DF252=DF225
                 075512                      DF253=DF227
                 075502                      DF254=DF225
                 075512                      DF255=DF227
                 075502                      DF256=DF225
                 075502                      DF257=DF225

        075516      004      000      005    DF260:    .BYTE    4,0,5,0,5,0,5,0,5,5,2,5,5,2,5,5,2
                 075516                      DF261=DF260
                 075516                      DF262=DF260
                 075516                      DF263=DF260
                 075516                      DF264=DF260
                 075516                      DF265=DF260
                 075516                      DF266=DF260
                 075516                      DF267=DF260
                 075516                      DF270=DF260
                 075516                      DF271=DF260
                 075516                      DF272=DF260

        075537      004      000      005    DF273:    .BYTE    4,0,5,0,5,0,5,0,5,5,2,5,5,3,5,5,3
```

```
 8935            075537                          DF274=DF273
 8936            075537                          DF275=DF273
 8937            075537                          DF276=DF273
 8938            075537                          DF277=DF273
 8939            075537                          DF300=DF273
 8940 075560       004       000       005       DF301:  .BYTE   4,0,5,0,5,0,5,0,5,5,3,5,5,0,5,5,3,5,5,3
 8941            075560                          DF302=DF301
 8942 075604       004       000       005       DF303:  .BYTE   4,0,5,0,5,0,0,0,5,5,3,5,5,0,5,5,3,5,5,3
 8943            075560                          DF304=DF301
 8944            075560                          DF305=DF301
 8945            075560                          DF306=DF301
 8946            075560                          DF307=DF301
 8947            075560                          DF310=DF301
 8948            075560                          DF311=DF301
 8949            075560                          DF312=DF301
 8950            075560                          DF313=DF301
 8951            075560                          DF314=DF301
 8952            075560                          DF315=DF301
 8953            075560                          DF316=DF301
 8954            075560                          DF317=DF301
 8955            075560                          DF320=DF301
 8956            075560                          DF321=DF301
 8957
 8958 075630       004       000       005       DF322:  .BYTE   4,0,5,0,5,0,5,0,5,5,3,5,5,2,5,5,2
 8959            075630                          DF323=DF322
 8960 075651       004       000       005       DF324:  .BYTE   4,0,5,0,5,0,0,0,5,5,3,5,5,2,5,5,2
 8961            075630                          DF325=DF322
 8962            075630                          DF326=DF322
 8963            075630                          DF327=DF322
 8964            075630                          DF330=DF322
 8965            075630                          DF331=DF322
 8966            075630                          DF332=DF322
 8967            075630                          DF333=DF322
 8968            075630                          DF334=DF322
 8969            075630                          DF335=DF322
 8970            075630                          DF336=DF322
 8971            075630                          DF337=DF322
 8972            075630                          DF340=DF322
 8973            075630                          DF341=DF322
 8974            075630                          DF342=DF322
 8975            075630                          DF343=DF322
 8976            075630                          DF344=DF322
 8977            075630                          DF345=DF322
 8978            075630                          DF346=DF322
 8979
 8980 075672       004       000       005       DF347:  .BYTE   4,0,5,0,5,0,5,0,5,5,3,5,5,0,5,5,0
 8981            075672                          DF350=DF347
 8982            075512                          DF351=DF227
 8983            075672                          DF352=DF347
 8984            075672                          DF353=DF347
 8985            075672                          DF354=DF347
 8986            075672                          DF355=DF347
 8987            075502                          DF356=DF225
 8988            075502                          DF357=DF225
 8989            075512                          DF360=DF227
 8990            075502                          DF361=DF225
 8991
```

```
 8992 075713     004     000     005  DF362:  .BYTE   4,0,5,0,5,0
 8993 075721     004     000     005  DF363:  .BYTE   4,0,5,0,5,0,0
 8994          075405                  DF364=DF6
 8995          075405                  DF365=DF6
 8996 075730     004     000     005  DF366:  .BYTE   4,0,5,0,5,0,5,5,0,0,0,0,5,5,0,0,0,0
 8997 075752     004     000     005  DF367:  .BYTE   4,0,5,0,5,0,5,0
 8998 075762     004     000     005  DF370:  .BYTE   4,0,5,0,0,5,5,0,0,0,0,5,5,0,0,0,0
 8999          000000                  DF371=0
 9000          000000                  DF372=0
 9001          000000                  DF373=0
 9002          000000                  DF374=0
 9003          000000                  DF375=0
 9004          000000                  DF376=0
 9005          000000                  DF377=0
 9006          000000                  DF400=0
 9007
 9008          075502                  DF401=DF225
 9009          075502                  DF402=DF225
 9010          075512                  DF403=DF227
 9011          075512                  DF404=DF227
 9012          075502                  DF405=DF225
 9013          075502                  DF406=DF225
 9014          075512                  DF407=DF227
 9015          075512                  DF410=DF227
 9016          075502                  DF411=DF225
 9017          075502                  DF412=DF225
 9018          075512                  DF413=DF227
 9019          075512                  DF414=DF227
 9020          075502                  DF415=DF225
 9021          075502                  DF416=DF225
 9022          075512                  DF417=DF227
 9023          075512                  DF420=DF227
 9024          075502                  DF421=DF225
 9025          075502                  DF422=DF225
 9026          075512                  DF423=DF227
 9027          075512                  DF424=DF227
 9028          075502                  DF425=DF225
 9029          075502                  DF426=DF225
 9030          075512                  DF427=DF227
 9031          075512                  DF430=DF227
 9032          075512                  DF431=DF227
 9033          075502                  DF432=DF225
 9034          075502                  DF433=DF225
 9035          075512                  DF434=DF227
 9036          075512                  DF435=DF227
 9037          075512                  DF436=DF227
 9038          075502                  DF437=DF225
 9039          075502                  DF440=DF225
 9040 076003     004     000     005  DF441:  .BYTE   4,0,5,0,5,0
 9041          076003                  DF442=DF441
 9042          076003                  DF443=DF441
 9043
 9044
 9045                                  .EVEN
 9046
 9047                                  ;THESE ARE THE ERROR MESSAGE DATA TABLES:
 9048
```

```
 9049 076012  001232  001234  046435  DT1:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TMP4,0
 9050 076032  001232  001234  046435  DT2:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TAB,$TMP5,0
 9051 076054  001232  001234  046435  DT3:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TAB,$TMP6,0
 9052 076076  001232  001234  046435  DT4:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TAB,$TMP3,0
 9053 076120  001232  001234  046435  DT5:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS1,$TMP3
 9054 076136  001313  046455  001242          .WORD   $CRLF,MS2,$TMP4,0
 9055 076146  001232  001234  046435  DT6:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,0
 9056         076076                  DT7=DT4
 9057         076120                  DT10=DT5
 9058         076076                  DT11=DT4
 9059         076120                  DT12=DT5
 9060         076146                  DT13=DT6
 9061         076146                  DT14=DT6
 9062         076120                  DT15=DT5
 9063 076160  001232  001234  046435  DT16:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TAB,$TMP3,0
 9064         076146                  DT17=DT6
 9065         076160                  DT20=DT16
 9066         076120                  DT21=DT5
 9067         076120                  DT22=DT5
 9068         076146                  DT23=DT6
 9069         076160                  DT24=DT16
 9070         076120                  DT25=DT5
 9071         076146                  DT26=DT6
 9072         076160                  DT27=DT16
 9073         076120                  DT30=DT5
 9074         076146                  DT31=DT6
 9075         076160                  DT32=DT16
 9076         076120                  DT33=DT5
 9077         076146                  DT34=DT6
 9078         076160                  DT35=DT16
 9079         076120                  DT36=DT5
 9080 076202  001232  001234  046435  DT37:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TAB,$TMP10,$CRLF
 9081 076224  046515  001240  001313          .WORD   MS4,$TMP3,$CRLF,MS1,$TMP4,$CRLF,MS2,$TMP5,0
 9082         076202                  DT40=DT37
 9083 076246  001232  001234  046435  DT41:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TMP11,$TMP12
 9084 076266  001313  046515  001240          .WORD   $CRLF,MS4,$TMP3,$CRLF,MS1,$TMP4,$CRLF,MS2,$TMP5,0
 9085         076202                  DT42=DT37
 9086         076202                  DT43=DT37
 9087         076202                  DT44=DT37
 9088         076202                  DT45=DT37
 9089         076202                  DT46=DT37
 9090         076202                  DT47=DT37
 9091         076202                  DT50=DT37
 9092         076202                  DT51=DT37
 9093         076202                  DT52=DT37
 9094         076246                  DT53=DT41
 9095         076202                  DT54=DT37
 9096         076202                  DT55=DT37
 9097         076202                  DT56=DT37
 9098         076202                  DT57=DT37
 9099         076202                  DT60=DT37
 9100         076202                  DT61=DT37
 9101         076160                  DT62=DT16
 9102         076160                  DT63=DT16
 9103         076120                  DT64=DT5
 9104         076160                  DT65=DT16
 9105         076076                  DT66=DT4
```

```
9106            076076                  DT67=DT4
9107            076076                  DT70=DT4
9108            076076                  DT176=DT4
9109            076076                  DT177=DT4
9110 076312 001232  001234  046435  DT71:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS3,$TMP3,$CRLF,MS1
9111 076334 001244  001313  046455          .WORD   $TMP5,$CRLF,MS2,$TMP4,0
9112            076076                  DT72=DT4
9113            076146                  DT107=DT6
9114 076346 001232  001234  046435  DT73:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS4,$TMP4
9115 076364 001313  046437  001244          .WORD   $CRLF,MS1,$TMP5,$CRLF,MS2,$TMP3,0
9116            076076                  DT74=DT4
9117            076032                  DT75=DT2
9118            076146                  DT76=DT6
9119            076346                  DT77=DT73
9120            076076                  DT100=DT4
9121            076032                  DT101=DT2
9122            076146                  DT102=DT6
9123            076346                  DT103=DT73
9124            076076                  DT104=DT4
9125            076032                  DT105=DT2
9126            076146                  DT106=DT6
9127            076346                  DT110=DT73
9128            076076                  DT111=DT4
9129            076032                  DT112=DT2
9130            076146                  DT113=DT6
9131            076346                  DT114=DT73
9132            076076                  DT115=DT4
9133            076032                  DT116=DT2
9134            076146                  DT117=DT6
9135            076346                  DT120=DT73
9136            076076                  DT121=DT4
9137            076032                  DT122=DT2
9138            076146                  DT123=DT6
9139            076346                  DT124=DT73
9140            076076                  DT125=DT4
9141            076032                  DT126=DT2
9142            076146                  DT127=DT6
9143            076346                  DT130=DT73
9144            076032                  DT131=DT2
9145            076146                  DT132=DT6
9146            076346                  DT133=DT73
9147            076032                  DT134=DT2
9148            076120                  DT135=DT5
9149            076120                  DT136=DT5
9150            076160                  DT137=DT16
9151            076120                  DT140=DT5
9152            076076                  DT141=DT4
9153            076076                  DT142=DT4
9154            076120                  DT143=DT5
9155            076076                  DT144=DT4
9156            076076                  DT145=DT4
9157            076120                  DT146=DT5
9158            076076                  DT147=DT4
9159            076076                  DT150=DT4
9160            076120                  DT151=DT5
9161            076076                  DT152=DT4
9162            076076                  DT153=DT4
```

```
 9163          076120                  DT154=DT5
 9164          076076                  DT155=DT4
 9165          076076                  DT156=DT4
 9166          076120                  DT157=DT5
 9167          076076                  DT160=DT4
 9168          076076                  DT161=DT4
 9169          076120                  DT162=DT5
 9170          076076                  DT163=DT4
 9171          076076                  DT164=DT4
 9172          076076                  DT215=DT4
 9173          076120                  DT216=DT5
 9174          076076                  DT217=DT4
 9175          076076                  DT220=DT4
 9176          076076                  DT221=DT4
 9177          076120                  DT222=DT5
 9178          076076                  DT223=DT4
 9179          076076                  DT224=DT4
 9180          076202                  DT165=DT37
 9181          076202                  DT166=DT37
 9182          076202                  DT167=DT37
 9183          076202                  DT170=DT37
 9184          076202                  DT171=DT37
 9185          076202                  DT172=DT37
 9186          076246                  DT173=DT41
 9187          076246                  DT174=DT41
 9188          076246                  DT175=DT41
 9189          076202                  DT200=DT37
 9190          076202                  DT201=DT37
 9191          076202                  DT202=DT37
 9192          076202                  DT203=DT37
 9193          076202                  DT204=DT37
 9194          076202                  DT205=DT37
 9195          076202                  DT206=DT37
 9196          076202                  DT207=DT37
 9197          076202                  DT210=DT37
 9198          076202                  DT211=DT37
 9199          076202                  DT212=DT37
 9200          076202                  DT213=DT37
 9201          076202                  DT214=DT37
 9202          076076                  DT225=DT4
 9203          076076                  DT226=DT4
 9204          076146                  DT227=DT6
 9205          076076                  DT230=DT4
 9206          076076                  DT231=DT4
 9207          076146                  DT232=DT6
 9208          076076                  DT233=DT4
 9209          076076                  DT234=DT4
 9210          076146                  DT235=DT6
 9211          076076                  DT236=DT4
 9212          076076                  DT237=DT4
 9213          076146                  DT240=DT6
 9214          076076                  DT241=DT4
 9215          076076                  DT242=DT4
 9216          076146                  DT243=DT6
 9217          076076                  DT244=DT4
 9218          076076                  DT245=DT4
 9219          076076                  DT246=DT4
```

```
 9220          076146                  DT247=DT6
 9221          076076                  DT250=DT4
 9222          076076                  DT251=DT4
 9223          076076                  DT252=DT4
 9224          076146                  DT253=DT6
 9225          076076                  DT254=DT4
 9226          076146                  DT255=DT6
 9227          076076                  DT256=DT4
 9228          076076                  DT257=DT4
 9229          076202                  DT260=DT37
 9230          076202                  DT261=DT37
 9231          076202                  DT262=DT37
 9232          076202                  DT263=DT37
 9233          076202                  DT264=DT37
 9234          076202                  DT265=DT37
 9235          076202                  DT266=DT37
 9236          076202                  DT267=DT37
 9237          076202                  DT270=DT37
 9238          076202                  DT271=DT37
 9239          076202                  DT272=DT37
 9240          076202                  DT273=DT37
 9241          076202                  DT274=DT37
 9242          076202                  DT275=DT37
 9243          076202                  DT276=DT37
 9244          076202                  DT277=DT37
 9245          076202                  DT300=DT37
 9246 076402  001232  001234  046435  DT301:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TAB,$TMP10
 9247 076422  001313  046477  001240          .WORD   $CRLF,MS10,$TMP3,$CRLF,MS11,$TMP4
 9248 076436  001313  046437  001246          .WORD   $CRLF,MS1,$TMP6,$CRLF,MS2,$TMP5,0
 9249          076402          DT302=DT301
 9250 076454  001232  001234  046435  DT303:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TMP11,$TMP12
 9251 076474  001313  046477  001240          .WORD   $CRLF,MS10,$TMP3,$CRLF,MS11,$TMP4
 9252 076510  001313  046437  001246          .WORD   $CRLF,MS1,$TMP6,$CRLF,MS2,$TMP5,0
 9253          076402          DT304=DT301
 9254          076402          DT305=DT301
 9255          076402          DT306=DT301
 9256          076402          DT307=DT301
 9257          076402          DT310=DT301
 9258          076402          DT311=DT301
 9259          076402          DT312=DT301
 9260          076402          DT313=DT301
 9261          076402          DT314=DT301
 9262          076402          DT315=DT301
 9263          076402          DT316=DT301
 9264          076402          DT317=DT301
 9265          076402          DT320=DT301
 9266          076402          DT321=DT301
 9267 076526  001232  001234  046435  DT322:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TAB,$TMP10
 9268 076546  001313  046477  001240          .WORD   $CRLF,MS10,$TMP3,$CRLF,MS1,$TMP4,$CRLF,MS2,$TMP5,0
 9269          076526          DT323=DT322
 9270 076572  001232  001234  046435  DT324:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TMP11,$TMP12
 9271 076612  001313  046477  001240          .WORD   $CRLF,MS10,$TMP3,$CRLF,MS1,$TMP4,$CRLF,MS2,$TMP5,0
 9272          076526          DT325=DT322
 9273          076526          DT326=DT322
 9274          076526          DT327=DT322
 9275          076526          DT330=DT322
 9276          076526          DT331=DT322
```

```
 9277            076526                          DT332=DT322
 9278            076526                          DT333=DT322
 9279            076526                          DT334=DT322
 9280            076526                          DT335=DT322
 9281            076526                          DT336=DT322
 9282            076526                          DT337=DT322
 9283            076526                          DT340=DT322
 9284            076526                          DT341=DT322
 9285            076526                          DT342=DT322
 9286            076526                          DT343=DT322
 9287            076526                          DT344=DT322
 9288            076526                          DT345=DT322
 9289            076526                          DT346=DT322
 9290            076526                          DT347=DT322
 9291            076526                          DT350=DT322
 9292            076146                          DT351=DT6
 9293            076526                          DT352=DT322
 9294            076526                          DT353=DT322
 9295            076526                          DT354=DT322
 9296            076526                          DT355=DT322
 9297            076032                          DT356=DT2
 9298            076054                          DT357=DT3
 9299            076146                          DT360=DT6
 9300            076402                          DT361=DT302
 9301 076636    001232  001234  046435  DT362:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,0
 9302 076654    001232  001234  046435  DT363:  .WORD   $TMP0,$TMP1,$TAB,$TMP13,$TAB,$TMP2,$TMP3,0
 9303 076674    001232  001234  046435  DT364:  .WORD   $TMP0,$TMP1,$TAB,$TMP13,0
 9304            076146                  DT365=DT6
 9305 076706    001232  001234  046435  DT366:  .WORD   $TMP0,$TMP1,$TAB,$TMP13,$TAB,$TMP2
 9306 076722    001313  046455  001240          .WORD   $CRLF,MS2,$TMP3,$TMP4,$TMP6,$TMP7
 9307 076736    001313  046437  001252          .WORD   $CRLF,MS1,$TMP10,$TMP11,$TMP12,$TMP21,0
 9308 076754    001232  001234  046435  DT367:  .WORD   $TMP0,$TMP1,$TAB,$TMP13,$TAB,$TMP6,$TAB,$TMP3,0
 9309 076776    001232  001234  046435  DT370:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TMP13
 9310 077010    001313  046574  001240          .WORD   $CRLF,MS21,$TMP3,$TMP4,$TMP6,$TMP7
 9311 077024    001313  046562  001252          .WORD   $CRLF,MS20,$TMP10,$TMP11,$TMP12,$TMP21,0
 9312            000000                  DT371=0
 9313            000000                  DT372=0
 9314            000000                  DT373=0
 9315            000000                  DT374=0
 9316            000000                  DT375=0
 9317            000000                  DT376=0
 9318            000000                  DT377=0
 9319            000000                  DT400=0
 9320
 9321            076076                  DT401=DT4
 9322            076076                  DT402=DT4
 9323            076146                  DT403=DT6
 9324            076146                  DT404=DT6
 9325            076076                  DT405=DT4
 9326            076076                  DT406=DT4
 9327            076146                  DT407=DT6
 9328            076146                  DT410=DT6
 9329            076076                  DT411=DT4
 9330            076076                  DT412=DT4
 9331            076146                  DT413=DT6
 9332            076146                  DT414=DT6
 9333            076076                  DT415=DT4
```

```
 9334            076076                  DT416=DT4
 9335            076146                  DT417=DT6
 9336            076146                  DT420=DT6
 9337            076076                  DT421=DT4
 9338            076076                  DT422=DT4
 9339            076146                  DT423=DT6
 9340            076146                  DT424=DT6
 9341            076076                  DT425=DT4
 9342            076076                  DT426=DT4
 9343            076146                  DT427=DT6
 9344            076146                  DT430=DT6
 9345            076146                  DT431=DT6
 9346            076076                  DT432=DT4
 9347            076076                  DT433=DT4
 9348            076146                  DT434=DT6
 9349            076146                  DT435=DT6
 9350            076146                  DT436=DT6
 9351            076076                  DT437=DT4
 9352            076076                  DT440=DT4
 9353 077042    001232  001234  046435  DT441:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,0
 9354 077060    001232  001234  046435  DT442:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,0
 9355            077060                  DT443=DT442
 9356
 9357
 9358
 9359
```

```
9361
9362                                    ;12345
9363        000001                              .END
```

SYMBOL TABLE

```
AABBF0  013602
AABDON  013632
AABTP1  013612
AABTP2  013622
AAB1    013424
AAB2    013526
AAB3    013546
AAB4    013564
AACDON  023754
AACTP1  023660
AAC1    023600
AAC10   023664
AAC11   023702
AAC2    023634
AAC20   023720
ABASE = 000000
ACDW1 = 000000
ACDW2 = 000000
ACPUOP= 000000
AC0   =%000000
AC1   =%000001
AC2   =%000002
AC3   =%000003
AC4   =%000004
AC5   =%000005
AC6   =%000006
AC7   =%000007
ADDW0 = 000000
ADDW1 = 000000
ADDW10= 000000
ADDW11= 000000
ADDW12= 000000
ADDW13= 000000
ADDW14= 000000
ADDW15= 000000
ADDW2 = 000000
ADDW3 = 000000
ADDW4 = 000000
ADDW5 = 000000
ADDW6 = 000000
ADDW7 = 000000
ADDW8 = 000000
ADDW9 = 000000
ADEVCT= 000000
ADEVM = 000000
AENV  = 000000
AENVM = 000000
AFATAL= 000000
AMADR1= 000000
AMADR2= 000000
AMADR3= 000000
AMADR4= 000000
AMAMS1= 000000
AMAMS2= 000000
AMAMS3= 000000
AMAMS4= 000000
AMSGAD= 000000
AMSGLG= 000000
AMSGTY= 000000
AMTYP1= 000000
AMTYP2= 000000
AMTYP3= 000000
AMTYP4= 000000
APASS = 000000
APRIOR= 000000
APTCSU= 000040
APTENV= 000001
APTSIZ= 000200
APTSPO= 000100
ASWREG= 000000
ATESTN= 000000
AUNIT = 000000
AUSWR = 000000
AVECT1= 000000
AVECT2= 000000
BBCDON  024132
BBCTP1  024040
BBC1    023760
BBC10   024044
BBC11   024062
BBC2    024014
BBC20   024100
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BPTVEC= 000014
CCBDON  013732
CCB1    013636
CCB10   013700
CCB15   013716
CCB2    013654
CKSWR = 104406
CNT   = 000444
CPSPUR  046250
CPTWO   046270
CR    = 000015
CRLF  = 000200
DATA  = 117760
DDBBF0  014230
DDBDON  014250
DDBTP1  014210
DDBTP2  014220
DDBTP3  014240
DDB1    014036
DDB2    014102
DDB5    014144
DDB6    014172
DDCDON  024326
DDCTP1  024226
DDC1    024136
DDC10   024240
DDC11   024256
DDC2    024174
DDC20   024274
DDISP = 177570
DF1     075354
DF10  = 075373
DF100 = 075363
DF101 = 075363
DF102 = 075405
DF103 = 075465
DF104 = 075363
DF105 = 075363
DF106 = 075405
DF107 = 075405
DF11  = 075363
DF110 = 075465
DF111 = 075363
DF112 = 075363
DF113 = 075405
DF114 = 075465
DF115 = 075363
DF116 = 075363
DF117 = 075405
DF12    075411
DF120 = 075465
DF121 = 075363
DF122 = 075363
DF123 = 075405
DF124 = 075465
DF125 = 075363
DF126 = 075363
DF127 = 075405
DF13  = 075405
DF130 = 075465
DF131 = 075363
DF132 = 075405
DF133 = 075465
DF134 = 075363
DF135 = 075411
DF136 = 075411
DF137 = 075363
DF14  = 075405
DF140 = 075411
DF141 = 075363
DF142 = 075363
DF143 = 075411
DF144 = 075363
DF145 = 075363
DF146 = 075411
DF147 = 075363
DF15  = 075411
DF150 = 075363
DF151 = 075411
DF152 = 075363
DF153 = 075363
DF154 = 075411
DF155 = 075363
DF156 = 075363
DF157 = 075411
DF16  = 075363
DF160 = 075363
DF161 = 075363
DF162 = 075411
DF163 = 075363
DF164 = 075363
DF165 = 075423
DF166 = 075423
DF167 = 075423
DF17  = 075405
DF170 = 075423
DF171 = 075423
DF172 = 075423
DF173 = 075444
DF174 = 075444
DF175 = 075444
DF176 = 075363
DF177 = 075363
DF2     075363
DF20  = 075363
DF200 = 075423
DF201 = 075423
DF202 = 075423
DF203 = 075423
DF204 = 075423
DF205 = 075423
DF206 = 075423
DF207 = 075423
DF21  = 075411
DF210 = 075423
DF211 = 075423
DF212 = 075423
DF213 = 075423
DF214 = 075423
DF215 = 075363
DF216 = 075411
DF217 = 075363
DF22  = 075411
DF220 = 075363
DF221 = 075363
DF222 = 075411
DF223 = 075363
DF224 = 075363
DF225   075502
DF226 = 075502
DF227   075512
DF23  = 075405
DF230 = 075502
DF231 = 075502
DF232 = 075512
DF233 = 075502
DF234 = 075502
DF235 = 075512
DF236 = 075502
DF237 = 075502
DF24  = 075363
DF240 = 075512
DF241 = 075502
DF242 = 075502
DF243 = 075512
DF244 = 075502
DF245 = 075502
DF246 = 075502
DF247 = 075512
DF25  = 075411
DF250 = 075502
DF251 = 075502
DF252 = 075502
DF253 = 075512
DF254 = 075502
DF255 = 075512
DF256 = 075502
DF257 = 075502
DF26  = 075405
DF260   075516
DF261 = 075516
DF262 = 075516
DF263 = 075516
DF264 = 075516
DF265 = 075516
DF266 = 075516
DF267 = 075516
DF27  = 075363
DF270 = 075516
DF271 = 075516
DF272 = 075516
DF273   075537
DF274 = 075537
DF275 = 075537
DF276 = 075537
DF277 = 075537
DF3   = 075363
```

```
DF30  = 075411
DF300 = 075537
DF301   075560
DF302 = 075560
DF303   075604
DF304 = 075560
DF305 = 075560
DF306 = 075560
DF307 = 075560
DF31  = 075405
DF310 = 075560
DF311 = 075560
DF312 = 075560
DF313 = 075560
DF314 = 075560
DF315 = 075560
DF316 = 075560
DF317 = 075560
DF32  = 075363
DF320 = 075560
DF321 = 075560
DF322   075630
DF323 = 075630
DF324   075651
DF325 = 075630
DF326 = 075630
DF327 = 075630
DF33  = 075411
DF330 = 075630
DF331 = 075630
DF332 = 075630
DF333 = 075630
DF334 = 075630
DF335 = 075630
DF336 = 075630
DF337 = 075630
DF34  = 075405
DF340 = 075630
DF341 = 075630
DF342 = 075630
DF343 = 075630
DF344 = 075630
DF345 = 075630
DF346 = 075630
DF347   075672
DF35  = 075363
DF350 = 075672
DF351 = 075512
DF352 = 075672
DF353 = 075672
DF354 = 075672
DF355 = 075672
DF356 = 075502
DF357 = 075502
DF36  = 075411
DF360 = 075512
DF361 = 075502
DF362   075713
DF363   075721
DF364 = 075405
DF365 = 075405
DF366   075730
DF367   075752
DF37    075423
DF370   075762
DF371 = 000000
DF372 = 000000
DF373 = 000000
DF374 = 000000
DF375 = 000000
DF376 = 000000
DF377 = 000000
DF4   = 075363
DF40  = 075423
DF400 = 000000
DF401 = 075502
DF402 = 075502
DF403 = 075512
DF404 = 075512
DF405 = 075502
DF406 = 075502
DF407 = 075512
DF41    075444
DF410 = 075512
DF411 = 075502
DF412 = 075502
DF413 = 075512
DF414 = 075512
DF415 = 075502
DF416 = 075502
DF417 = 075512
DF42  = 075423
DF420 = 075512
DF421 = 075502
DF422 = 075502
DF423 = 075512
DF424 = 075512
DF425 = 075502
DF426 = 075502
DF427 = 075512
DF43  = 075423
DF430 = 075512
DF431 = 075512
DF432 = 075502
DF433 = 075502
DF434 = 075512
DF435 = 075512
DF436 = 075512
DF437 = 075502
DF44  = 075423
DF440 = 075502
DF441   076003
DF442 = 076003
DF443 = 076003
DF45  = 075423
DF46  = 075423
DF47  = 075423
DF5     075373
DF50  = 075423
DF51  = 075423
DF52  = 075423
DF53  = 075423
DF54  = 075423
DF55  = 075423
DF56  = 075423
DF57  = 075423
DF6     075405
DF60  = 075423
DF61  = 075423
DF62  = 075363
DF63  = 075363
DF64  = 075373
DF65  = 075363
DF66  = 075363
DF67  = 075363
DF7   = 075363
DF70  = 075363
DF71    075465
DF72  = 075363
DF73  = 075465
DF74  = 075363
DF75  = 075363
DF76  = 075405
DF77  = 075465
DH1     073547
DH10  = 074071
DH100 = 074002
DH101 = 073622
DH102 = 074132
DH103 = 074071
DH104 = 074002
DH105 = 073622
DH106 = 074132
DH107 = 074132
DH11  = 074002
DH110 = 074071
DH111 = 074002
DH112 = 073622
DH113 = 074132
DH114 = 074071
DH115 = 074002
DH116 = 073622
DH117 = 074132
DH12  = 074071
DH120 = 074071
DH121 = 074002
DH122 = 073622
DH123 = 074132
DH124 = 074071
DH125 = 074002
DH126 = 073622
DH127 = 074132
DH13    074132
DH130 = 074071
DH131 = 073622
DH132 = 074132
DH133 = 074071
DH134 = 073622
DH135 = 074071
DH136 = 074071
DH137 = 073622
DH14  = 074132
DH140 = 074071
DH141 = 074002
DH142 = 073622
DH143 = 074071
DH144 = 074002
DH145 = 073622
DH146 = 074071
DH147 = 074002
DH15  = 074071
DH150 = 073622
DH151 = 074071
DH152 = 074002
DH153 = 073622
DH154 = 074071
DH155 = 074002
DH156 = 073622
DH157 = 074071
DH16    074172
DH160 = 074002
DH161 = 073622
DH162 = 074071
DH163 = 073622
DH164 = 074172
DH165 = 074261
DH166 = 074261
DH167 = 074261
DH17  = 074132
DH170 = 074261
DH171 = 074261
DH172 = 074261
DH173 = 074351
DH174 = 074351
DH175 = 074351
DH176 = 073622
DH177 = 073712
DH2     073622
DH20  = 074002
DH200 = 074261
DH201 = 074261
DH202 = 074261
DH203 = 074261
DH204 = 074261
DH205 = 074261
DH206 = 074261
DH207 = 074261
DH21  = 074071
DH210 = 074261
DH211 = 074261
DH212 = 074261
DH213 = 074261
DH214 = 074261
DH215 = 074172
DH216 = 074071
DH217 = 074002
DH22  = 074071
DH220 = 073622
DH221 = 074172
DH222 = 074071
DH223 = 074002
DH224 = 073622
DH225 = 074002
DH226 = 073622
DH227   074446
DH23  = 074132
DH230 = 074002
DH231 = 073622
DH232 = 074446
DH233 = 074002
DH234 = 073622
DH235 = 074446
DH236 = 074002
DH237 = 073622
DH24  = 074002
DH240 = 074446
DH241 = 074002
DH242 = 073622
DH243 = 074446
DH244 = 074002
DH245 = 073622
DH246 = 074172
DH247 = 074446
DH25  = 074071
DH250 = 074002
DH251 = 073622
DH252 = 074172
DH253 = 074446
DH254 = 074172
DH255 = 074446
DH256 = 074002
DH257 = 073622
DH26  = 074132
DH260 = 074261
DH261 = 074261
DH262 = 074261
DH263 = 074261
DH264 = 074261
DH265 = 074261
DH266 = 074261
DH267 = 074261
DH27  = 074002
DH270 = 074261
DH271 = 074261
DH272 = 074261
```

SYMBOL TABLE

```
DH273 = 074261        DH355 = 074261        DH437 = 074002        DT116 = 076032        DT20  = 076160
DH274 = 074261        DH356 = 074002        DH44  = 074261        DT117 = 076146        DT200 = 076202
DH275 = 074261        DH357   074506        DH440 = 074002        DT12  = 076120        DT201 = 076202
DH276 = 074261        DH36  = 074071        DH441   075245        DT120 = 076346        DT202 = 076202
DH277 = 074261        DH360 = 074132        DH442   075313        DT121 = 076076        DT203 = 076202
DH3     073712        DH361 = 073622        DH443 = 075313        DT122 = 076032        DT204 = 076202
DH30  = 074071        DH362   074576        DH45  = 074261        DT123 = 076146        DT205 = 076202
DH300 = 074261        DH363   074641        DH46  = 074261        DT124 = 076346        DT206 = 076202
DH301 = 074261        DH364   074716        DH47  = 074261        DT125 = 076076        DT207 = 076202
DH302 = 074261        DH365   075003        DH5     074071        DT126 = 076032        DT21  = 076120
DH303 = 074351        DH366   075041        DH50  = 074261        DT127 = 076146        DT210 = 076202
DH304 = 074261        DH367   075114        DH51  = 074261        DT13  = 076146        DT211 = 076202
DH305 = 074261        DH37    074261        DH52  = 074261        DT130 = 076346        DT212 = 076202
DH306 = 074261        DH370   075202        DH53  = 074351        DT131 = 076032        DT213 = 076202
DH307 = 074261        DH371 = 000000        DH54  = 074261        DT132 = 076146        DT214 = 076202
DH31  = 074132        DH372 = 000000        DH55  = 074261        DT133 = 076346        DT215 = 076076
DH310 = 074261        DH373 = 000000        DH56  = 074261        DT134 = 076032        DT216 = 076120
DH311 = 074261        DH374 = 000000        DH57  = 074261        DT135 = 076120        DT217 = 076076
DH312 = 074261        DH375 = 000000        DH6   = 074071        DT136 = 076120        DT22  = 076120
DH313 = 074261        DH376 = 000000        DH60  = 074261        DT137 = 076160        DT220 = 076076
DH314 = 074261        DH377 = 000000        DH61  = 074261        DT14  = 076146        DT221 = 076076
DH315 = 074261        DH4     074002        DH62  = 073622        DT140 = 076120        DT222 = 076120
DH316 = 074261        DH40  = 074261        DH63  = 073712        DT141 = 076076        DT223 = 076076
DH317 = 074261        DH400 = 000000        DH64  = 074071        DT142 = 076076        DT224 = 076076
DH32  = 074002        DH401 = 074002        DH65  = 073622        DT143 = 076120        DT225 = 076076
DH320 = 074261        DH402 = 073622        DH66  = 074002        DT144 = 076076        DT226 = 076076
DH321 = 074261        DH403 = 074132        DH67  = 073622        DT145 = 076076        DT227 = 076146
DH322 = 074261        DH404 = 074446        DH7   = 074002        DT146 = 076120        DT23  = 076146
DH323 = 074261        DH405 = 074002        DH70  = 073712        DT147 = 076076        DT230 = 076076
DH324 = 074351        DH406 = 073622        DH71  = 074071        DT15  = 076120        DT231 = 076076
DH325 = 074261        DH407 = 074132        DH72  = 073622        DT150 = 076076        DT232 = 076146
DH326 = 074261        DH41    074351        DH73  = 074071        DT151 = 076120        DT233 = 076076
DH327 = 074261        DH410 = 074446        DH74  = 074002        DT152 = 076076        DT234 = 076076
DH33  = 074071        DH411 = 074002        DH75  = 073622        DT153 = 076076        DT235 = 076146
DH330 = 074261        DH412 = 073622        DH76  = 074132        DT154 = 076120        DT236 = 076076
DH331 = 074261        DH413 = 074132        DH77  = 074071        DT155 = 076076        DT237 = 076076
DH332 = 074261        DH414 = 074446        DIDONE  042522        DT156 = 076076        DT24  = 076160
DH333 = 074261        DH415 = 074002        DISPLA  001142        DT157 = 076120        DT240 = 076146
DH334 = 074261        DH416 = 073622        DISPRE  000174        DT16    076160        DT241 = 076076
DH335 = 074261        DH417 = 074132        DSWR  = 177570        DT160 = 076076        DT242 = 076076
DH336 = 074261        DH42  = 074261        DT1     076012        DT161 = 076076        DT243 = 076146
DH337 = 074261        DH420 = 074446        DT10  = 076120        DT162 = 076120        DT244 = 076076
DH34  = 074132        DH421 = 074002        DT100 = 076076        DT163 = 076076        DT245 = 076076
DH340 = 074261        DH422 = 073622        DT101 = 076032        DT164 = 076076        DT246 = 076076
DH341 = 074261        DH423 = 074132        DT102 = 076146        DT165 = 076202        DT247 = 076146
DH342 = 074261        DH424 = 074446        DT103 = 076346        DT166 = 076202        DT25  = 076120
DH343 = 074261        DH425 = 074002        DT104 = 076076        DT167 = 076202        DT250 = 076076
DH344 = 074261        DH426 = 073622        DT105 = 076032        DT17  = 076146        DT251 = 076076
DH345 = 074261        DH427 = 074132        DT106 = 076146        DT170 = 076202        DT252 = 076076
DH346 = 074261        DH43  = 074261        DT107 = 076146        DT171 = 076202        DT253 = 076146
DH347 = 074261        DH430 = 074446        DT11  = 076076        DT172 = 076202        DT254 = 076076
DH35  = 074002        DH431 = 074132        DT110 = 076346        DT173 = 076246        DT255 = 076146
DH350 = 074261        DH432 = 074002        DT111 = 076076        DT174 = 076246        DT256 = 076076
DH351 = 074132        DH433 = 073622        DT112 = 076032        DT175 = 076246        DT257 = 076076
DH352 = 074261        DH434 = 074132        DT113 = 076146        DT176 = 076076        DT26  = 076146
DH353 = 074261        DH435 = 074446        DT114 = 076346        DT177 = 076076        DT260 = 076202
DH354 = 074261        DH436 = 074132        DT115 = 076076        DT2     076032        DT261 = 076202
```

```
DT262 = 076202
DT263 = 076202
DT264 = 076202
DT265 = 076202
DT266 = 076202
DT267 = 076202
DT27  = 076160
DT270 = 076202
DT271 = 076202
DT272 = 076202
DT273 = 076202
DT274 = 076202
DT275 = 076202
DT276 = 076202
DT277 = 076202
DT3   = 076054
DT30  = 076120
DT300 = 076202
DT301   076402
DT302 = 076402
DT303   076454
DT304 = 076402
DT305 = 076402
DT306 = 076402
DT307 = 076402
DT31  = 076146
DT310 = 076402
DT311 = 076402
DT312 = 076402
DT313 = 076402
DT314 = 076402
DT315 = 076402
DT316 = 076402
DT317 = 076402
DT32  = 076160
DT320 = 076402
DT321 = 076402
DT322   076526
DT323 = 076526
DT324   076572
DT325 = 076526
DT326 = 076526
DT327 = 076526
DT33  = 076120
DT330 = 076526
DT331 = 076526
DT332 = 076526
DT333 = 076526
DT334 = 076526
DT335 = 076526
DT336 = 076526
DT337 = 076526
DT34  = 076146
DT340 = 076526
DT341 = 076526
DT342 = 076526
DT343 = 076526
DT344 = 076526
DT345 = 076526
DT346 = 076526
DT347 = 076526
DT35  = 076160
DT350 = 076526
DT351 = 076146
DT352 = 076526
DT353 = 076526
DT354 = 076526
DT355 = 076526
DT356 = 076032
DT357 = 076054
DT36  = 076120
DT360 = 076146
DT361 = 076402
DT362   076636
DT363   076654
DT364   076674
DT365 = 076146
DT366   076706
DT367   076754
DT37    076202
DT370   076776
DT371 = 000000
DT372 = 000000
DT373 = 000000
DT374 = 000000
DT375 = 000000
DT376 = 000000
DT377 = 000000
DT4     076076
DT40  = 076202
DT400 = 000000
DT401 = 076076
DT402 = 076076
DT403 = 076146
DT404 = 076146
DT405 = 076076
DT406 = 076076
DT407 = 076146
DT41    076246
DT410 = 076146
DT411 = 076076
DT412 = 076076
DT413 = 076146
DT414 = 076146
DT415 = 076076
DT416 = 076076
DT417 = 076146
DT42  = 076202
DT420 = 076146
DT421 = 076076
DT422 = 076076
DT423 = 076146
DT424 = 076146
DT425 = 076076
DT426 = 076076
DT427 = 076146
DT43  = 076202
DT430 = 076146
DT431 = 076146
DT432 = 076076
DT433 = 076076
DT434 = 076146
DT435 = 076146
DT436 = 076146
DT437 = 076076
DT44  = 076202
DT440 = 076076
DT441   077042
DT442   077060
DT443 = 077060
DT45  = 076202
DT46  = 076202
DT47  = 076202
DT5     076120
DT50  = 076202
DT51  = 076202
DT52  = 076202
DT53  = 076246
DT54  = 076202
DT55  = 076202
DT56  = 076202
DT57  = 076202
DT6     076146
DT60  = 076202
DT61  = 076202
DT62  = 076160
DT63  = 076160
DT64  = 076120
DT65  = 076160
DT66  = 076076
DT67  = 076076
DT7   = 076076
DT70  = 076076
DT71    076312
DT72  = 076076
DT73    076346
DT74  = 076076
DT75  = 076032
DT76  = 076146
DT77  = 076346
EEBBF0  014404
EEBBF1  014414
EEBDON  014540
EEBTP1  014364
EEBTP2  014374
EEB1    014254
EEB10   014424
EEB15   014460
EEB2    014324
EEB20   014506
EEB25   014524
EECDON  024536
EECTP1  024426
EECTP2  024436
EEC1    024332
EEC10   024450
EEC11   024466
EEC2    024374
EEC20   024504
EMTVEC= 000030
EM1     046614
EM10    047170
EM100   053107
EM101   053132
EM102   053156
EM103   053217
EM104   053241
EM105   053264
EM106   053310
EM107   052676
EM11    047213
EM110   053352
EM111   053375
EM112   053421
EM113   053446
EM114   053510
EM115   053533
EM116   053557
EM117   053604
EM12    047254
EM120   053645
EM121   053667
EM122   053712
EM123   053736
EM124   054000
EM125   054023
EM126   054047
EM127   054074
EM13    047277
EM130   054136
EM131   054161
EM132   054206
EM133   054251
EM134   054275
EM135   054323
EM136   054376
EM137   054415
EM14  = 047277
EM140   054436
EM141   054457
EM142   054526
EM143   054551
EM144   054573
EM145   054643
EM146   054667
EM147   054711
EM15    047333
EM150   054761
EM151   055005
EM152   055030
EM153   055101
EM154   055126
EM155   055151
EM156   055222
EM157   055247
EM16    047354
EM160   055271
EM161   055363
EM162   055407
EM163   055432
EM164   055457
EM165   056255
EM166   056276
EM167   056317
EM17    047403
EM170   056340
EM171   056363
EM172   056406
EM173   056431
EM174   056454
EM175   056477
EM176   052570
EM177   052613
EM2     046653
EM20    047441
EM200   056522
EM201   056577
EM202   056700
EM203   057001
EM204   057161
EM205   057236
EM206   057335
EM207   057436
EM21    047502
EM210   057535
EM211   057634
EM212   057742
EM213   060043
EM214   060170
EM215   055533
EM216   055664
EM217   055706
EM22  = 047502
EM220   055756
EM221   056002
EM222   056134
EM223   056157
EM224   056230
EM225   060315
EM226   060340
EM227   060364
EM23    047525
EM230   060414
EM231   060440
EM232   060465
```

SYMBOL TABLE

```
EM233   060516      EM315   064444      EM4     046741      EM62    052154      GGB20   015266
EM234   060542      EM316   064545      EM40    050272      EM63    052260      GGB25   015304
EM235   060567      EM317   064646      EM400 = 000000      EM64    052306      GGCDON  025156
EM236   060620      EM32    050035      EM401   070503      EM65    052362      GGCTP1  025036
EM237   060645      EM320   064747      EM402   070526      EM66    052405      GGC1    024744
EM24    047564      EM321   065050      EM403   070550      EM67    052444      GGC10   025050
EM240   060673      EM322   065151      EM404   070702      EM7     047127      GGC11   025066
EM241   060725      EM323   065206      EM405   070732      EM70    052545      GGC2    025004
EM242   060752      EM324   065245      EM406   070756      EM71    052636      GGC20   025124
EM243   061000      EM325   065304      EM407   071001      EM72    052655      GGC25   025104
EM244   061032      EM326 = 065304      EM41    050320      EM73    052736      GTSWR = 104405
EM245   061056      EM327   065445      EM410   071134      EM74    052757      HHBBF0  015464
EM246   061103      EM33    050076      EM411   071165      EM75    053001      HHBBF1  015474
EM247   061134      EM330   065547      EM412   071211      EM76    053024      HHBDON  015620
EM25    047626      EM331   065652      EM413   071234      EM77    053065      HHBTP1  015434
EM250   061165      EM332   067126      EM414   071367      ENDTES  042502      HHBTP2  015454
EM251   061212      EM333 = 065206      EM415   071420      ENDTST  040000      HHB1    015324
EM252   061240      EM334   065755      EM416   071445      ERM10   043532      HHB10   015504
EM253   061272      EM335   066051      EM417   071471      ERROR = 104000      HHB15   015540
EM254   061324      EM336   066153      EM42    050346      ERRVEC= 000004      HHB2    015374
EM255   061360      EM337   066227      EM420   071537      ERTYPE  045564      HHB20   015566
EM256   061414      EM34    050121      EM421   071571      ERT1    045744      HHB25   015604
EM257   061442      EM340   066331      EM422   071616      ERT2    046162      HHCDON  025410
EM26    047652      EM341   066433      EM423   071642      ERT3    046166      HHCTP1  025262
EM260   061471      EM342   066537      EM424   071710      ERT4    046176      HHCTP2  025272
EM261   061526      EM343   066641      EM425   071742      ERT5    046210      HHC1    025162
EM262   061565      EM344   066743      EM426   071766      FALTRP  041156      HHC10   025302
EM263   061665      EM345   067220      EM427   072011      FFBBF0  014674      HHC11   025320
EM264   061713      EM346   067320      EM43    050425      FFBBF1  014704      HHC2    025230
EM265   062010      EM347   067416      EM430   072144      FFBDON  015030      HHC20   025356
EM266   062101      EM35    050160      EM431   072175      FFBTP1  014654      HHC25   025336
EM267   062214      EM350   067442      EM432   072250      FFBTP2  014664      HT    = 000011
EM27    047711      EM351   067470      EM433   072275      FFB1    014544      IDONE   040012
EM270   062311      EM352   067574      EM434   072321      FFB10   014714      IIBBF0  015764
EM271   062352      EM353   067700      EM435   072455      FFB15   014750      IIBBF1  015774
EM272   062420      EM354   070004      EM436   072507      FFB2    014614      IIBDON  016120
EM273   062511      EM355   070110      EM437   072564      FFB20   014776      IIBTP1  015734
EM274   062546      EM356   070214      EM44    050531      FFB25   015014      IIBTP2  015754
EM275   062605      EM357   070312      EM440   072612      FFCDON  024740      IIB1    015624
EM276   062705      EM36    050222      EM441   073426      FFCTP1  024632      IIB10   016004
EM277   063002      EM360   070410      EM442   073462      FFCTP2  024642      IIB15   016040
EM3     046706      EM361   072640      EM443   073514      FFC1    024542      IIB2    015674
EM30    047753      EM362   072663      EM45    050631      FFC10   024652      IIB20   016066
EM300   063056      EM363   072773      EM46    050707      FFC11   024670      IIB25   016104
EM301   063153      EM364   073041      EM47    051013      FFC2    024606      IICDON  025524
EM302   063177      EM365   073141      EM5     047001      FFC20   024706      IIC1    025414
EM303   063225      EM366   073224      EM50    051113      FPSPUR  046214      IIC2    025442
EM304   063253      EM367   073307      EM51    051227      FPVECT= 000244      IIC20   025514
EM305   063342      EM37    050246      EM52    051253      GGBBF0  015164      IIC3    025462
EM306   063445      EM370   073372      EM53    051277      GGBBF1  015174      INCDCT  040016
EM307   063632      EM371 = 000000      EM54    051323      GGBDON  015320      IOTRAP= 000020
EM31    047777      EM372 = 000000      EM55    051402      GGBTP1  015144      IOTVEC= 000020
EM310   063734      EM373 = 000000      EM56    051530      GGBTP2  015154      JJBBF0  016260
EM311   064037      EM374 = 000000      EM57    051632      GGB1    015034      JJBBF1  016270
EM312   064140      EM375 = 000000      EM6     047023      GGB10   015204      JJBDON  016414
EM313   064242      EM376 = 000000      EM60    051742      GGB15   015240      JJBTP1  016236
EM314   064343      EM377 = 000000      EM61    052052      GGB2    015104      JJBTP2  016250
```

SYMBOL TABLE

```
JJB1     016124          KKC5     026056          MMC5     030536          OOC10    032462          QQB10    020424
JJB10    016300          KKC6     026114          MMC6     030614          OOC15    032502          QQB15    020444
JJB15    016334          KKC7     026152          MMC7     030672          OOC2     032414          QQB2     020330
JJB2     016174          KKC8     026210          MMC8     030750          OOC20    032524          QQB20    020462
JJB20    016362          KKC9     026246          MMC9     031026          OOC25    032546          QQCDON   033270
JJB25    016400          LABEL1   040524          MMR0   = 177572          OOODON   006726          QQCTB0   033136
KDPAR0=  172360          LDCDSU   030036          MMR2   = 177576          OOOT     006636          QQCTB1   033142
KDPAR1=  172362          LDCFSU   027020          MMR3   = 172516          OOO1     006572          QQCTB2   033152
KDPAR2=  172364          LDCT     027224          MMVECT=  000250          OOO2     006616          QQC1     033032
KDPAR3=  172366          LDXSUB   031610          MNUMBE=  000443          OOO3     006652          QQC10    033156
KDPAR4=  172370          LDXT     032100          MODE1    040472          OOO4     006720          QQC15    033176
KDPAR7=  172376          LF     = 000012          MS1      046437          PIRQ   = 177772          QQC2     033104
KDPDR0=  172320          LLBBF0   017042          MS10   = 046477          PIRQVE=  000240          QQC20    033220
KDPDR1=  172322          LLBBF1   017052          MS11     046537          POWERM   046366          QQC25    033242
KDPDR2=  172324          LLBDON   017160          MS2      046455          PPBDON   020254          QQQBF0   007332
KDPDR3=  172326          LLBTP1   017022          MS20     046562          PPBTP1   020162          QQQBF1   007346
KDPDR4=  172330          LLBTP2   017032          MS21     046574          PPBTP2   020172          QQQDON   007550
KDPDR7=  172336          LLB1     016722          MS3      046477          PPB1     020050          QQQTP1   007362
KIPAR0=  172340          LLB10    017062          MS4      046515          PPB10    020202          QQQ1     007172
KIPAR1=  172342          LLB15    017116          NATBF1   023562          PPB15    020222          QQQ10    007372
KIPAR2=  172344          LLB2     016766          NATER1   023512          PPB2     020116          QQQ2     007210
KIPAR3=  172346          LLB25    017144          NATER2   023530          PPB20    020240          QQQ20    007412
KIPAR4=  172350          LLCDON   030242          NATER3   023544          PPCDON   033026          QQQ22    007426
KIPAR7=  172356          LLC1     027240          NATINS   023240          PPCTB0   032700          QQQ23    007460
KIPDR0=  172300          LLC10    027766          NATRET   023524          PPCTB1   032704          QQQ24    007506
KIPDR1=  172302          LLC2     027306          NATSUB   023160          PPC1     032600          QQQ25    007532
KIPDR2=  172304          LLC3     027354          NNBBF0   017554          PPC10    032714          QQQ3     007246
KIPDR3=  172306          LLC4     027422          NNBDON   017634          PPC15    032734          QQQ4     007274
KIPDR4=  172310          LLC5     027470          NNBTP1   017534          PPC2     032646          RDCHR  = 104407
KIPDR7=  172316          LLC6     027536          NNBTP2   017544          PPC20    032756          RESREG=  104411
KKBBF0   016562          LLC7     027604          NNB1     017436          PPC25    033000          RESVEC=  000010
KKBBF1   016572          LLC8     027652          NNB10    017564          PPPBF0   007070          RRBDON   020716
KKBDON   016716          LLC9     027720          NNB11    017614          PPPBF1   007104          RRBTP1   020622
KKBTP1   016532          LOOP     006570          NNB15    017620          PPPDON   007166          RRBTP2   020632
KKBTP2   016552          LPERR  = 104413          NNB2     017462          PPPTP1   007120          RRBTP3   020642
KKB1     016420          MMBBF0   017314          NNCDON   032342          PPP1     006732          RRB1     020502
KKB10    016602          MMBBF1   017324          NNCTB0   032214          PPP10    007130          RRB10    020644
KKB15    016636          MMBDON   017432          NNCTB1   032220          PPP15    007150          RRB15    020664
KKB2     016470          MMBTP1   017264          NNC1     032114          PPP2     006750          RRB2     020556
KKB20    016664          MMBTP2   017304          NNC10    032230          PPP3     007012          RRB20    020702
KKB25    016702          MMB1     017164          NNC15    032250          PPP4     007032          RRCDON   033534
KKCDON   027234          MMB10    017334          NNC2     032162          PROGNU=  000003          RRCTB0   033402
KKC1     025666          MMB15    017370          NNC20    032272          PR0    = 000000          RRCTB1   033406
KKC10    026304          MMB2     017230          NNC25    032314          PR1    = 000040          RRCTB2   033416
KKC11    026342          MMB25    017416          NODAT    042406          PR2    = 000100          RRC1     033274
KKC12    026400          MMCDON   032110          OOBDON   020044          PR3    = 000140          RRC10    033422
KKC13    026436          MMC1     030246          OOBTP1   017752          PR4    = 000200          RRC15    033442
KKC14    026474          MMC10    031104          OOBTP2   017762          PR5    = 000240          RRC2     033350
KKC15    026532          MMC11    031162          OOB1     017640          PR6    = 000300          RRC20    033464
KKC16    026570          MMC12    031240          OOB10    017772          PR7    = 000340          RRC25    033506
KKC17    026626          MMC13    031316          OOB15    020012          PS     = 177776          RRRDON   010072
KKC18    026664          MMC14    031374          OOB2     017706          PSW    = 177776          RRREXP   007722
KKC19    026722          MMC15    031452          OOB20    020030          PWRVEC=  000024          RRRTP1   007702
KKC2     025724          MMC16    031530          OOCDON   032574          QQBDON   020476          RRRTP2   007712
KKC20    026760          MMC2     030324          OOCTB0   032446          QQBTP1   020374          RRR1     007554
KKC3     025762          MMC3     030402          OOCTB1   032452          QQBTP2   020414          RRR10    007732
KKC4     026020          MMC4     030460          OOC1     032346          QQB1     020260          RRR11    007754
```

SYMBOL TABLE

```
RRR12   007774          SWREG   000176          TST25   015622          TTCTB0  034132          VVB15   014016
RRR15   010026          SW0   = 000001          TST26   016122          TTCTB1  034136          VVB2    013754
RRR2    007632          SW00  = 000001          TST27   016416          TTCTB2  034146          VVCBF0  034474
RRR25   010006          SW01  = 000002          TST3    007170          TTC1    034012          VVCDON  034502
RRR3    007634          SW02  = 000004          TST30   016720          TTC10   034152          VVCTP1  034462
RRR4    007660          SW03  = 000010          TST31   017162          TTC15   034172          VVC1    034402
RSETUP= 104412          SW04  = 000020          TST32   017434          TTC2    034070          VVC2    034432
R6    =%000006          SW05  = 000040          TST33   017636          TTC20   034214          VVC3    034442
R7    =%000007          SW06  = 000100          TST34   020046          TTC25   034236          VVVBF0  011246
SAVREG= 104410          SW07  = 000200          TST35   020256          TTC30   034266          VVVDON  011356
SCOPE = 000004          SW08  = 000400          TST36   020500          TTTA1   010540          VVVTP1  011256
SPACE   046432          SW09  = 001000          TST37   020720          TTTA2   010542          VVV1    011126
SR1   = 177574          SW1   = 000002          TST4    007552          TTTA3   010544          VVV10   011266
SSBDON  021140          SW10  = 002000          TST40   021142          TTTBF0  010526          VVV11   011310
SSBTP1  021044          SW11  = 004000          TST41   021352          TTTDON  010652          VVV15   011322
SSBTP2  021054          SW12  = 010000          TST42   021600          TTTTP1  010552          VVV2    011210
SSBTP3  021064          SW13  = 020000          TST43   022042          TTT1    010406          VVV20   011340
SSB1    020722          SW14  = 040000          TST44   022314          TTT10   010562          WWBDON  023574
SSB10   021066          SW15  = 100000          TST45   023576          TTT11   010604          WWB1    022316
SSB15   021106          SW2   = 000004          TST46   023756          TTT15   010616          WWB2    022374
SSB2    021000          SW3   = 000010          TST47   024134          TTT2    010472          WWB3    022452
SSB20   021124          SW4   = 000020          TST5    010074          TTT20   010634          WWB4    022530
SSCDON  034006          SW5   = 000040          TST50   024330          TTT3    010516          WWB5    022606
SSCTB0  033652          SW6   = 000100          TST51   024540          TYPE  = 104401          WWB6    022664
SSCTB1  033656          SW7   = 000200          TST52   024742          TYPOC = 104402          WWB7    022742
SSC1    033540          SW8   = 000400          TST53   025160          TYPON = 104404          WWB8    023020
SSC10   033666          SW9   = 001000          TST54   025412          TYPOS = 104403          WWB9    023076
SSC15   033706          TAB   = 000011          TST55   025526          UUBDON  021576          WWCDON  036144
SSC2    033610          TBITVE= 000014          TST56   025664          UUBTP1  021430          WWC1    034506
SSC20   033730          TCCBF0  025642          TST57   027236          UUBTP2  021502          WWC10   035146
SSC25   033752          TCCBF1  025652          TST6    010404          UUB1    021354          WWC11   035212
SSC30   034002          TCCDON  025662          TST60   030244          UUB10   021512          WWC12   035256
SSSA1   010244          TCC1    025530          TST61   032112          UUB15   021550          WWC13   035322
SSSBF0  010234          TCC2    025552          TST62   032344          UUB2    021426          WWC14   035366
SSSDON  010402          TCC3    025616          TST63   032576          UUB20   021532          WWC15   035432
SSSTP1  010254          TKVEC = 000060          TST64   033030          UUCBF0  034370          WWC16   035476
SSSTP2  010264          TPVEC = 000064          TST65   033272          UUCDON  034376          WWC17   035542
SSS1    010076          TRAPV   037716          TST66   033536          UUCTP1  034356          WWC18   035606
SSS10   010274          TRAPVE= 000034          TST67   034010          UUC1    034276          WWC2    034552
SSS11   010316          TRPV    041242          TST7    010654          UUC2    034326          WWC4    034616
SSS15   010326          TRTVEC= 000014          TST70   034274          UUC3    034336          WWC5    034662
SSS2    010154          TST1    006570          TST71   034400          UUUA1   011010          WWC6    034726
SSS20   010344          TST10   011124          TST72   034504          UUUA2   011012          WWC7    034772
SSS25   010364          TST100  042524          TST73   036146          UUUA3   011014          WWC8    035036
STACK = 001100          TST11   011360          TST74   036216          UUUBF0  010776          WWC9    035102
START   006106          TST12   011632          TST75   036730          UUUDON  011122          WWWBF0  011510
STCDFS  013026          TST13   012466          TST76   037134          UUUTP1  011022          WWWBF1  011530
STCDT   013310          TST14   013322          TST77   040014          UUU1    010656          WWWDON  011630
STCFDS  012172          TST15   013422          TTBDON  021350          UUU10   011032          WWWTP1  011520
STCFT   012454          TST16   013634          TTBTP1  021256          UUU11   011054          WWW1    011362
STCIBF  036134          TST17   013734          TTBTP2  021266          UUU15   011066          WWW10   011540
STCSUB  035654          TST2    006730          TTB1    021144          UUU2    010742          WWW11   011562
STKLMT= 177774          TST20   014034          TTB10   021276          UUU20   011104          WWW15   011574
STORE   042422          TST21   014252          TTB15   021316          UUU3    010766          WWW2    011452
STXBF   036714          TST22   014542          TTB2    021212          VVBDON  014032          WWW20   011612
STXSUB  036506          TST23   015032          TTB20   021334          VVB1    013736          XXBDON  022040
SWR     001140          TST24   015322          TTCDON  034272          VVB10   014000          XXBTP1  021726
```

SYMBOL TABLE

```
XXBTP2  021736        ZZZ15   013404        $EOPCT  042552        $NWTST= 000001        $TESTN  001322
XXB1    021602        ZZZ2    013342        $ERFLG  001103        $OCNT   044404        $TIMES  001302
XXB10   021766        $APTHD  006072        $ERMAX  001115        $OMODE  044406        $TKB    001146
XXB15   022006        $ATYC   044434        $ERROR  043276        $OVER   043260        $TKS    001144
XXB2    021654        $ATY1   044410        $ERRPC  001116        $PASS   001324        $TMP0   001232
XXB20   022024        $ATY3   044416        $ERRTB  001442        $PASTM  006100        $TMP1   001234
XXB25   021746        $ATY4   044426        $ERTTL  001112        $PWRAD  045540        $TMP10  001252
XXCDON  036214        $AUTOB  001134        $ESCAP  001304        $PWRDN  045400        $TMP11  001254
XXC1    036150        $BASE   001372        $ETABL  001336        $PWRMG  045534        $TMP12  001256
XXX ON  012464        $BDADR  001122        $ETEND  001442        $PWRUP  045452        $TMP13  001260
XXX1    011634        $BDDAT  001126        $FATAL  001320        $QUES   001312        $TMP14  001262
XXX2    011710        $BELL   001306        $FFLG   044654        $RDCHR  045140        $TMP15  001264
XXX3    011764        $CDW1   001376        $FILLC  001156        $RDSZ = 000001        $TMP16  001266
XXX4    012040        $CDW2   001400        $FILLS  001155        $REGAD  001160        $TMP17  001270
XXX5    012114        $CHARC  044156        $GDADR  001120        $REG0   001162        $TMP2   001236
YYBDON  022312        $CKSWR  044656        $GDDAT  001124        $REG1   001164        $TMP20  001272
YYBTP1  022176        $CLR.T  042726        $GET42  042710        $REG10  001202        $TMP21  001274
YYBTP2  022206        $CMTAG  001100        $GTSWR  044726        $REG11  001204        $TMP22  001276
YYBTP3  022216        $CM1  = 000024        $HD   = 000003        $REG12  001206        $TMP23  001300
YYB1    022044        $CM2  = 000050        $HIBTS  006072        $REG13  001210        $TMP3   001240
YYB10   022240        $CM3  = 000024        $ICNT   001104        $REG14  001212        $TMP4   001242
YYB15   022260        $CM4  = 000024        $ILLUP  045556        $REG15  001214        $TMP5   001244
YYB2    022124        $CNTLG  045265        $INTAG  001135        $REG16  001216        $TMP6   001246
YYB20   022276        $CNTLU  045260        $ITEMB  001114        $REG17  001220        $TMP7   001250
YYB25   022220        $CPUOP  001344        $LF     001314        $REG2   001166        $TN   = 000100
YYCDON  036726        $CRLF   001313        $LFLG   044653        $REG20  001222        $TPB    001152
YYC1    036220        $DDW0   001402        $LOOP   043004        $REG21  001224        $TPFLG  001157
YYC2    036256        $DDW1   001404        $LPADR  001106        $REG22  001226        $TPS    001150
YYC3    036314        $DDW10  001426        $LPERR  001110        $REG23  001230        $TRAP   045314
YYC4    036352        $DDW11  001430        $MADR1  001350        $REG3   001170        $TRAP2  045336
YYC5    036410        $DDW12  001432        $MADR2  001354        $REG4   001172        $TRP  = 000014
YYC6    036446        $DDW13  001434        $MADR3  001360        $REG5   001174        $TRPAD  045350
YYYDON  013320        $DDW14  001436        $MADR4  001364        $REG6   001176        $TSTM   006076
YYY1    012470        $DDW15  001440        $MAIL   001316        $REG7   001200        $TSTNM  001102
YYY2    012544        $DDW2   001406        $MAMS1  001346        $RESRE  043572        $TYPE   043630
YYY3    012620        $DDW3   001410        $MAMS2  001352        $RTNAD  043006        $TYPEC  044042
YYY4    012674        $DDW4   001412        $MAMS3  001356        $RTRN   043002        $TYPEX  044160
YYY5    012750        $DDW5   001414        $MAMS4  001362        $SAVRE  043534        $TYPOC  044206
ZZCBF   037120        $DDW6   001416        $MBADR  006074        $SAVR6  045562        $TYPON  044222
ZZCDON  037132        $DDW7   001420        $MFLG   044652        $SCOPE  043016        $TYPOS  044162
ZZC1    036732        $DDW8   001422        $MNEW   045303        $SETUP= 000137        $UNIT   001330
ZZC10   037072        $DDW9   001424        $MSGAD  001332        $STUP = 177777        $UNITM  006102
ZZC12   037106        $DEVCT  001326        $MSGLG  001334        $SVLAD  043224        $USWR   001342
ZZC15   037112        $DEVM   001374        $MSGTY  001316        $SVPC = 006072        $VECT1  001366
ZZC2    036742        $DOAGN  042746        $MSWR   045272        $SWR  = 177400        $VECT2  001370
ZZC3    036770        $ENDAD  042736        $MTYP1  001347        $SWREG  001340        $XTSTR  043030
ZZC5    037066        $ENDCT  042560        $MTYP2  001353        $SWRMK= 000000        $$GET4= 000001
ZZF1    037136        $ENULL  043012        $MTYP3  001357        $SWRMS= 000200        $OFILL  044405
ZZZDON  013420        $ENV    001336        $MTYP4  001363        $TAB    046435        .LPER   046310
ZZZ1    013324        $ENVM   001337        $MXCNT  043274        $TBIT   043010        .RSET   046316
ZZZ10   013366        $EOP    042524        $NULL   001154        $TERM = 000030        .$X   = 006072

. ABS.  077072      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  58592 WORDS  ( 229 PAGES)
```

DYNAMIC MEMORY: 20746 WORDS (79 PAGES)

SEQ 0238