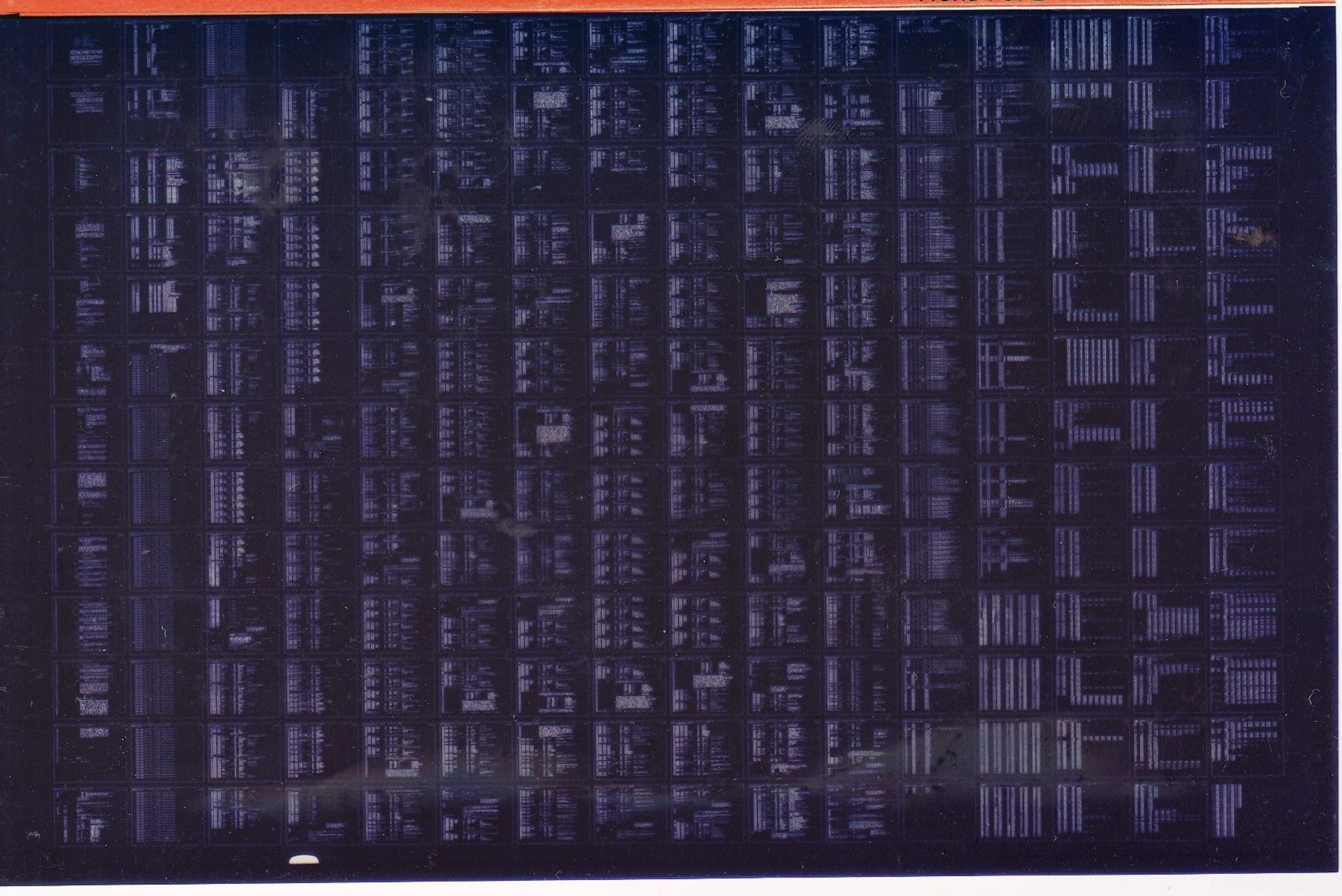
FP11-F

FP11F FLTG PNT B
CKFPBA0

AH-F635A-MC

FICHE 1 OF 2

JAN 1980
digital
MADE IN USA



FP11-F

FP11F FLTG PNT B
CKFPBA0

AH-F635A-MC

FICHE 2 OF 2

JAN 1980
digital
MADE IN USA

```
                .REM    &
```

IDENTIFICATION

PRODUCT CODE: AC-F633A-MC

PRODUCT NAME: CKFPBA0 FP11F FLTG PNT PRT B

DATE CREATED: OCTOBER, 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: ANTHONY VEZZA, DAN MILLEVILLE

```
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
```

HISTORY

NO CHANGES TO THE 11/34 FLOATING POINT DIAGNOSTICS PART 'A' WERE FOUND TO BE NEEDE 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 VARIFY ALL THE LOGIC. THEY CONSIST OF 157 (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 CKFPBA0)
- 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 ABSD
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

2. REQUIREMENTS

2.1 EQUIPMENT

A PDP 11/44 (WITH OR WITHOUT CONSOLE), LA30 (OR EQUIVALENT) AND AN FP11-F FLOATING POINT PROCESSOR. NOTE THAT A SPECIAL INTERRUPTS TEST MODULE IS BEING DESIGNED FOR USE IN THE MANUFACTURING 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 11/44 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). IF 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. IF APT IS SELECTING THE TEST, THE SWITCH REGISTER IS EXAMINED TO SEE IF THE USER HAS SELECTED THIS TEST BY A <1> IN SW<7>

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 IF 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.

6.2 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

EACH OF THE THREE PROGRAMS WILL RUN WITH OR WITHOUT A CONSOLE SWITCH REGISTER. IF A PHYSICAL CONSOLE SWITCH REGISTER IS PRESENT ON THE SYSTEM, THEN THESE PROGRAMS WILL GO AHEAD AND USE IT FOR THE SWITCH

FUNCTIONS DESCRIBED IN 5.1 ABOVE. IF HOWEVER THERE IS NO CONSOLE SWITCH REGISTER ON THE SYSTEM A SOFTWARE SWITCH REGISTER WILL BE USED. THIS SOFTWARE SWITCH REGISTER CAN BE EXAMINED OR MODIFIED AT ANY TIME BY THE USER IF HE TYPES CONTROL G WHILE THE PROGRAM IS RUNNING. THIS CONTROL 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 CONTROL 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 NO CONSOLE SWITCH REGISTER IS ON THE SYSTEM).

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>=0 THIS TEST WILL BE RUN. IF SW<7>=1 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 ROUND\TRUNK TEST

```
                        THIS  IS  A  TEST  OF  THE  ROUND\TRUNK  FLOWS.   IN
                        PARTICULAR TWO THINGS ARE TESTED:  FIRST A CONDITION
                        IN  WHICH  ROUNDING  RESULTS  IN  THE  NEED  FOR
                        RENORMALIZATION,  AND SECOND THE PSW CONDITION CODES
                        N AND Z BIT COMMBINATIONS

TEST 2          OVER\UNDER TEST
------          ---------------

                        THIS IS A PARTIAL TEST OF THE OVER\UNDER FLOWS.  ONE
                        OVERFLOW  AND  TWO UNDERFLOW CONDITIONS ARE CHECKED.
                        THE REMAINING UNDERFLOW  COND.   AND  THE  REMAINING
                        OVERFLOW  COND.  WILL BE CHECKED LATER USING THE XXX
                        INSTRUCTION.  HERE EACH CONDITION TESTED IS  CHECKED
                        BOTH  WITH  TRAPS  ENABLED (FIU=1 OR FIV=1) AND ALSO
                        WITH TRAPS DISABLED (FIU=0 OR FIV=0).

TEST 3          LDCFD AND LDCDF TEST
------          --------------------

                        THIS IS A TEST OF LDCFD AND LDCDF.

TEST 4          CMPD TEST
------          ---------

                        THIS IS A TEST OF THE CMPD INSTRUCTION.  NOTE THAT A
                        SUBROUTINE  IS  USED TO SET UP OPERANDS, EXECUTE THE
                        INSTRUCTION AND CHECK THE RESULTS

TEST 5          DIVD WITH (FSRC=0) AND (BUT FD) TEST
------          ------------------------------------

                        THIS IS A TEST OF THE DIVD INSTRUCTION WITH  A  ZERO
                        DIVISOR.   THE  CONDITION  IS CHECKED WITH BOTH TRAP
                        ENABLED AND TRAPS DISABLED.

TEST 6          DIVF TEST
------          ---------

                        THIS IS A TEST OF THE DIVF INSTRUCTION.  NOTE THAT A
                        SUBROUTINE  IS  USED TO SET UP THE OPERANDS, EXECUTE
                        THE INSTRUCTION AND CHECK THE RESULTS.

TEST 7          DIVD TEST
------          ---------

                        THIS IS A TEST OF THE DIVD INSTRUCTION.  NOTE THAT A
                        SUBROUTINE  IS  USED TO SET UP THE OPERANDS, EXECUTE
                        THE INSTRUCTION AND CHECK THE RESULTS.

TEST 10         MULF TEST
------          ---------
                        THIS IS A TEST OF THE MULF  INSTRUCTION.   IT  MAKES
                        USE  OF A SUBROUTINE TO SET UP THE OPERANDS, EXECUTE
                        THE MULF INSTRUCTION AND CHECK THE RESULTS.
```

```
                        TEST 11         MULD TEST
                        -------         ---------

                            THIS IS A TEST OF THE MULD INSTRUCTION.  NOTE THAT A
                            SUBROUTINE  IS  USED TO SET UP THE OPERANDS, EXECUTE
                            THE MULD INSTRUCTION AND CHECK THE RESULTS.

                        TEST 12         UNDER\OVER FLOW, USING MULF WITH TRAPS DISABLED, TEST
                        -------         ----------------------------------------------------

                            THIS  IS  A  TEST  OF  THE  OVERFLOW  AND  UNDERFLOW
                            CONDITIONS  USING  THE  MULF  INSTRUCTION WITH TRAPS
                            DISABLED.  NOTE THAT A SUBROUTINE IS USED TO SET  UP
                            THE OPERANDS, EXECUTE THE MULF INSTRUCTION AND CHECK
                            THE RESULTS.

                        TEST 13         UNDER\OVER FLOW, USING MULD WITH TRAPS DISABLED, TEST
                        -------         ----------------------------------------------------

                            THIS  IS  A  TEST  OF  THE  OVERFLOW  AND  UNDERFLOW
                            CONDITIONS   THAT   CAN   ARRISE   USING   THE  MULD
                            INSTRUCTION WITH TRAPS DISABLED.   A  SUBROUTINE  IS
                            USED  TO  SET  UP  THE  OPERANDS,  EXECUTE  THE MULD
                            INSTRUCTION AND CHECK THE RESULTS.

                        TEST 14         UNDER\OVER FLOW, USING MULF WITH TRAPS ENABLED, TEST
                        -------         ----------------------------------------------------

                            THIS  IS  A  TEST  OF  THE  UNDERFLOW  AND  OVERFLOW
                            CONDITIONS   THAT   CAN   OCCUR   USING   THE   MULF
                            INSTRUCTION.  A SUBROUTINE IS CALLED TO SET  UP  THE
                            OPERANDS, EXECUTE THE MULF INSTRUCTION AND CHECK THE
                            RESULTS.   HERE  THE  PARTICULAR  INTERRUPT,  EITHER
                            OVERFLOW  OR  UNDERFLOW, IS ENABLED SO A TRAP SHOULD
                            OCCUR.

                        TEST 15         UNDER\OVER FLOW, USING MULD WITH TRAPS ENABLED, TEST
                        -------         ----------------------------------------------------

                            THIS IS A TEST OF  THE  OVER  FLOW  AND  UNDER  FLOW
                            CONDITIONS  USING  THE  MULD  INSTRUCTION WITH TRAPS
                            ENABLED.   A  SUBROUTINE  IS  USED  TO  SET  UP  THE
                            OPERANDS, EXECUTE THE MULD INSTRUCTION AND CHECK THE
                            RESULTS.

                        TEST 16         MODF TEST
                        -------         ---------

                            THIS IS A TEST OF THE MODF INSTRUCTION, WHICH  MAKES
                            USE  OF A SUBROUTINE TO SET UP THE OPERANDS, EXECUTE
                            THE MODF INSTRUCTION AND CHECK THE RESULTS.

                        TEST 17         MODD TEST
                        -------         ---------

                            THIS IS A TEST OF THE MODD  INSTRUCTION.   IT  MAKES
                            USE OF A SUBROUTINE TO SET UP THE ARGUMENTS, EXECUTE
```

```
                        THE INSTRUCTION AND CHECK THE RESULTS.

TEST 20         UNDER\OVER FLOW, USING MODF WITH TRAPS DISABLED, TEST
-------         ----------------------------------------------------

                THIS IS A TEST OF THE MODF  OVERFLOW  AND  UNDERFLOW
                CONDITIONS.   IT  MAKES USE OF A SUBROUTINE TO SETUP
                THE OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK
                THE RESULTS.  TRAPS ARE DISABLED DURING THIS TEST.

TEST 21         UNDER\OVER FLOW, USING MODD WITH TRAPS DISABLED, TEST
-------         ----------------------------------------------------

                THIS IS A TEST OF THE MODD INSTRUCTION'S  OVER  FLOW
                AND  UNDER FLOW CONDITIONS.  A SUBROUTINE IS USED TO
                SET UP THE OPERANDS, EXECUTE THE MODD INSTUCTION AND
                CHECK THE RESULTS.

TEST 22         INTERRUPT CORRECT FLOWS TEST
-------         ----------------------------

                THIS IS A TEST OF THE 'CORRECT' FLOWS.  THIS PART OF
                THE  MICRO  CODE  HAS  AS  ITS PURPOSE INSURING THAT
                INTERRUPT REQUESTS MADE DURING CERTAIN  LENGTHY  FPP
                INSTRUCTIONS  GET  HONORED.   THIS  IS DONE IN A WAY
                SUCH THAT IF AN INTERRUPT REQUEST OCCURS DURING  ONE
                OF    THESE   INSTRUCTIONS   THE   STATE   OF   THAT
                INSTRUCTION'S EXECUTION WILL BE THE SAME AS IF  THAT
                INSTRUCTION HAD NEVER BEEN FETCHED AND ITS EXECUTION
                NEVER STARTED.  THUS THE MICRO CODE WILL RESTORE ALL
                REGISTERS,  BACK UP THE PC AND LEAVE THE FPS AND AC0
                THROUGH AC5 UNMODIFIED.
                THE INSTRUCTIONS FOR WHICH THIS IS NECESSARY ARE:

                        ADD (OR SUB)
                        DIV
                        MUL
                        MCD
                (BOTH DOUBLE AND FLOATING)

                ALL ADDRESSING MODES WILL BE  TRIED  WITH  THE  ADDD
                INSTRUCTION.   THEN  EACH  OF THE OTHER INSTRUCTIONS
                WILL BE TRIED USING MODE 1.   NOTE  THAT  THIS  TEST
                NEEDS   A   SPECIAL  INTERRUPT  MODULE,  WHICH  WILL
                PROBABLY ONLY  BE  PRESENT  IN  DEC'S  MANUFACTURING
                ENVIRONMENT,  TO  RUN.   THIS  SPECIAL  EQUIPMENT IS
                DESIGNED  TO  RAISE  AN  INTERRUPT  REQUEST  IN  THE
                PROCESSOR IF A BIT IS SET IN ITS STATUS REGISTER AND
                ONLY  WHEN  AN  FPP  INSTRUCTION   IS   ENCOUNTERED.
                THEREFORE   THIS   TEST   WILL   BE   RUN  CONDITIONALLY
                (DEPENDENT UPON WHETHER OR NOT THE  STATUS  REGISTER
                OF  THE  TEST  EQUIPMENT TIMES OUT WHEN REFERENCED).
                THIS TEST CAN ALSO BE SELECTED  BY  TURNING SWITCH 7
                OF  THE  SWITCH  REGISTER  (PHYSICAL OR VIRTUAL) ON.
                THE TEST ASSUMES THAT THE  TEST  EQUIPMENT'S  STATUS
                REGISTER  IS  AT  LOCATION  777774  (NOTE  THAT  ALL
                REFERENCES  TO  THIS  LOCATION  ARE  MADE   INDIRECT
```

```
                        THROUGH  THIS  PROGRAMS  LOCATION CORINT, SO THAT IF
                        THE USER HAS MODIFIED THE  TEST  EQUIPMENT'S  STATUS
                        REGISTER  TO RESPOND TO A DIFFERENT ADDRESS LOCATION
                        CORINT  MUST  BE  MADE  TO  CONTAIN  THAT  STATUS
                        REGISTER'S  NEW ADDRESS).  THIS PROGRAM ASSUMES THAT
                        THE TRAP VECTOR  FOR  THE  TEST  EQUIPMENT  IS  110.
                        AGAIN  NOTE  THAT ALL REFERENCES TO THIS TRAP VECTOR
                        ARE INDIRECT, THROUGH THIS PROGRAM'S LOCATION CORTRP
                        (IF  THE  TEST  EQUIPMENT  IS  MADE  TO  TRAP  TO  A
                        DIFFERENT VECTOR LOCATION CORTRP  MUST  CONTAIN  THE
                        ADDRESS OF THIS VECTOR).
```

10. LISTING

```
                                &

000266                          MNUMBER=266
000002                          PROGNUM=2


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

```
856 000000                              .ENABL  ABS
863                                     .MCALL  .HEADER,.SWRHI,.EQUAT,.SETUP,.$CATCH,.$ACT11,.$CMTAG
864                                     .MCALL  .$EOP,.$SCOPE,.$ERROR,.$SAVE,.$TYPE,.$TYPOCT
865                                     .MCALL  .$TYPDEC,.$TRAP,.$POWER,.$APTHDR,.$APTBLS
866                                     .MCALL  .$APTYPE,.$READ
867                                     .MCALL  .EQUIV                  ;REMOVE FOR PDP-10 ASSEMBLY
868                             .TITLE  CKFPBA0 FP11F FLTG PNT PRT B
                                ;*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
869
870
871     000244                  FPVECT=244
872     177400                  $SWR=177400
873     000200                  $SWRMSK=200
874     000011                  TAB=11
875     000015                  CRLF=15
876
877                             .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
    878                                         .SBTTL  FPP REGISTER DEFINITIONS
    879         000000                  AC0     =%0
    880         000001                  AC1     =%1
    881         000002                  AC2     =%2
    882         000003                  AC3     =%3
    883         000004                  AC4     =%4
    884         000005                  AC5     =%5
    885         000006                  AC6     =%6
    886         000007                  AC7     =%7
    888
    889                                 .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  004336              JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
```

```
                                .SBTTL  COMMON TAGS
                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.
                                        .=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:
894             000266                          .REPT   MNUMBER
896                                     ;ITEM 1
    001442  037724  066262  070604              .WORD   EM1,DH1,DT1,DF1
    001450  070056
                                        ;ITEM 2
    001452  037756  066352  070642              .WORD   EM2,DH2,DT2,DF2
    001460  070074
                                        ;ITEM 3
    001462  040012  066262  070604              .WORD   EM3,DH3,DT3,DF3
    001470  070056
                                        ;ITEM 4
    001472  040117  066262  070604              .WORD   EM4,DH4,DT4,DF4
    001500  070056
                                        ;ITEM 5
    001502  040224  066262  070604              .WORD   EM5,DH5,DT5,DF5
    001510  070056
                                        ;ITEM 6
    001512  040331  066262  070604              .WORD   EM6,DH6,DT6,DF6
    001520  070056
                                        ;ITEM 7
    001522  040436  066262  070604              .WORD   EM7,DH7,DT7,DF7
    001530  070056
                                        ;ITEM 10
    001532  040543  066262  070604              .WORD   EM10,DH10,DT10,DF10
    001540  070056
                                        ;ITEM 11
    001542  040650  066262  070604              .WORD   EM11,DH11,DT11,DF11
    001550  070056
                                        ;ITEM 12
    001552  040757  066262  070604              .WORD   EM12,DH12,DT12,DF12
    001560  070056
                                        ;ITEM 13
    001562  041066  066262  070604              .WORD   EM13,DH13,DT13,DF13
    001570  070056
                                        ;ITEM 14
    001572  041202  066262  070604              .WORD   EM14,DH14,DT14,DF14
    001600  070056
                                        ;ITEM 15
    001602  041314  066262  070604              .WORD   EM15,DH15,DT15,DF15
    001610  070056
                                        ;ITEM 16
    001612  041426  066262  070604              .WORD   EM16,DH16,DT16,DF16
    001620  070056
                                        ;ITEM 17
    001622  041523  066417  070670              .WORD   EM17,DH17,DT17,DF17
    001630  070106
```

```
                                        ;ITEM 20
001632  041600  066262  070710                .WORD   EM20,DH20,DT20,DF20
001640  070115
                                        ;ITEM 21
001642  041632  066466  070732                .WORD   EM21,DH21,DT21,DF21
001650  070115
                                        ;ITEM 22
001652  041664  066555  070754                .WORD   EM22,DH22,DT22,DF22
001660  070115
                                        ;ITEM 23
001662  041745  066262  070772                .WORD   EM23,DH23,DT23,DF23
001670  070125
                                        ;ITEM 24
001672  042022  066262  070772                .WORD   EM24,DH24,DT24,DF24
001700  070125
                                        ;ITEM 25
001702  042120  066262  070772                .WORD   EM25,DH25,DT25,DF25
001710  070125
                                        ;ITEM 26
001712  042205  066262  070772                .WORD   EM26,DH26,DT26,DF26
001720  070125
                                        ;ITEM 27
001722  042120  066262  070772                .WORD   EM27,DH27,DT27,DF27
001730  070125
                                        ;ITEM 30
001732  042303  066262  070772                .WORD   EM30,DH30,DT30,DF30
001740  070125
                                        ;ITEM 31
001742  042355  066262  070772                .WORD   EM31,DH31,DT31,DF31
001750  070125
                                        ;ITEM 32
001752  041770  066262  070772                .WORD   EM32,DH32,DT32,DF32
001760  070125
                                        ;ITEM 33
001762  042422  066262  070772                .WORD   EM33,DH33,DT33,DF33
001770  070151
                                        ;ITEM 34
001772  042445  066262  070772                .WORD   EM34,DH34,DT34,DF34
002000  070151
                                        ;ITEM 35
002002  042477  066262  070772                .WORD   EM35,DH35,DT35,DF35
002010  070151
                                        ;ITEM 36
002012  042552  066262  070772                .WORD   EM36,DH36,DT36,DF36
002020  070151
                                        ;ITEM 37
002022  042620  066262  070772                .WORD   EM37,DH37,DT37,DF37
002030  070125
                                        ;ITEM 40
002032  042643  066262  070772                .WORD   EM40,DH40,DT40,DF40
002040  070125
                                        ;ITEM 41
002042  042675  066262  070772                .WORD   EM41,DH41,DT41,DF41
002050  070125
                                        ;ITEM 42
002052  042750  066262  070772                .WORD   EM42,DH42,DT42,DF42
002060  070125
```

```
                                        ;ITEM 43
002062  043101  066262  070772                .WORD   EM43,DH43,DT43,DF43
002070  070125
                                        ;ITEM 44
002072  043232  066262  070772                .WORD   EM44,DH44,DT44,DF44
002100  070125
                                        ;ITEM 45
002102  043300  066262  070772                .WORD   EM45,DH45,DT45,DF45
002110  070125
                                        ;ITEM 46
002112  043353  066262  070772                .WORD   EM46,DH46,DT46,DF46
002120  070151
                                        ;ITEM 47
002122  043430  066262  070772                .WORD   EM47,DH47,DT47,DF47
002130  070151
                                        ;ITEM 50
002132  043564  066262  070772                .WORD   EM50,DH50,DT50,DF50
002140  070151
                                        ;ITEM 51
002142  043637  066262  070772                .WORD   EM51,DH51,DT51,DF51
002150  070151
                                        ;ITEM 52
002152  043705  066262  070772                .WORD   EM52,DH52,DT52,DF52
002160  070151
                                        ;ITEM 53
002162  051535  066262  071120                .WORD   EM53,DH53,DT53,DF53
002170  070245
                                        ;ITEM 54
002172  051566  066262  071120                .WORD   EM54,DH54,DT54,DF54
002200  070245
                                        ;ITEM 55
002202  051503  066262  071120                .WORD   EM55,DH55,DT55,DF55
002210  070245
                                        ;ITEM 56
002212  051616  066262  071120                .WORD   EM56,DH56,DT56,DF56
002220  070245
                                        ;ITEM 57
002222  051707  066262  071120                .WORD   EM57,DH57,DT57,DF57
002230  070245
                                        ;ITEM 60
002232  051777  066262  071120                .WORD   EM60,DH60,DT60,DF60
002240  070245
                                        ;ITEM 61
002242  052101  066262  071120                .WORD   EM61,DH61,DT61,DF61
002250  070245
                                        ;ITEM 62
002252  052275  066262  071120                .WORD   EM62,DH62,DT62,DF62
002260  070245
                                        ;ITEM 63
002262  052471  066262  071120                .WORD   EM63,DH63,DT63,DF63
002270  070245
                                        ;ITEM 64
002272  052602  066262  071120                .WORD   EM64,DH64,DT64,DF64
002300  070245
                                        ;ITEM 65
002302  052721  066262  071120                .WORD   EM65,DH65,DT65,DF65
002310  070245
```

```
                                        ;ITEM 66
002312  053034  066262  071120                .WORD   EM66,DH66,DT66,DF66
002320  070245
                                        ;ITEM 67
002322  053103  066262  071120                .WORD   EM67,DH67,DT67,DF67
002330  070245
                                        ;ITEM 70
002332  053166  066262  071120                .WORD   EM70,DH70,DT70,DF70
002340  070277
                                        ;ITEM 71
002342  053217  066262  071120                .WORD   EM71,DH71,DT71,DF71
002350  070277
                                        ;ITEM 72
002352  053247  066262  071120                .WORD   EM72,DH72,DT72,DF72
002360  070277
                                        ;ITEM 73
002362  053247  066262  071120                .WORD   EM73,DH73,DT73,DF73
002370  070277
                                        ;ITEM 74
002372  053301  066262  071120                .WORD   EM74,DH74,DT74,DF74
002400  070277
                                        ;ITEM 75
002402  053410  066262  071120                .WORD   EM75,DH75,DT75,DF75
002410  070277
                                        ;ITEM 76
002412  053501  066262  071120                .WORD   EM76,DH76,DT76,DF76
002420  070277
                                        ;ITEM 77
002422  053571  066262  071120                .WORD   EM77,DH77,DT77,DF77
002430  070277
                                        ;ITEM 100
002432  053701  066262  071120                .WORD   EM100,DH100,DT100,DF100
002440  070277
                                        ;ITEM 101
002442  053764  066262  071120                .WORD   EM101,DH101,DT101,DF101
002450  070277
                                        ;ITEM 102
002452  054160  066262  071120                .WORD   EM102,DH102,DT102,DF102
002460  070277
                                        ;ITEM 103
002462  054354  066262  071120                .WORD   EM103,DH103,DT103,DF103
002470  070277
                                        ;ITEM 104
002472  054466  066262  071120                .WORD   EM104,DH104,DT104,DF104
002500  070277
                                        ;ITEM 105
002502  054633  066262  071120                .WORD   EM105,DH105,DT105,DF105
002510  070277
                                        ;ITEM 106
002512  054742  066262  071120                .WORD   EM106,DH106,DT106,DF106
002520  070277
                                        ;ITEM 107
002522  055051  066262  071120                .WORD   EM107,DH107,DT107,DF107
002530  070277
                                        ;ITEM 110
002532  055160  066262  071120                .WORD   EM110,DH110,DT110,DF110
002540  070277
```

```
                                        ;ITEM 111
002542  043775  066262  070772                .WORD   EM111,DH111,DT111,DF111
002550  070125
                                        ;ITEM 112
002552  044052  066262  070772                .WORD   EM112,DH112,DT112,DF112
002560  070125
                                        ;ITEM 113
002562  044130  066262  070772                .WORD   EM113,DH113,DT113,DF113
002570  070125
                                        ;ITEM 114
002572  044206  066262  070772                .WORD   EM114,DH114,DT114,DF114
002600  070125
                                        ;ITEM 115
002602  044265  066623  071044                .WORD   EM115,DH115,DT115,DF115
002610  070175
                                        ;ITEM 116
002612  044343  066623  071044                .WORD   EM116,DH116,DT116,DF116
002620  070175
                                        ;ITEM 117
002622  044422  066623  071044                .WORD   EM117,DH117,DT117,DF117
002630  070175
                                        ;ITEM 120
002632  044560  066623  071044                .WORD   EM120,DH120,DT120,DF120
002640  070175
                                        ;ITEM 121
002642  044716  066623  071044                .WORD   EM121,DH121,DT121,DF121
002650  070175
                                        ;ITEM 122
002652  045053  066623  071044                .WORD   EM122,DH122,DT122,DF122
002660  070175
                                        ;ITEM 123
002662  045210  066262  070772                .WORD   EM123,DH123,DT123,DF123
002670  070151
                                        ;ITEM 124
002672  045266  066262  070772                .WORD   EM124,DH124,DT124,DF124
002700  070151
                                        ;ITEM 125
002702  045345  066262  070772                .WORD   EM125,DH125,DT125,DF125
002710  070151
                                        ;ITEM 126
002712  045423  066262  070772                .WORD   EM126,DH126,DT126,DF126
002720  070151
                                        ;ITEM 127
002722  045502  066623  071044                .WORD   EM127,DH127,DT127,DF127
002730  070221
                                        ;ITEM 130
002732  045560  066623  071044                .WORD   EM130,DH130,DT130,DF130
002740  070221
                                        ;ITEM 131
002742  045637  066623  071044                .WORD   EM131,DH131,DT131,DF131
002750  070221
                                        ;ITEM 132
002752  045775  066262  070772                .WORD   EM132,DH132,DT132,DF132
002760  070151
                                        ;ITEM 133
002762  046133  066623  071044                .WORD   EM133,DH133,DT133,DF133
002770  070221
```

```
                                        ;ITEM 134
002772  046271  066623  071044                .WORD   EM134,DH134,DT134,DF134
003000  070221
                                        ;ITEM 135
003002  046426  066262  070772                .WORD   EM135,DH135,DT135,DF135
003010  070151
                                        ;ITEM 136
003012  046563  066623  071044                .WORD   EM136,DH136,DT136,DF136
003020  070221
                                        ;ITEM 137
003022  046720  066623  071044                .WORD   EM137,DH137,DT137,DF137
003030  070175
                                        ;ITEM 140
003032  047006  066623  071044                .WORD   EM140,DH140,DT140,DF140
003040  070175
                                        ;ITEM 141
003042  043775  066623  071044                .WORD   EM141,DH141,DT141,DF141
003050  070175
                                        ;ITEM 142
003052  044052  066623  071044                .WORD   EM142,DH142,DT142,DF142
003060  070175
                                        ;ITEM 143
003062  047075  066623  071044                .WORD   EM143,DH143,DT143,DF143
003070  070175
                                        ;ITEM 144
003072  047153  066623  071044                .WORD   EM144,DH144,DT144,DF144
003100  070175
                                        ;ITEM 145
003102  047232  066262  070772                .WORD   EM145,DH145,DT145,DF145
003110  070125
                                        ;ITEM 146
003112  047377  066262  070772                .WORD   EM146,DH146,DT146,DF146
003120  070125
                                        ;ITEM 147
003122  047544  066262  070772                .WORD   EM147,DH147,DT147,DF147
003130  070125
                                        ;ITEM 150
003132  047710  066262  070772                .WORD   EM150,DH150,DT150,DF150
003140  070125
                                        ;ITEM 151
003142  050054  066623  071044                .WORD   EM151,DH151,DT151,DF151
003150  070221
                                        ;ITEM 152
003152  050142  066623  071044                .WORD   EM152,DH152,DT152,DF152
003160  070221
                                        ;ITEM 153
003162  045210  066623  071044                .WORD   EM153,DH153,DT153,DF153
003170  070221
                                        ;ITEM 154
003172  045266  066623  071044                .WORD   EM154,DH154,DT154,DF154
003200  070221
                                        ;ITEM 155
003202  050231  066623  071044                .WORD   EM155,DH155,DT155,DF155
003210  070221
                                        ;ITEM 156
003212  050307  066623  071044                .WORD   EM156,DH156,DT156,DF156
003220  070221
```

```
                                        ;ITEM 157
003222  050366  066262  070772                .WORD   EM157,DH157,DT157,DF157
003230  070151
                                        ;ITEM 160
003232  050533  066623  071044                .WORD   EM160,DH160,DT160,DF160
003240  070221
                                        ;ITEM 161
003242  050671  066262  070772                .WORD   EM161,DH161,DT161,DF161
003250  070151
                                        ;ITEM 162
003252  051036  066262  070772                .WORD   EM162,DH162,DT162,DF162
003260  070151
                                        ;ITEM 163
003262  051202  066623  071044                .WORD   EM163,DH163,DT163,DF163
003270  070221
                                        ;ITEM 164
003272  051337  066262  070772                .WORD   EM164,DH164,DT164,DF164
003300  070151
                                        ;ITEM 165
003302  055270  066623  071206                .WORD   EM165,DH165,DT165,DF165
003310  070331
                                        ;ITEM 166
003312  055354  066623  071206                .WORD   EM166,DH166,DT166,DF166
003320  070331
                                        ;ITEM 167
003322  055437  066623  071206                .WORD   EM167,DH167,DT167,DF167
003330  070331
                                        ;ITEM 170
003332  055471  066623  071206                .WORD   EM170,DH170,DT170,DF170
003340  070331
                                        ;ITEM 171
003342  055557  066623  071206                .WORD   EM171,DH171,DT171,DF171
003350  070331
                                        ;ITEM 172
003352  055702  066623  071206                .WORD   EM172,DH172,DT172,DF172
003360  070331
                                        ;ITEM 173
003362  055767  066623  071206                .WORD   EM173,DH173,DT173,DF173
003370  070331
                                        ;ITEM 174
003372  056135  066623  071206                .WORD   EM174,DH174,DT174,DF174
003400  070331
                                        ;ITEM 175
003402  056303  066623  071206                .WORD   EM175,DH175,DT175,DF175
003410  070331
                                        ;ITEM 176
003412  056415  066623  071206                .WORD   EM176,DH176,DT176,DF176
003420  070363
                                        ;ITEM 177
003422  056501  066623  071206                .WORD   EM177,DH177,DT177,DF177
003430  070363
                                        ;ITEM 200
003432  056564  066623  071206                .WORD   EM200,DH200,DT200,DF200
003440  070363
                                        ;ITEM 201
003442  056616  066623  071206                .WORD   EM201,DH201,DT201,DF201
003450  070363
```

```
                                        ;ITEM 202
003452  056705  066623  071206                .WORD   EM202,DH202,DT202,DF202
003460  070363
                                        ;ITEM 203
003462  057047  066623  071206                .WORD   EM203,DH203,DT203,DF203
003470  070363
                                        ;ITEM 204
003472  057211  066623  071206                .WORD   EM204,DH204,DT204,DF204
003500  070363
                                        ;ITEM 205
003502  057277  066623  071206                .WORD   EM205,DH205,DT205,DF205
003510  070363
                                        ;ITEM 206
003512  057445  066623  071206                .WORD   EM206,DH206,DT206,DF206
003520  070363
                                        ;ITEM 207
003522  057613  066762  071316                .WORD   EM207,DH207,DT207,DF207
003530  070425
                                        ;ITEM 210
003532  057655  067052  071274                .WORD   EM210,DH210,DT210,DF210
003540  070415
                                        ;ITEM 211
003542  057717  067052  071316                .WORD   EM211,DH211,DT211,DF211
003550  070425
                                        ;ITEM 212
003552  060063  066762  071370                .WORD   EM212,DH212,DT212,DF212
003560  070451
                                        ;ITEM 213
003562  060247  066762  071370                .WORD   EM213,DH213,DT213,DF213
003570  070451
                                        ;ITEM 214
003572  060433  066762  071370                .WORD   EM214,DH214,DT214,DF214
003600  070451
                                        ;ITEM 215
003602  060617  066762  071370                .WORD   EM215,DH215,DT215,DF215
003610  070451
                                        ;ITEM 216
003612  061003  067052  071316                .WORD   EM216,DH216,DT216,DF216
003620  070425
                                        ;ITEM 217
003622  061165  067113  071452                .WORD   EM217,DH217,DT217,DF217
003630  070501
                                        ;ITEM 220
003632  061227  066722  071316                .WORD   EM220,DH220,DT220,DF220
003640  070425
                                        ;ITEM 221
003642  061457  067052  071316                .WORD   EM221,DH221,DT221,DF221
003650  070425
                                        ;ITEM 222
003652  061723  066722  071316                .WORD   EM222,DH222,DT222,DF222
003660  070425
                                        ;ITEM 223
003662  062154  067052  071316                .WORD   EM223,DH223,DT223,DF223
003670  070425
                                        ;ITEM 224
003672  062421  066722  071316                .WORD   EM224,DH224,DT224,DF224
003700  070425
```

ERROR POINTER TABLE

```
                                        ;ITEM 225
003702  062652  067052  071316                .WORD   EM225,DH225,DT225,DF225
003710  070425
                                        ;ITEM 226
003712  063117  067166  071370                .WORD   EM226,DH226,DT226,DF226
003720  070451
                                        ;ITEM 227
003722  063252  067255  071370                .WORD   EM227,DH227,DT227,DF227
003730  070451
                                        ;ITEM 230
003732  063405  067166  071370                .WORD   EM230,DH230,DT230,DF230
003740  070451
                                        ;ITEM 231
003742  063541  067255  071274                .WORD   EM231,DH231,DT231,DF231
003750  070451
                                        ;ITEM 232
003752  057655  067255  071316                .WORD   EM232,DH232,DT232,DF232
003760  070451
                                        ;ITEM 233
003762  063675  067344  071472                .WORD   EM233,DH233,DT233,DF233
003770  070510
                                        ;ITEM 234
003772  063734  067405  071526                .WORD   EM234,DH234,DT234,DF234
004000  070525
                                        ;ITEM 235
004002  064020  067473  071546                .WORD   EM235,DH235,DT235,DF235
004010  070534
                                        ;ITEM 236
004012  064066  067533  071526                .WORD   EM236,DH236,DT236,DF236
004020  070525
                                        ;ITEM 237
004022  064121  067405  071526                .WORD   EM237,DH237,DT237,DF237
004030  070525
                                        ;ITEM 240
004032  064205  067621  071526                .WORD   EM240,DH240,DT240,DF240
004040  070525
                                        ;ITEM 241
004042  064241  067344  071472                .WORD   EM241,DH241,DT241,DF241
004050  070510
                                        ;ITEM 242
004052  064344  067621  071526                .WORD   EM242,DH242,DT242,DF242
004060  070525
                                        ;ITEM 243
004062  064400  067344  071472                .WORD   EM243,DH243,DT243,DF243
004070  070510
                                        ;ITEM 244
004072  064503  067344  071472                .WORD   EM244,DH244,DT244,DF244
004100  070510
                                        ;ITEM 245
004102  064542  067344  071472                .WORD   EM245,DH245,DT245,DF245
004110  070510
                                        ;ITEM 246
004112  043405  066262  070772                .WORD   EM246,DH246,DT246,DF246
004120  070151
                                        ;ITEM 247
004122  064624  067711  071560                .WORD   EM247,DH247,DT247,DF247
004130  070540
```

```
                                        ;ITEM 250
004132  064660  067756  071576                .WORD   EM250,DH250,DT250,DF250
004140  070540
                                        ;ITEM 251
004142  064712  067756  071576                .WORD   EM251,DH251,DT251,DF251
004150  070540
                                        ;ITEM 252
004152  064745  066352  071610                .WORD   EM252,DH252,DT252,DF252
004160  070546
                                        ;ITEM 253
004162  065036  066262  071622                .WORD   EM253,DH253,DT253,DF253
004170  070552
                                        ;ITEM 254
004172  065122  066262  071622                .WORD   EM254,DH254,DT254,DF254
004200  070552
                                        ;ITEM 255
004202  065207  066262  071622                .WORD   EM255,DH255,DT255,DF255
004210  070552
                                        ;ITEM 256
004212  065275  066262  071622                .WORD   EM256,DH256,DT256,DF256
004220  070552
                                        ;ITEM 257
004222  065364  066262  071622                .WORD   EM257,DH257,DT257,DF257
004230  070552
                                        ;ITEM 260
004232  065452  066262  071622                .WORD   EM260,DH260,DT260,DF260
004240  070552
                                        ;ITEM 261
004242  065541  066262  071622                .WORD   EM261,DH261,DT261,DF261
004250  070552
                                        ;ITEM 262
004252  065631  066262  071622                .WORD   EM262,DH262,DT262,DF262
004260  070552
                                        ;ITEM 263
004262  065722  066262  071622                .WORD   EM263,DH263,DT263,DF263
004270  070552
                                        ;ITEM 264
004272  066007  066262  071622                .WORD   EM264,DH264,DT264,DF264
004300  070552
                                        ;ITEM 265
004302  066074  066262  071622                .WORD   EM265,DH265,DT265,DF265
004310  070552
                                        ;ITEM 266
004312  066161  070016  071610                .WORD   EM266,DH266,DT266,DF266
004320  070546
897
898
899                                     .SBTTL  ACT11 HOOKS
                                        ;;******************************************************************
                                        ;HOOKS REQUIRED BY ACT11
                004322                          $SVPC=.                 ;SAVE PC
                000046                          .=46
000046          033356                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052                          .=52
000052          000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
                004322                          .=$SVPC                 ;; RESTORE PC
900                                     .SBTTL  APT PARAMETER BLOCK
```

```
                                ;;**********************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;**********************************************************************
                    004322              .$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  004322              $APTHDR  ;;POINT TO APT HEADER BLOCK
                    004322              .=.$X    ;;RESET LOCATION COUNTER
                                ;;**********************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.
            004322              $APTHD:
            004322  000000      $HIBTS: .WORD    0        ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
            004324  001316      $MBADR: .WORD    $MAIL    ;;ADDRESS OF APT MAILBOX (BITS 0-15)
            004326  000010      $TSTM:  .WORD    10       ;;RUN TIM OF LONGEST TEST
            004330  000040      $PASTM: .WORD    40       ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
            004332  000000      $UNITM: .WORD    0        ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
            004334  000052              .WORD    $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
        901
        902
        903 004336              START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
            004336  012706  001100      MOV      #$CMTAG,R6       ;;FIRST LOCATION TO BE CLEARED
            004342  005026              CLR      (R6)+            ;;CLEAR MEMORY LOCATION
            004344  022706  001140      CMP      #SWR,R6 ;;DONE?
            004350  001374              BNE      .-6              ;;LOOP BACK IF NO
            004352  012706  001100      MOV      #STACK,SP        ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
            004356  012737  033436  000020      MOV      #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
            004364  012737  000340  000022      MOV      #340,@#IOTVEC+2 ;;LEVEL 7
            004372  012737  033716  000030      MOV      #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
            004400  012737  000340  000032      MOV      #340,@#EMTVEC+2 ;;LEVEL 7
            004406  012737  035734  000034      MOV      #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
            004414  012737  000340  000036      MOV      #340,@#TRAPVEC+2;LEVEL 7
            004422  012737  036020  000024      MOV      #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
            004430  012737  000340  000026      MOV      #340,@#PWRVEC+2 ;;LEVEL 7
            004436  016767  026536  026526      MOV      $ENDCT,$EOPCT    ;;SETUP END-OF-PROGRAM COUNTER
            004444  005067  174632              CLR      $TIMES           ;;INITIALIZE NUMBER OF ITERATIONS
            004450  005067  174630              CLR      $ESCAPE          ;;CLEAR THE ESCAPE ON ERROR ADDRESS
            004454  112767  000001  174433      MOVB     #1,$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''.
            004462  012737  033422  000014      MOV      #$RTRN,@#TBITVEC ;;SET ''T'' BIT VECTOR TO $RTRN
            004470  012737  000340  000016      MOV      #340,@#TBITVEC+2 ;;LEVEL 7
            004476  012767  000002  026716      MOV      #RTI,$RTRN       ;;SET $RTRN TO A RTI
            004504  012737  004532  000010      MOV      #65$,@#RESVEC    ;;TRY TO DO A RTT
            004512  005046              CLR      -(SP)            ;;DUMMY PS
            004514  012746  004522              MOV      #64$,-(SP)       ;;AND PC
            004520  000006              RTT                       ;;TRY THE RTT
            004522  012767  000006  026672  64$:  MOV      #RTT,$RTRN       ;;RTT IS LEGAL--SET $RTRN TO A RTT
            004530  000402              BR       66$
            004532  062706  000010      65$:  ADD      #10,SP           ;;RTT ILLEGAL--CLEAN OFF THE STACK
            004536  012737  000012  000010  66$:  MOV      #RESVEC+2,@#RESVEC ;;RESTORE TRAP CATCHER
            004544  005067  026660              CLR      $TBIT            ;;CLEAR ''T'' BIT SWITCH
            004550  012767  004550  174330      MOV      #.,$LPADR        ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
```

```
004556  012767  004556  174324          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
004564  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
004570  012737  004624  000004  MOV     #67$,@#ERRVEC   ;;SET UP ERROR VECTOR
004576  012767  177570  174334  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
004604  012767  177570  174330  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
004612  022777  177777  174320  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
004620  001012                  BNE     69$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
004622  000403                  BR      68$             ;;BRANCH IF NO TIMEOUT
004624  012716  004632  67$:    MOV     #68$,(SP)       ;;SET UP FOR TRAP RETURN
004630  000002                  RTI
004632  012767  000176  174300 68$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
004640  012767  000174  174274          MOV     #DISPREG,DISPLAY
004646  012637  000004  69$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
004652  005067  174446          CLR     $PASS           ;;CLEAR PASS COUNT
004656  132767  000200  174453  BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
004664  001403                  BEQ     70$             ;;YES,USE NON-APT SWITCH
004666  012767  001340  174244  MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
004674                  70$:
904                             .SBTTL  TYPE PROGRAM NAME
                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
004674  005227  177777          INC     #-1             ;;FIRST TIME?
004700  001047                  BNE     71$             ;;BRANCH IF NO
004702  022737  033356  000042  CMP     #$ENDAD,@#42    ;;ACT-11?
004710  001443                  BEQ     71$             ;;BRANCH IF YES
004712  104401  004760          TYPE    ,72$            ;;TYPE ASCIZ STRING
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
004716  005737  000042          TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
004722  001012                  BNE     73$             ;;BRANCH IF YES
004724  126727  174406  000001  CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
004732  001406                  BEQ     73$             ;;BRANCH IF YES
004734  026727  174200  000176  CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
004742  001005                  BNE     74$             ;;BRANCH IF NO
004744  104405                  GTSWR                   ;;GET SOFT-SWR SETTINGS
004746  000403                  BR      74$
004750  112767  000001  174156 73$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
004756                  74$:
004756  000420                  BR      71$             ;;GET OVER THE ASCIZ
                                ;;72$:  .ASCIZ  <CRLF>*CKFPBA0 FP11F FLTG PNT PRT B*<CRLF>
005020                  71$:
905
906 005020              LOOP:
907
908
919                             ;;******************************************************************
                                ;*TEST 1         ROUND\TRUNK TEST
                                ;*
                                ;* THIS IS A TEST OF THE ROUND\TRUNK
                                ;* FLOWS.  IN PARTICULAR TWO THINGS ARE TESTED:
                                ;* FIRST A CONDITION IN WHICH ROUNDING
                                ;* RESULTS IN THE NEED FOR RENORMALIZATION, AND
                                ;* SECOND  THE PSW CONDITION CODES N AND
                                ;* Z BIT COMMBINATIONS
                                ;*
                                ;;******************************************************************
```

```
     005020  000004                    TST1:   SCOPE
  920
  921                                        ;ROUND AND NORMALIZE TEST
  922
  923 005022                            HH1:
      005022  104413                           LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
  924 005024  012704  003200                    MOV     #3200,R4                 ;SET FIU, FIV, AND FD
  925 005030  170104                            LDFPS   R4
  926 005032  012737  005052  001236            MOV     #HH2,@#$TMP2
  927 005040  012700  006602                    MOV     #HHP0,R0                 ;SET AC0 OPERAND
  928 005044  172410                            LDD     (R0),AC0
  929 005046  012700  006612                    MOV     #HHP1,R0                 ;FSPC
  930 005052  172010                    HH2:    ADDD    (R0),AC0                 ;TEST INSTRUCTION
  931 005054  170205                            STFPS   R5                       ;GET FPS
  932 005056  012700  006572                    MOV     #HHDAT0,R0               ;GET THE RESULT
  933 005062  174010                            STD     AC0,(R0)
  934 005064  012701  006622                    MOV     #HHP2,R1                 ;IS IT CORRECT
  935 005070  012702  000004                    MOV     #4,R2
  936 005074  022021                    HH3:    CMP     (R0)+,(R1)+
  937 005076  001415                            BEQ     HH6
  938 005100  012700  006572                    MOV     #HHDAT0,R0               ;DID FLOW GO
  939 005104  012701  006632                    MOV     #HHP3,R1                 ;FROM STATE 663
  940 005110  012702  000004                    MOV     #4,R2                    ;TO 313 INSTEAD
  941 005114  022021                    HH4:    CMP     (R0)+,(R1)+              ;OF TO 353
  942 005116  001402                            BEQ     HH5
  943 005120  000137  005612                    JMP     @#HHER0
  944 005124  077205                    HH5:    SOB     R2,HH4
  945 005126  000137  005660                    JMP     @#HHER1
  946 005132  077220                    HH6:    SOB     R2,HH3
  947 005134  020405                            CMP     R4,R5            ;FPS CORRECT?
  948 005136  001402                            BEQ     HH7
  949 005140  000137  005726                    JMP     @#HHER00
  950
  951                                        ;THIS IS A TEST OF THE ABILITY
  952                                        ;OF NORMALIZE TO PRODUCE A ZERO EXP. AND
  953                                        ;OF THE R\T ALGORITHM TO PORPERLY SET THE FPS
  954
  955 005144                            HH7:
      005144  104413                           LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
  956 005146  012704  043200                    MOV     #043200,R4               ;SET FIU,FIV,AND FD
  957                                                                            ;FID
  958 005152  170104                            LDFPS   R4
  959 005154  012737  005202  001236            MOV     #HH8,@#$TMP2             ;IN CASE UNDERFLOW
  960 005162  012737  006522  000244            MOV     #HHTRAP,@#FPVECT         ;TRAP OCCURS
  961 005170  012700  006652                    MOV     #HHP5,R0                 ;SET AC0 OPERAND
  962 005174  172410                            LDD     (R0),AC0
  963 005176  012700  006662                    MOV     #HHP6,R0                 ;FSPC
  964 005202  172010                    HH8:    ADDD    (R0),AC0                 ;TEST INSTRUCTION
  965 005204  170205                            STFPS   R5                       ;GET FPS
  966 005206  012700  006572                    MOV     #HHDAT0,R0               ;GET THE RESULT
  967 005212  174010                            STD     AC0,(R0)
  968 005214  012701  006642                    MOV     #HHP4,R1                 ;IS IT CORRECT
  969 005220  012702  000004                    MOV     #4,R2
  970 005224  022021                    HH9:    CMP     (R0)+,(R1)+
  971 005226  001402                            BEQ     HH10
  972 005230  000137  005774                    JMP     @#HHER2
  973 005234  077205                    HH10:   SOB     R2,HH9
```

```
                                                                G 3
 974 005236  052704  100004                 BIS     #100004,R4              ;FPS CORRECT?
 975 005242  020405                         CMP     R4,R5
 976 005244  001402                         BEQ     HH11
 977 005246  000137  006042                 JMP     @#HHER3
 978                                ;THIS IS A TEST OF THE R\T ALGORITHM'S
 979                                ;ABILITY TO SET BOTH N AND Z ON A - 0 RESULT.
 980 005252                         HH11:
     005252  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 981                                                                 ;SET FIV, FIV, AND FD
 982 005254  012704  043200                 MOV     #043200,R4
 983 005260  170104                         LDFPS   R4
 984 005262  012737  005302  001236         MOV     #HH12,@#$TMP2
 985 005270  012700  006702                 MOV     #HHP8,R0                ;SET AC0 OPERAND
 986 005274  172410                         LDD     (R0),AC0
 987 005276  012700  006712                 MOV     #HHP9,R0                ;FSPC
 988 005302  172010                 HH12:   ADDD    (R0),AC0                ;TEST INSTRUCTION
 989 005304  170205                         STFPS   R5                      ;GET FPS
 990 005306  012700  006572                 MOV     #HHDAT0,R0              ;GET THE RESULT
 991 005312  174010                         STD     AC0,(R0)
 992 005314  012701  006672                 MOV     #HHP7,R1                ;IS IT CORRECT
 993 005320  012702  000004                 MOV     #4,R2
 994 005324  022021                 HH13:   CMP     (R0)+,(R1)+
 995 005326  001415                         BEQ     HH16
 996 005330  012700  006572                 MOV     #HHDAT0,R0
 997 005334  012701  006642                 MOV     #HHP4,R1
 998 005340  012702  000004                 MOV     #4,R2
 999 005344  022021                 HH14:   CMP     (R0)+,(R1)+
1000 005346  001402                         BEQ     HH15
1001 005350  000137  006110                 JMP     @#HHER4
1002 005354  077205                 HH15:   SOB     R2,HH14
1003 005356  000137  006156                 JMP     @#HHER5
1004 005362  077220                 HH16:   SOB     R2,HH13
1005 005364  052704  100014                 BIS     #100014,R4              ;FPS CORRECT?
1006 005370  020405                         CMP     R4,R5
1007 005372  001402                         BEQ     HH17
1008 005374  000137  006224                 JMP     @#HHER6
1009                               ;TEST THAT CC ARE CLEARED BY R\T
1010 005400                         HH17:
     005400  104413                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1011 005402  012704  000200                 MOV     #00200,R4               ;SET FIV, FIV, AND FD
1012 005406  170104                         LDFPS   R4
1013 005410  012737  005436  001236         MOV     #HH18,@#$TMP2
1014 005416  012737  036660  000244         MOV     #FPSPUR,@#FPVECT
1015 005424  012700  006702                 MOV     #HHP8,R0                ;SET AC0 OPERAND
1016 005430  172410                         LDD     (R0),AC0
1017 005432  012700  006702                 MOV     #HHP8,R0                ;FSPC
1018 005436  172010                 HH18:   ADDD    (R0),AC0                ;TEST INSTRUCTION
1019 005440  170205                         STFPS   R5                      ;GET FPS
1020 005442  012700  006572                 MOV     #HHDAT0,R0              ;GET THE RESULT
1021 005446  174010                         STD     AC0,(R0)
1022 005450  012701  006722                 MOV     #HHP10,R1               ;IS IT CORRECT
1023 005454  012702  000004                 MOV     #4,R2
1024 005460  022021                 HH19:   CMP     (R0)+,(R1)+
1025 005462  001402                         BEQ     HH20
1026 005464  000137  006272                 JMP @#HHER7
1027 005470  077205                 HH20:   SOB     R2,HH19
1028 005472  052704  000000                 BIS     #00000,R4               ;FPS CORRECT?
```

```
1029 005476  020405                        CMP     R4,R5
1030 005500  001402                        BEQ     HH21
1031 005502  000137  006340                JMP     @#HHER8
1032                                ;TEST THAT N IS SET BY R\T
1033 005506                         HH21:
     005506  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1034 005510  012704  003200                MOV     #3200,R4                ;SET FIV, FIV, AND FD
1035 005514  170104                        LDFPS   R4
1036 005516  012737  005536  001236        MOV     #HH22,@#$TMP2
1037 005524  012700  006652                MOV     #HHP5,R0                ;SET AC0 OPERAND
1038 005530  172410                        LDD     (R0),AC0
1039 005532  012700  006652                MOV     #HHP5,R0                ;FSPC
1040 005536  172010                 HH22:  ADDD    (R0),AC0                ;TEST INSTRUCTION
1041 005540  170205                        STFPS   R5                      ;GET FPS
1042 005542  012700  006572                MOV     #HHDAT0,R0              ;GET THE RESULT
1043 005546  174010                        STD     AC0,(R0)
1044 005550  012701  006732                MOV     #HHP11,R1               ;IS IT CORRECT
1045 005554  012702  000004                MOV     #4,R2
1046 005560  022021                 HH23:  CMP     (R0)+,(R1)+
1047 005562  001402                        BEQ     HH24
1048 005564  000137  006406                JMP     @#HHER9
1049 005570  077205                 HH24:  SOB     R2,HH23
1050 005572  052704  000010                BIS     #10,R4
1051 005576  020405                        CMP     R4,R5                   ;FPS CORRECT?
1052 005600  001402                        BEQ     HH25
1053 005602  000137  006454                JMP     @#HHER10
1054 005606  000137  006742         HH25:  JMP     @#HHDONE
1055 005612                         HHER0:
     005612  010537  001252                MOV     R5,@#$TMP10
     005616  010437  001254                MOV     R4,@#$TMP11
     005622  012737  006612  001240        MOV     #HHP1,@#$TMP3
     005630  012737  006602  001242        MOV     #HHP0,@#$TMP4
     005636  012737  006572  001244        MOV     #HHDAT0,@#$TMP5
     005644  012737  006622  001246        MOV     #HHP2,@#$TMP6
     005652  104207                 1$:    ERROR   +207
     005654  000137  006742                JMP     @#HHDONE
1056 005660                         HHER1:
     005660  010537  001252                MOV     R5,@#$TMP10
     005664  010437  001254                MOV     R4,@#$TMP11
     005670  012737  006612  001240        MOV     #HHP1,@#$TMP3
     005676  012737  006602  001242        MOV     #HHP0,@#$TMP4
     005704  012737  006572  001244        MOV     #HHDAT0,@#$TMP5
     005712  012737  006622  001246        MOV     #HHP2,@#$TMP6
     005720  104211                 1$:    ERROR   +211
     005722  000137  006742                JMP     @#HHDONE
1057 005726                         HHER00:
     005726  010537  001252                MOV     R5,@#$TMP10
     005732  010437  001254                MOV     R4,@#$TMP11
     005736  012737  006612  001240        MOV     #HHP1,@#$TMP3
     005744  012737  006602  001242        MOV     #HHP0,@#$TMP4
     005752  012737  006572  001244        MOV     #HHDAT0,@#$TMP5
     005760  012737  006622  001246        MOV     #HHP2,@#$TMP6
     005766  104210                 1$:    ERROR   +210
     005770  000137  006742                JMP     @#HHDONE
1058 005774                         HHER2:
     005774  010537  001252                MOV     R5,@#$TMP10
     006000  010437  001254                MOV     R4,@#$TMP11
```

```
                006004  012737  006662  001240          MOV     #HHP6,@#$TMP3
                006012  012737  006652  001242          MOV     #HHP5,@#$TMP4
                006020  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
                006026  012737  006642  001246          MOV     #HHP4,@#$TMP6
                006034  104207                  1$:     ERROR   +207
                006036  000137  006742                  JMP     @#HHDONE
          1059  006042                          HHER3:
                006042  010537  001252                  MOV     R5,@#$TMP10
                006046  010437  001254                  MOV     R4,@#$TMP11
                006052  012737  006662  001240          MOV     #HHP6,@#$TMP3
                006060  012737  006652  001242          MOV     #HHP5,@#$TMP4
                006066  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
                006074  012737  006642  001246          MOV     #HHP4,@#$TMP6
                006102  104214                  1$:     ERROR   +214
                006104  000137  006742                  JMP     @#HHDONE
          1060  006110                          HHER4:
                006110  010537  001252                  MOV     R5,@#$TMP10
                006114  010437  001254                  MOV     R4,@#$TMP11
                006120  012737  006712  001240          MOV     #HHP9,@#$TMP3
                006126  012737  006702  001242          MOV     #HHP8,@#$TMP4
                006134  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
                006142  012737  006672  001246          MOV     #HHP7,@#$TMP6
                006150  104207                  1$:     ERROR   +207
                006152  000137  006742                  JMP     @#HHDONE
          1061  006156                          HHER5:
                006156  010537  001252                  MOV     R5,@#$TMP10
                006162  010437  001254                  MOV     R4,@#$TMP11
                006166  012737  006712  001240          MOV     #HHP9,@#$TMP3
                006174  012737  006702  001242          MOV     #HHP8,@#$TMP4
                006202  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
                006210  012737  006672  001246          MOV     #HHP7,@#$TMP6
                006216  104216                  1$:     ERROR   +216
                006220  000137  006742                  JMP     @#HHDONE
          1062  006224                          HHER6:
                006224  010537  001252                  MOV     R5,@#$TMP10
                006230  010437  001254                  MOV     R4,@#$TMP11
                006234  012737  006712  001240          MOV     #HHP9,@#$TMP3
                006242  012737  006702  001242          MOV     #HHP8,@#$TMP4
                006250  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
                006256  012737  006672  001246          MOV     #HHP7,@#$TMP6
                006264  104215                  1$:     ERROR   +215
                006266  000137  006742                  JMP     @#HHDONE
          1063  006272                          HHER7:
                006272  010537  001252                  MOV     R5,@#$TMP10
                006276  010437  001254                  MOV     R4,@#$TMP11
                006302  012737  006702  001240          MOV     #HHP8,@#$TMP3
                006310  012737  006702  001242          MOV     #HHP8,@#$TMP4
                006316  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
                006324  012737  006722  001246          MOV     #HHP10,@#$TMP6
                006332  104207                  1$:     ERROR   +207
                006334  000137  006742                  JMP     @#HHDONE
          1064  006340                          HHER8:
                006340  010537  001252                  MOV     R5,@#$TMP10
                006344  010437  001254                  MOV     R4,@#$TMP11
                006350  012737  006702  001240          MOV     #HHP8,@#$TMP3
                006356  012737  006702  001242          MOV     #HHP8,@#$TMP4
                006364  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
```

```
                006372  012737  006722  001246          MOV     #HHP10,@#$TMP6
                006400  104212                  1$:     ERROR   +212
                006402  000137  006742                  JMP     @#HHDONE
     1065       006406                          HHER9:
                006406  010537  001252                  MOV     R5,@#$TMP10
                006412  010437  001254                  MOV     R4,@#$TMP11
                006416  012737  006652  001240          MOV     #HHP5,@#$TMP3
                006424  012737  006652  001242          MOV     #HHP5,@#$TMP4
                006432  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
                006440  012737  006732  001246          MOV     #HHP11,@#$TMP6
                006446  104207                  1$:     ERROR   +207
                006450  000137  006742                  JMP     @#HHDONE
     1066       006454                          HHER10:
                006454  010537  001252                  MOV     R5,@#$TMP10
                006460  010437  001254                  MOV     R4,@#$TMP11
                006464  012737  006652  001240          MOV     #HHP5,@#$TMP3
                006472  012737  006652  001242          MOV     #HHP5,@#$TMP4
                006500  012737  006572  001244          MOV     #HHDAT0,@#$TMP5
                006506  012737  006732  001246          MOV     #HHP11,@#$TMP6
                006514  104213                  1$:     ERROR   +213
                006516  000137  006742                  JMP     @#HHDONE
     1067       006522  013703  001236          HHTRAP: MOV     @#$TMP2,R3      ;WAS THE TRAP TO 244
     1068       006526  062703  000002                  ADD     #2,R3           ;ON THE INSTRUCTION
     1069       006532  020316                          CMP     R3,(SP)         ;BEING TESTED?
     1070       006534  001402                          BEQ     1$
     1071       006536  000137  036660                  JMP     @#FPSPUR
     1072       006542  011637  001236          1$:     MOV     (SP),@#$TMP2    ;FAILURE OF FPS INTERRUPT
     1073                                                                       ;DISABLE BIT (FID=1)
     1074       006546  022626                          CMP     (SP)+,(SP)+     ;TO INHIBIT TRAP.
     1075       006550  170201                          STFPS   R1
     1076       006552  010137  001240                  MOV     R1,@#$TMP3
     1077       006556  170301                          STST    R1
     1078       006560  010137  001242                  MOV     R1,@#$TMP4
     1079       006564  104217                  2$:     ERROR   +217
     1080       006566  000137  006742                  JMP     @#HHDONE
     1081       006572  000000                  HHDATO: 0
     1082       006574  000000                          0
     1083       006576  000000                          0
     1084       006600  000000                          0
     1085       006602  000452                  HHP0:   452
     1086       006604  125252                          125252
     1087       006606  125252                          125252
     1088       006610  125253                          125253
     1089       006612  000252                  HHP1:   252
     1090       006614  125252                          125252
     1091       006616  125252                          125252
     1092       006620  125252                          125252
     1093       006622  000600                  HHP2:   600                     ;HHP0 + HHP1 WITH
     1094       006624  000000                          0                       ;PROPER NORMALIZATION
     1095       006626  000000                          0
     1096       006630  000000                          0
     1097       006632  000400                  HHP3:   400                     ;HHP0 + HHP1 WITH
     1098       006634  000000                          0                       ;BAD NORMALIZATION
     1099       006636  000000                          0
     1100       006640  000000                          0
     1101       006642  000000                  HHP4:   0
     1102       006644  000000                          0
```

```
1103 006646  000000                          0
1104 006650  000000                          0
1105 006652  100200              HHP5:       100200
1106 006654  000000                          0
1107 006656  000000                          0
1108 006660  000000                          0
1109 006662  000300              HHP6:       300
1110 006664  000000                          0
1111 006666  000000                          0
1112 006670  000000                          0
1113 006672  100000              HHP7:       100000              ;HHP7 = HHP8 + HHP9
1114 006674  000000                          0                   ;      = HHP5 + HHP6
1115 006676  000000                          0
1116 006700  000000                          0
1117 006702  000200              HHP8:       200
1118 006704  000000                          0
1119 006706  000000                          0
1120 006710  000000                          0
1121 006712  100300              HHP9:       100300
1122 006714  000000                          0
1123 006716  000000                          0
1124 006720  000000                          0
1125 006722  000400              HHP10:      400                 ;HHP10 = HHP8 + HHP8
1126 006724  000000                          0
1127 006726  000000                          0
1128 006730  000000                          0
1129 006732  100400              HHP11:      100400              ;HHP11 = HHP5 + HHP5
1130 006734  000000                          0
1131 006736  000000                          0
1132 006740  000000                          0
1133 006742                      HHDONE:
     006742  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?).
1134
1148
1149                                 ;;*****************************************************************
                                     ;*TEST 2        OVER\UNDER TEST
                                     ;*
                                     ;*THIS IS A PARTIAL TEST OF THE OVER\UNDER
                                     ;*FLOWS.  ONE OVERFLOW AND TWO UNDERFLOW
                                     ;*CONDITIONS ARE CHECKED.  THE REMAINING
                                     ;*UNDERFLOW COND. AND THE REMAINING OVERFLOW
                                     ;*COND. WILL BE CHECKED LATER USING THE
                                     ;*XXX INSTRUCTION.  HERE EACH CONDITION TESTED
                                     ;*IS CHECKED BOTH WITH TRAPS ENABLED
                                     ;*(FIU=1 OR FIV=1) AND ALSO WITH TRAPS
                                     ;*DISABLED (FIU=0 OR FIV=0).
                                     ;*
                                     ;;*****************************************************************
     006744  000004                  TST2:   SCOPE
1150
1151
1152 006746                          ;TEST OVERFLOW CONDITION WITH TRAP DISABLER FIV=0
                                     GG1:
     006746  104413                          LPERR               ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
1153 006750  012704  000200                    MOV     #200,R4          ;CLEAR FIU, FIV, AND SET FD
1154 006754  170104                            LDFPS   R4
1155 006756  012737  007004  001236            MOV     #GG2,@#$TMP2
1156 006764  012737  010036  000244            MOV     #GGER0,@#FPVECT
1157 006772  012700  011610                    MOV     #GGP5,R0         ;SET AC0 OPERAND
1158 006776  172410                            LDD     (R0),AC0
1159 007000  012700  011610                    MOV     #GGP5,R0         ;FSRC
1160 007004  172010                GG2:        ADDD    (R0),AC0         ;TEST INSTRUCTION
1161 007006  170205                            STFPS   R5               ;GET FPS
1162 007010  012700  011540                    MOV     #GGDAT0,R0       ;GET THE RESULT
1163 007014  174010                            STD     AC0,(R0)
1164 007016  012701  011620                    MOV     #GGP6,R1         ;IS IT CORRECT
1165 007022  012702  000004                    MOV     #4,R2
1166 007026  022021                GG3:        CMP     (R0)+,(R1)+
1167 007030  001402                            BEQ     GG4
1168 007032  000137  010134                    JMP     @#GGER1
1169 007036  077205                GG4:        SOB     R2,GG3
1170 007040  052704  000006                    BIS     #6,R4            ;FPS CORRECT?
1171 007044  020405                            CMP     R4,R5
1172 007046  001402                            BEQ     GG5
1173 007050  000137  010202                    JMP     @#GGER2
1174                                           ;TEST OVERFLOW WITH TRAPS ENABLED
1175                                           ;FIV = 1
1176 007054                        GG5:
     007054  104413                            LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
1177 007056  012704  001200                    MOV     #1200,R4         ;CLEAR FIU, SET FIV, AND FD
1178 007062  170104                            LDFPS   R4
1179 007064  012737  007112  001236            MOV     #GG6,@#$TMP2
1180 007072  012737  007130  000244            MOV     #GG7,@#FPVECT
1181 007100  012700  011610                    MOV     #GGP5,R0         ;SET AC0 OPERAND
1182 007104  172410                            LDD     (R0),AC0
1183 007106  012700  011610                    MOV     #GGP5,R0         ;FSPC
1184 007112  172010                GG6:        ADDD    (R0),AC0         ;TEST INSTRUCTION
1185 007114  170000                            CFCC
1186 007116  012700  011540                    MOV     #GGDAT0,R0
1187 007122  174010                            STD     AC0,(R0)
1188 007124  000137  010250                    JMP     @#GGER3
1189 007130  013703  001236                GG7:MOV     @#$TMP2,R3
1190 007134  062703  000002                    ADD     #2,R3
1191 007140  020316                            CMP     R3,(SP)
1192 007142  001402                            BEQ     1$
1193 007144  000137  036660                    JMP     @#FPSPUR
1194 007150  011637  001236                1$: MOV     (SP),@#$TMP2
1195 007154  022626                            CMP     (SP)+,(SP)+
1196 007156  170205                            STFPS   R5
1197 007160  012700  011540                    MOV     #GGDAT0,R0       ;GET THE RESULT
1198 007164  174010                            STD     AC0,(R0)
1199 007166  012701  011620                    MOV     #GGP6,R1         ;IS IT CORRECT
1200 007172  012702  000004                    MOV     #4,R2
1201 007176  022021                GG8:        CMP     (R0)+,(R1)+
1202 007200  001402                            BEQ     GG9
1203 007202  000137  010316                    JMP     @#GGER4
1204 007206  077205                GG9:        SOB     R2,GG8
1205 007210  052704  100006                    BIS     #100006,R4
1206 007214  020405                            CMP     R4,R5            ;FPS CORRECT?
1207 007216  001402                            BEQ     1$
1208 007220  000137  010366                    JMP     @#GGER6
```

```
1209 007224  012704  000010          1$:     MOV     #10,R4
1210                                 ;CHECK FEC
1211 007230  170305                          STST    R5
1212 007232  020405                          CMP     R4,R5
1213 007234  001402                          BEQ     GG10
1214 007236  000137  010320                  JMP     @#GGER5
1215                                 ;CHECK UNDER FLOW CONDITION WITH
1216                                 ;TRAPS DISABLED (FIU = 0)
1217 007242                          GG10:
     007242  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1218 007244  012704  000200                  MOV     #0200,R4        ;SET FIU, FIV, AND FD
1219 007250  170104                          LDFPS   R4
1220 007252  012737  007300  001236          MOV     #GG11,@#$TMP2
1221 007260  012737  010434  000244          MOV     #GGER7,@#FPVECT
1222 007266  012700  011560                  MOV     #GGP2,R0        ;SET AC0 OPERAND
1223 007272  172410                          LDD     (R0),AC0        ;FSRC
1224 007274  012700  011570                  MOV     #GGP3,R0
1225 007300  172010                  GG11:   ADDD    (R0),AC0        ;TEST INSTRUCTION
1226 007302  170205                          STFPS   R5              ;GET FPS
1227 007304  012700  011540                  MOV     #GGDAT0,R0      ;GET THE RESULT
1228 007310  174010                          STD     AC0,(R0)
1229 007312  012701  011620                  MOV     #GGP6,R1        ;IS IT CORRECT
1230 007316  012702  000004                  MOV     #4,R2
1231 007322  022021                  GG12:   CMP     (R0)+,(R1)+
1232 007324  001402                          BEQ     GG13
1233 007326  000137  010532                  JMP     @#GGER8
1234 007332  077205                  GG13:   SOB     R2,GG12
1235 007334  052704  000004                  BIS     #4,R4           ;FPS CORRECT?
1236 007340  020405                          CMP     R4,R5
1237 007342  001402                          BEQ     GG14
1238 007344  000137  010600                  JMP     @#GGER9
1239                                 ;CHECK UNDERFLOW CONDITION WITH
1240                                 ;TRAP ENABLED (FIU = 1)
1241 007350                          GG14:
     007350  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1242 007352  012704  002200                  MOV     #2200,R4        ;SET FIU, FIV, AND FD
1243 007356  170104                          LDFPS   R4
1244 007360  012737  007406  001236          MOV     #GG15,@#$TMP2
1245 007366  012737  007424  000244          MOV     #GG16,@#FPVECT
1246 007374  012700  011560                  MOV     #GGP2,R0        ;SET AC0 OPERAND
1247 007400  172410                          LDD     (R0),AC0        ;FSPC
1248 007402  012700  011570                  MOV     #GGP3,R0
1249 007406  172010                  GG15:   ADDD    (R0),AC0        ;TEST INSTRUCTION
1250 007410  170000                          CFCC
1251 007412  012700  011540                  MOV     #GGDAT0,R0
1252 007416  174010                          STD     AC0,(R0)
1253 007420  000137  010646                  JMP     @#GGER10
1254 007424  013703  001236          GG16:   MOV     @#$TMP2,R3
1255 007430  062703  000002                  ADD     #2,R3
1256 007434  021603                          CMP     (SP),R3
1257 007436  001402                          BEQ     1$
1258 007440  000137  036660                  JMP     @#FPSPUR
1259 007444  011637  001236          1$:     MOV     (SP),@#$TMP2
1260 007450  022626                          CMP     (SP)+,(SP)+
1261 007452  170205                          STFPS   R5              ;GET FPS
1262 007454  012700  011540                  MOV     #GGDAT0,R0      ;GET THE RESULT
1263 007460  174010                          STD     AC0,(R0)
```

```
 1264 007462  012701  011630                MOV     #GGP7,R1        ;IS IT CORRECT
 1265 007466  012702  000004                MOV     #4,R2
 1266 007472  022021                GG17:   CMP     (R0)+,(R1)+
 1267 007474  001402                        BEQ     GG18
 1268 007476  000137  010714                JMP     @#GGER11
 1269 007502  077205                GG18:   SOB     R2,GG17
 1270 007504  052704  100000                BIS     #100000,R4
 1271 007510  020405                        CMP     R4,R5           ;FPS CORRECT?
 1272 007512  001402                        BEQ     2$
 1273 007514  000137  010762                JMP     @#GGER12
 1274 007520                        2$:
 1275 007520  012704  000012        1$:     MOV     #12,R4
 1276                               ;CHECK FEC
 1277 007524  170305                        STST    R5
 1278 007526  020405                        CMP     R4,R5
 1279 007530  001402                        BEQ     GG19
 1280 007532  000177  001272                JMP     @GGER13
 1281                               ;CHECK UNDERFLOW CONDITION WITH TRAPS
 1282                               ;DISABLED (FIU = 0)
 1283 007536                        GG19:
    . 007536  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 1284 007540  012704  000200                MOV     #0200,R4        ;SET FIU, FIV, AND FD
 1285 007544  170104                        LDFPS   R4
 1286 007546  012737  007574  001236        MOV     #GG20,@#$TMP2
 1287 007554  012737  011076  000244        MOV     #GGER14,@#FPVECT
 1288 007562  012700  011560                MOV     #GGP2,R0        ;SET AC0 OPERAND
 1289 007566  172410                        LDD     (R0),AC0
 1290 007570  012700  011640                MOV     #GGP8,R0        ;FSPC
 1291 007574  172010                GG20:   ADDD    (R0),AC0        ;TEST INSTRUCTION
 1292 007576  170205                        STFPS   R5              ;GET FPS
 1293 007600  012700  011540                MOV     #GGDAT0,R0      ;GET THE RESULT
 1294 007604  174010                        STD     AC0,(R0)
 1295 007606  012701  011620                MOV     #GGP6,R1        ;IS IT CORRECT
 1296 007612  012702  000004                MOV     #4,R2
 1297 007616  022021                GG21:   CMP     (R0)+,(R1)+
 1298 007620  001402                        BEQ     GG22
 1299 007622  000137  011174                JMP     @#GGER15
 1300 007626  077205                GG22:   SOB     R2,GG21
 1301 007630  052704  000004                BIS     #4,R4           ;FPS CORRECT?
 1302 007634  020405                        CMP     R4,R5
 1303 007636  001402                        BEQ     GG23
 1304 007640  000137  011242                JMP     @#GGER16
```

```
1306                                    ;CHECK UNDERFLOW CONDITION WITH TRAP
```

```
1308                                            ;ENABLED (FIU = 1)
1309 007644                             GG23:
     007644  104413                             LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1310 007646  012704  002200                    MOV     #2200,R4        ;SET FIU, FIV, AND FD
1311 007652  170104                            LDFPS   R4
1312 007654  012737  007702  001236            MOV     #GG24,@#$TMP2
1313 007662  012737  007720  000244            MOV     #GG25,@#FPVECT
1314 007670  012700  011560                    MOV     #GGP2,R0        ;SET AC0 OPERAND
1315 007674  172410                            LDD     (R0),AC0
1316 007676  012700  011640                    MOV     #GGP8,R0        ;FSRC
1317 007702  172010                     GG24:  ADDD    (R0),AC0        ;TEST INSTRUCTION
1318 007704  170000                            CFCC
1319 007706  012700  011540                    MOV     #GGDAT0,R0
1320 007712  174010                            STD     AC0,(R0)
1321 007714  000137  011310                    JMP     @#GGER17
1322 007720  013700  001236            GG25:  MOV     @#$TMP2,R0
1323 007724  062700  000002                    ADD     #2,R0
1324 007730  020016                            CMP     R0,(SP)
1325 007732  001402                            BEQ     1$
1326 007734  000137  036660                    JMP     @#FPSPUR
1327 007740  011637  001236            1$:    MOV     (SP),@#$TMP2
1328 007744  022626                            CMP     (SP)+,(SP)+
1329 007746  170205                            STFPS   R5              ;GET FPS
1330 007750  012700  011540                    MOV     #GGDAT0,R0      ;GET THE RESULT
1331 007754  174010                            STD     AC0,(R0)
1332 007756  012701  011650                    MOV     #GGP9,R1        ;IS IT CORRECT
1333 007762  012702  000004                    MOV     #4,R2
1334 007766  022021                     GG26:  CMP     (R0)+,(R1)+
1335 007770  001402                            BEQ     GG27
1336 007772  000137  011356                    JMP     @#GGER18
1337 007776  077205                     GG27:  SOB     R2,GG26
1338 010000  052704  100004                    BIS     #100004,R4
1339 010004  020405                            CMP     R4,R5           ;FPS CORRECT?
1340 010006  001402                            BEQ     1$
1341 010010  000167  001456                    JMP     GGER20
1342 010014  012704  000012            1$:    MOV     #12,R4
1343                                            ;CHECK FEC
1344 010020  170305                            STST    R5
1345 010022  020405                            CMP     R4,R5
1346 010024  001402                            BEQ     GG28
1347 010026  000167  001372                    JMP     GGER19
1348
1349 010032  000137  011660             GG28:  JMP     @#GGDONE
1350
1351 010036  013701  001236             GGER0: MOV     @#$TMP2,R1
1352 010042  062701  000002                    ADD     #2,R1
1353 010046  020116                            CMP     R1,(SP)
1354 010050  001402                            BEQ     10$
1355 010052  000137  036660             5$:    JMP     @#FPSPUR
1356 010056                             10$:
1357 010056  170301                            STST    R1
1358 010060  020127  000010                    CMP     R1,#10
1359 010064  001372                            BNE     5$
1360 010066  022626                            CMP     (SP)+,(SP)+
1361 010070  012700  011540                    MOV     #GGDAT0,R0
1362 010074  174010                            STD     AC0,(R0)
1363 010076  012737  011610  001240            MOV     #GGP5,@#$TMP3
```

```
                010104  012737  011610  001242          MOV     #GGP5,@#$TMP4
                010112  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                010120  012737  011620  001246          MOV     #GGP6,@#$TMP6
                010126  104220                  1$:     ERROR   +220
                010130  000137  011660                  JMP     @#GGDONE
          1364
          1365  010134                          GGER1:
                010134  010537  001252                  MOV     R5,@#$TMP10
                010140  010437  001254                  MOV     R4,@#$TMP11
                010144  012737  011610  001240          MOV     #GGP5,@#$TMP3
                010152  012737  011610  001242          MOV     #GGP5,@#$TMP4
                010160  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                010166  012737  011620  001246          MOV     #GGP6,@#$TMP6
                010174  104207                  1$:     ERROR   +207
                010176  000137  011660                  JMP     @#GGDONE
          1366
          1367  010202                          GGER2:
                010202  010537  001252                  MOV     R5,@#$TMP10
                010206  010437  001254                  MOV     R4,@#$TMP11
                010212  012737  011610  001240          MOV     #GGP5,@#$TMP3
                010220  012737  011610  001242          MOV     #GGP5,@#$TMP4
                010226  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                010234  012737  011620  001246          MOV     #GGP6,@#$TMP6
                010242  104232                  1$:     ERROR   +232
                010244  000137  011660                  JMP     @#GGDONE
          1368
          1369  010250                          GGER3:
                010250  010537  001252                  MOV     R5,@#$TMP10
                010254  010437  001254                  MOV     R4,@#$TMP11
                010260  012737  011610  001240          MOV     #GGP5,@#$TMP3
                010266  012737  011610  001242          MOV     #GGP5,@#$TMP4
                010274  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                010302  012737  011620  001246          MOV     #GGP6,@#$TMP6
                010310  104221                  1$:     ERROR   +221
                010312  000137  011660                  JMP     @#GGDONE
          1370
          1371  010316  000706                  GGER4:  BR      GGER1
          1372
          1373  010320                          GGER5:
                010320  010537  001252                  MOV     R5,@#$TMP10
                010324  010437  001254                  MOV     R4,@#$TMP11
                010330  012737  011610  001240          MOV     #GGP5,@#$TMP3
                010336  012737  011610  001242          MOV     #GGP5,@#$TMP4
                010344  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                010352  012737  011620  001246          MOV     #GGP6,@#$TMP6
                010360  104226                  1$:     ERROR   +226
                010362  000137  011660                  JMP     @#GGDONE
          1374
          1375  010366                          GGER6:
                010366  010537  001252                  MOV     R5,@#$TMP10
                010372  010437  001254                  MOV     R4,@#$TMP11
                010376  012737  011610  001240          MOV     #GGP5,@#$TMP3
                010404  012737  011610  001242          MOV     #GGP5,@#$TMP4
                010412  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                010420  012737  011620  001246          MOV     #GGP6,@#$TMP6
                010426  104227                  1$:     ERROR   +227
                010430  000137  011660                  JMP     @#GGDONE
```

```
1376
1377 010434  013701  001236          GGER7:  MOV     @#$TMP2,R1
1378 010440  062701  000002                  ADD     #2,R1
1379 010444  020116                          CMP     R1,(SP)
1380 010446  001402                          BEQ     10$
1381 010450  000137  036660          5$:     JMP     @#FPSPUR
1382 010454                          10$:
1383 010454  170301                          STST    R1
1384 010456  020127  000012                  CMP     R1,#12
1385 010462  001372                          BNE     5$
1386 010464  022626                          CMP     (SP)+,(SP)+
1387 010466  012700  011540                  MOV     #GGDATO,R0
1388 010472  174010                          STD     AC0,(R0)
1389 010474  012737  011570  001240          MOV     #GGP3,@#$TMP3
     010502  012737  011560  001242          MOV     #GGP2,@#$TMP4
     010510  012737  011540  001244          MOV     #GGDATO,@#$TMP5
     010516  012737  011620  001246          MOV     #GGP6,@#$TMP6
     010524  104224                  1$:     ERROR   +224
     010526  000137  011660                  JMP     @#GGDONE
1390
1391 010532                          GGER8:
     010532  010537  001252                  MOV     R5,@#$TMP10
     010536  010437  001254                  MOV     R4,@#$TMP11
     010542  012737  011570  001240          MOV     #GGP3,@#$TMP3
     010550  012737  011560  001242          MOV     #GGP2,@#$TMP4
     010556  012737  011540  001244          MOV     #GGDATO,@#$TMP5
     010564  012737  011620  001246          MOV     #GGP6,@#$TMP6
     010572  104207                  1$:     ERROR   +207
     010574  000137  011660                  JMP     @#GGDONE
1392
1393 010600                          GGER9:
     010600  010537  001252                  MOV     R5,@#$TMP10
     010604  010437  001254                  MOV     R4,@#$TMP11
     010610  012737  011570  001240          MOV     #GGP3,@#$TMP3
     010616  012737  011560  001242          MOV     #GGP2,@#$TMP4
     010624  012737  011540  001244          MOV     #GGDATO,@#$TMP5
     010632  012737  011620  001246          MOV     #GGP6,@#$TMP6
     010640  104232                  1$:     ERROR   +232
     010642  000137  011660                  JMP     @#GGDONE
1394
1395 010646                          GGER10:
     010646  010537  001252                  MOV     R5,@#$TMP10
     010652  010437  001254                  MOV     R4,@#$TMP11
     010656  012737  011570  001240          MOV     #GGP3,@#$TMP3
     010664  012737  011560  001242          MOV     #GGP2,@#$TMP4
     010672  012737  011540  001244          MOV     #GGDATO,@#$TMP5
     010700  012737  011630  001246          MOV     #GGP7,@#$TMP6
     010706  104225                  1$:     ERROR   +225
     010710  000137  011660                  JMP     @#GGDONE
1396
1397 010714                          GGER11:
     010714  010537  001252                  MOV     R5,@#$TMP10
     010720  010437  001254                  MOV     R4,@#$TMP11
     010724  012737  011570  001240          MOV     #GGP3,@#$TMP3
     010732  012737  011560  001242          MOV     #GGP2,@#$TMP4
     010740  012737  011540  001244          MOV     #GGDATO,@#$TMP5
     010746  012737  011630  001246          MOV     #GGP7,@#$TMP6
```

```
        010754  104207                  1$:     ERROR   +207
        010756  000137  011660                  JMP     @#GGDONE
  1398
  1399  010762                          GGER12:
        010762  010537  001252                  MOV     R5,@#$TMP10
        010766  010437  001254                  MOV     R4,@#$TMP11
        010772  012737  011570  001240          MOV     #GGP3,@#$TMP3
        011000  012737  011560  001242          MOV     #GGP2,@#$TMP4
        011006  012737  011540  001244          MOV     #GGDATO,@#$TMP5
        011014  012737  011630  001246          MOV     #GGP7,@#$TMP6
        011022  104231                  1$:     ERROR   +231
        011024  000137  011660                  JMP     @#GGDONE
  1400
  1401  011030                          GGER13:
        011030  010537  001252                  MOV     R5,@#$TMP10
        011034  010437  001254                  MOV     R4,@#$TMP11
        011040  012737  011570  001240          MOV     #GGP3,@#$TMP3
        011046  012737  011560  001242          MOV     #GGP2,@#$TMP4
        011054  012737  011540  001244          MOV     #GGDATO,@#$TMP5
        011062  012737  011630  001246          MOV     #GGP7,@#$TMP6
        011070  104230                  1$:     ERROR   +230
        011072  000137  011660                  JMP     @#GGDONE
  1402
  1403  011076  013701  001236          GGER14: MOV     @#$TMP2,R1
  1404  011102  062701  000002                  ADD     #2,R1
  1405  011106  020116                          CMP     R1,(SP)
  1406  011110  001402                          BEQ     10$
  1407  011112  000137  036660          5$:     JMP     @#FPSPUR
  1408  011116                          10$:
  1409  011116  170301                          STST    R1
  1410  011120  020127  000012                  CMP     R1,#12
  1411  011124  001372                          BNE     5$
  1412  011126  022626                          CMP     (SP)+,(SP)+
  1413  011130  012700  011540                  MOV     #GGDATO,R0
  1414  011134  174010                          STD     AC0,(R0)
  1415
  1416  011136  012737  011550  001240          MOV     #GGP1,@#$TMP3
        011144  012737  011570  001242          MOV     #GGP3,@#$TMP4
        011152  012737  011540  001244          MOV     #GGDATO,@#$TMP5
        011160  012737  011620  001246          MOV     #GGP6,@#$TMP6
        011166  104222                  1$:     ERROR   +222
        011170  000137  011660                  JMP     @#GGDONE
  1417
  1418  011174                          GGER15:
        011174  010537  001252                  MOV     R5,@#$TMP10
        011200  010437  001254                  MOV     R4,@#$TMP11
        011204  012737  011560  001240          MOV     #GGP2,@#$TMP3
        011212  012737  011640  001242          MOV     #GGP8,@#$TMP4
        011220  012737  011540  001244          MOV     #GGDATO,@#$TMP5
        011226  012737  011620  001246          MOV     #GGP6,@#$TMP6
        011234  104207                  1$:     ERROR   +207
        011236  000137  011660                  JMP     @#GGDONE
  1419
  1420  011242                          GGER16:
        011242  010537  001252                  MOV     R5,@#$TMP10
        011246  010437  001254                  MOV     R4,@#$TMP11
        011252  012737  011560  001240          MOV     #GGP2,@#$TMP3
```

```
                011260  012737  011640  001242          MOV     #GGP8,@#$TMP4
                011266  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                011274  012737  011620  001246          MOV     #GGP6,@#$TMP6
                011302  104232                  1$:     ERROR   +232
                011304  000137  011660                  JMP     @#GGDONE
          1421
          1422  011310                          GGER17:
                011310  010537  001252                  MOV     R5,@#$TMP10
                011314  010437  001254                  MOV     R4,@#$TMP11
                011320  012737  011560  001240          MOV     #GGP2,@#$TMP3
                011326  012737  011640  001242          MOV     #GGP8,@#$TMP4
                011334  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                011342  012737  011650  001246          MOV     #GGP9,@#$TMP6
                011350  104223                  1$:     ERROR   +223
                011352  000137  011660                  JMP     @#GGDONE
          1423
          1424  011356                          GGER18:
                011356  010537  001252                  MOV     R5,@#$TMP10
                011362  010437  001254                  MOV     R4,@#$TMP11
                011366  012737  011560  001240          MOV     #GGP2,@#$TMP3
                011374  012737  011640  001242          MOV     #GGP8,@#$TMP4
                011402  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                011410  012737  011650  001246          MOV     #GGP9,@#$TMP6
                011416  104207                  1$:     ERROR   +207
                011420  000137  011660                  JMP     @#GGDONE
          1425
          1426  011424                          GGER19:
                011424  010537  001252                  MOV     R5,@#$TMP10
                011430  010437  001254                  MOV     R4,@#$TMP11
                011434  012737  011560  001240          MOV     #GGP2,@#$TMP3
                011442  012737  011640  001242          MOV     #GGP8,@#$TMP4
                011450  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                011456  012737  011650  001246          MOV     #GGP9,@#$TMP6
                011464  104230                  1$:     ERROR   +230
                011466  000137  011660                  JMP     @#GGDONE
          1427
          1428  011472                          GGER20:
                011472  010537  001252                  MOV     R5,@#$TMP10
                011476  010437  001254                  MOV     R4,@#$TMP11
                011502  012737  011560  001240          MOV     #GGP2,@#$TMP3
                011510  012737  011640  001242          MOV     #GGP8,@#$TMP4
                011516  012737  011540  001244          MOV     #GGDAT0,@#$TMP5
                011524  012737  011650  001246          MOV     #GGP9,@#$TMP6
                011532  104231                  1$:     ERROR   +231
                011534  000137  011660                  JMP     @#GGDONE
          1429
          1430  011540  000000                  GGDAT0: 0
          1431  011542  000000                          0
          1432  011544  000000                          0
          1433  011546  000000                          0
          1434
          1435  011550  000300                  GGP1:   300
          1436  011552  000000                          0
          1437  011554  000000                          0
          1438  011556  000000                          0
          1439  011560  100200                  GGP2:   100200
          1440  011562  000000                          0
```

```
1441 011564  000000                          0
1442 011566  000000                          0
1443 011570  000200                GGP3:     200
1444 011572  000000                          0
1445 011574  000000                          0
1446 011576  000001                          1
1447 011600  010200                GGP4:     10200
1448 011602  000000                          0
1449 011604  000000                          0
1450 011606  000000                          0
1451 011610  077600                GGP5:     77600          ;OVER FLOW = GGP5 + GGP5
1452 011612  000000                          0
1453 011614  000000                          0
1454 011616  000000                          0
1455 011620  000000                GGP6:     0              ;OVERFLOW RESULT
1456 011622  000000                          0              ;UNDERFLOW RESULT
1457 011624  000000                          0              ;GGP6 = GGP4 + GGP5
1458 011626  000000                          0              ;      = GGP3 + GGP2 (FIU = 0)
1459                                                        ;      = GGP3 + GGP1
1460 011630  062400                GGP7:     62400          ;GGP7 = GGP3 + GGP2 (FIU = 1)
1461 011632  000000                          0
1462 011634  000000                          0
1463 011636  000000                          0
1464 011640  000340                GGP8:     340
1465 011642  000000                          0
1466 011644  000000                          0
1467 011646  000000                          0
1468 011650  000100                GGP9:     100
1469 011652  000000                          0
1470 011654  000000                          0
1471 011656  000000                          0
1472 011660                        GGDONE:
     011660  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?).
1473
1479
1480                                   ;;*****************************************************************
                                       ;*TEST 3        LDCFD AND LDCDF TEST
                                       ;*
                                       ;*THIS IS A TEST OF LDCFD AND LDCDF.
                                       ;*
                                       ;;*****************************************************************
     011662  000004                    TST3:    SCOPE
1481                                   ;TEST FOR CORRECT AUTO INCREMENT CONSTANT.
1482 011664                            HX1:
     011664  104413                          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1483 011666  012704  000200                   MOV     #200,R4
1484 011672  170104                           LDFPS   R4
1485 011674  012700  013300                   MOV     #HXP1,R0
1486 011700  172410                           LDD     (R0),AC0
1487 011702  012700  013310                   MOV     #HXP2,R0
1488 011706  012737  011714  001236           MOV     #HX2,@#$TMP2
1489 011714  177420                    HX2:   LDCFD   (R0)+,AC0
1490 011716  020027  013314                   CMP     R0,#HXP2+4     ;IS R0 CORRECT
```

```
1491 011722  001402                        BEQ     HX3
1492 011724  000137  012656                JMP     @#HXER1
1493 011730                          HX3:
1494 011730  170205                        STFPS   R5              ;GET FPS
1495 011732  012700  013270                MOV     #HXDATO,R0
1496 011736  174010                        STD     AC0,(R0)        ;GET AC0
1497 011740  012701  013360                MOV     #HXP7,R1        ;SEE IF RESULT IS
1498 011744  012702  000004                MOV     #4,R2           ;CORRECT
1499 011750  022120                  HX4:  CMP     (R1)+,(R0)+
1500 011752  001415                        BEQ     HX7
1501 011754  012701  013310                MOV     #HXP2,R1        ;DID FD GET
1502 011760  012700  013270                MOV     #HXDATO,R0      ;COMPLIMENTED?
1503 011764  012702  000004                MOV     #4,R2
1504 011770  022120                  HX5:  CMP     (R1)+,(R0)+
1505 011772  001402                        BEQ     HX6
1506 011774  000137  012716                JMP     @#HXER2
1507 012000  077205                  HX6:  SOB     R2,HX5
1508 012002  000137  012746                JMP     @#HXER3
1509 012006  077220                  HX7:  SOB     R2,HX4
1510 012010  012704  000200                MOV     #200,R4         ;FPS CORRECT?
1511 012014  020405                        CMP     R4,R5
1512 012016  001402                        BEQ     HX8
1513 012020  000137  013014                JMP     @#HXER8
1514                                ;NOW    TEST    LDCDF
1515 012024                          HX8:
     012024  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1516 012026  012704  000200                        MOV #200,R4
1517 012032  170104                        LDFPS R4
1518
1519 012034  012700  013300                MOV     #HXP1,R0
1520 012040  172410                        LDD     (R0),AC0
1521
1522 012042  012700  013310                MOV     #HXP2,R0
1523 012046  012737  012056  001236        MOV     #HX9,@#$TMP2
1524
1525 012054  170001                        SETF
1526
1527 012056  177420                  HX9:  LDCDF   (R0)+,AC0       ;TEST INSTRUCTION
1528
1529 012060  020027  013320                CMP     R0,#HXP2+10     ;WAS A GOOD
1530 012064  001402                        BEQ     HX10            ;CONSTANT USED
1531 012066  000137  012676                JMP     @#HXER5         ;TO INCREMENT R0?
1532
1533 012072                          HX10:
1534 012072  170205                        STFPS   R5
1535 012074  012700  013270                MOV     #HXDATO,R0
1536 012100  170011                        SETD
1537 012102  174010                        STD     AC0,(R0)        ;GET RESULT
1538 012104  012701  013370                MOV     #HXP8,R1
1539 012110  012702  000004                MOV     #4,R2
1540 012114  022120                  HX11: CMP     (R1)+,(R0)+     ;IS IT CORRECT?
1541 012116  001415                        BEQ     HX14
1542
1543 012120  012701  013360                MOV     #HXP7,R1
1544 012124  012700  013270                MOV     #HXDATO,R0
1545 012130  012702  000004                MOV     #4,R2
1546 012134  022110                  HX12: CMP     (R1)+,(R0)      ;DID FD FAIL TO GET
```

```
1547 012136  001402                        BEQ     HX13            ;COMPLIMENTED?
1548 012140  000137  013032                JMP     @#HXER6
1549 012144  077205                HX13:   SOB     R2,HX12
1550 012146  000137  013062                JMP     @#HXER7
1551
1552 012152  077220                HX14:   SOB     R2,HX11
1553
1554 012154  012704  000000                MOV     #0,R4           ;FPS CORRECT?
1555 012160  020405                        CMP     R4,R5
1556 012162  001402                        BEQ     HX15
1557 012164  000137  013014                JMP     @#HXER8
1558
1559                               ;TEST GR7 IMMEDIATE MODE CONSTANT
1560
1561 012170                        HX15:
     012170  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1562
1563 012172  012704  000200                MOV     #200,R4
1564 012176  170104                        LDFPS   R4              ;SET FD
1565 012200  012737  012216  001236        MOV     #HX16,@#$TMP2
1566 012206  012737  013112  000004        MOV     #HXER9,@#ERRVECT
1567 012214  005001                        CLR     R1
1568 012216  177427  043243        HX16:   LDCFD   #5201,AC0
1569 012222  005201                HX165:  INC     R1
1570 012224  005201                        INC     R1
1571 012226  005201                        INC     R1
1572 012230  012737  036712  000004        MOV     #CPSPUR,@#ERRVECT
1573 012236  020127  000003                CMP     R1,#3           ;SEE IF PC WAS
1574 012242  001402                        BEQ     HX17            ;CORRECT
1575 012244  000137  013146                JMP     @#HXER10
1576
1577 012250                        HX17:
     012250  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1578
1579 012252  012704  000200                MOV     #200,R4
1580 012256  170104                        LDFPS   R4
1581 012260  012737  012306  001236        MOV     #HX18,@#$TMP2
1582 012266  012700  013350                MOV     #HXP6,R0
1583 012272  172410                        LDD     (R0),AC0
1584 012274  012737  036712  000004        MOV     #CPSPUR,@#ERRVECT
1585 012302  012700  013310                MOV     #HXP2,R0
1586 012306  177410                HX18:   LDCFD   (R0),AC0
1587
1588 012310  012700  013270                MOV     #HXDAT0,R0
1589 012314  174010                        STD     AC0,(R0)        ;GET RESULT.
1590 012316  012701  013360                MOV     #HXP7,R1
1591 012322  012702  000004                MOV     #4,R2
1592 012326  022021                HX19:   CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
1593 012330  001402                        BEQ     HX20
1594 012332  000137  012716                JMP     @#HXER2
1595 012336  077205                HX20:   SOB     R2,HX19
1596
1597                               ;TEST LDCFD WITH NEGATIVE OPERAND
1598 012340                        HX21:
     012340  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1599 012342  012704  000200                MOV     #200,R4
1600 012346  170104                        LDFPS   R4
```

```
1601 012350 012737 012370 001236        MOV     #HX22,@#$TMP2
1602 012356 012700 013350               MOV     #HXP6,R0
1603 012362 172410                      LDD     (R0),AC0
1604 012364 012700 013330               MOV     #HXP4,R0
1605 012370 177410              HX22:   LDCFD   (R0),AC0
1606
1607 012372 012700 013270               MOV     #HXDAT0,R0
1608 012376 174010                      STD     AC0,(R0)        ;GET RESULT
1609
1610 012400 012701 013340               MOV     #HXP5,R1
1611 012404 012702 000004               MOV     #4,R2
1612 012410 022120              HX23:   CMP     (R1)+,(R0)+
1613 012412 001415                      BEQ     HX26
1614
1615 012414 012701 013360               MOV     #HXP7,R1
1616 012420 012700 013270               MOV     #HXDAT0,R0
1617 012424 012702 000004               MOV     #4,R2
1618 012430 022120              HX24:   CMP     (R1)+,(R0)+     ;WAS SIGN INCORRECT
1619 012432 001402                      BEQ     HX25
1620 012434 000137 013200               JMP     @#HXER11
1621 012440 077205              HX25:   SOB     R2,HX24
1622 012442 000137 013220               JMP     @#HXER12
1623
1624 012446 077220              HX26:   SOB     R2,HX23
1625
1626                            ;TEST   LDCFD   0
1627
1628 012450                     HX27:
     012450 104413                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1629 012452 012704 000200               MOV     #200,R4
1630 012456 170104                      LDFPS   R4
1631
1632 012460 012700 013300               MOV     #HXP1,R0
1633 012464 172410                      LDD     (R0),AC0
1634 012466 172010                      ADDD    (R0),AC0
1635
1636 012470 012737 012502 001236        MOV     #HX28,@#$TMP2
1637 012476 012700 013300               MOV     #HXP1,R0
1638 012502 177410              HX28:   LDCFD   (R0),AC0
1639
1640 012504 170205                      STFPS   R5
1641
1642 012506 012700 013270               MOV     #HXDAT0,R0
1643 012512 174010                      STD     AC0,(R0)        ;GET RESULT
1644
1645 012514 012701 013300               MOV     #HXP1,R1
1646 012520 012702 000004               MOV     #4,R2
1647 012524 022120              HX29:   CMP     (R1)+,(R0)+     ;IS IT 0?
1648 012526 001402                      BEQ     HX30
1649 012530 000137 013250               JMP     @#HXER13
1650 012534 077205              HX30:   SOB     R2,HX29
1651
1652 012536 012704 000204               MOV     #204,R4         ;FPS CORRECT
1653 012542 020405                      CMP     R4,R5
1654 012544 001402                      BEQ     HX31
1655 012546 000137 012776               JMP     @#HXER4
1656
```

```
1657                                        ;TEST   LDCFD   0
1658
1659 012552                                 HX31:
     012552  104413                                LFERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1660 012554  012704  000200                        MOV     #200,R4
1661 012560  170104                                LDFPS   R4
1662
1663 012562  012700  013350                        MOV     #HXP6,R0
1664 012566  172410                                LDD     (R0),AC0
1665
1666 012570  012737  012602  001236                MOV     #HX32,@#$TMP2
1667 012576  012700  013300                        MOV     #HXP1,R0
1668 012602  177410                         HX32:  LDCFD   (R0),AC0
1669
1670 012604  170205                                STFPS   R5
1671
1672 012606  012700  013270                        MOV     #HXDAT0,R0
1673 012612  174010                                STD     AC0,(R0)        ;GET RESULT
1674
1675 012614  012701  013300                        MOV     #HXP1,R1
1676 012620  012702  000004                        MOV     #4,R2
1677 012624  022120                         HX33:  CMP     (R1)+,(R0)+     ;IS IT ZERO?
1678 012626  001402                                BEQ     HX34
1679 012630  000137  013250                        JMP     @#HXER13
1680 012634  077205                         HX34:  SOB     R2,HX33
1681
1682 012636  012704  000204                        MOV     #204,R4         ;FPS CORRECT?
1683 012642  020405                                CMP     R4,R5
1684 012644  001402                                BEQ     HX35
1685 012646  000137  012776                        JMP     @#HXER4
1686 012652  000137  013400                 HX35:  JMP     @#HXDONE
1687
1688                                        ;R0 INCORRECT
1689
1690 012656  012737  013314  001242         HXER1: MOV     #HXP2+4,@#$TMP4
1691 012664  010037  001240                        MOV     R0,@#$TMP3
1692 012670  104234                         1$:    ERROR   +234
1693 012672  000137  013400                        JMP     @#HXDONE
1694
1695 012676  012737  013320  001242         HXER5: MOV     #HXP2+10,@#$TMP4
1696 012704  010037  001240                        MOV     R0,@#$TMP3
1697 012710  104237                         1$:    ERROR   +237
1698 012712  000137  013400                        JMP     @#HXDONE
1699                                        ;REPORT BAD DATA
1700 012716  012737  013310  001244         HXER2: MOV     #HXP2,@#$TMP5
1701 012724  012737  013360  001250                MOV     #HXP7,@#$TMP7
1702 012732  012737  013270  001246         HXER22:MOV     #HXDAT0,@#$TMP6
1703 012740  104233                         1$:    ERROR   +233
1704 012742  000137  013400                        JMP     @#HXDONE
1705                                        ;
1706 012746  012737  013310  001244         HXER3: MOV     #HXP2,@#$TMP5
1707 012754  012737  013360  001250                MOV     #HXP7,@#$TMP7
1708 012762  012737  013270  001246         HXER33:MOV     #HXDAT0,@#$TMP6
1709 012770  104241                         1$:    ERROR   +241
1710 012772  000137  013400                        JMP     @#HXDONE
1711
1712 012776  010537  001240                 HXER4: MOV     R5,@#$TMP3
```

```
1713 013002  010437  001242            MOV     R4,@#$TMP4
1714 013006  104240              1$:   ERROR   +240
1715 013010  000137  013400            JMP     @#HXDONE
1716
1717 013014  010537  001240      HXER8:  MOV     R5,@#$TMP3
1718 013020  010437  001242              MOV     R4,@#$TMP4
1719 013024  104242              1$:   ERROR   +242
1720 013026  000137  013400            JMP     @#HXDONE
1721 013032  012737  013310  001244  HXER6:  MOV     #HXP2,@#$TMP5
1722 013040  012737  013370  001250          MOV     #HXP8,@#$TMP7
1723 013046  012737  013270  001246  HXER66: MOV     #HXDAT0,@#$TMP6
1724 013054  104244              1$:   ERROR   +244
1725 013056  000137  013400            JMP     @#HXDONE
1726
1727 013062  012737  013310  001244  HXER7:  MOV     #HXP2,@#$TMP5
1728 013070  012737  013370  001250          MOV     #HXP8,@#$TMP7
1729 013076  012737  013270  001246          MOV     #HXDAT0,@#$TMP6
1730 013104  104243              1$:   ERROR   +243
1731 013106  000137  013400            JMP     @#HXDONE
1732
1733 013112  032716  000001      HXER9:  BIT     #1,(SP)         ;SEE IF IT
1734 013116  001005                      BNE     1$              ;AN ODD ADDRESS
1735 013120  022716  012222              CMP     #HX165,(SP)
1736 013124  001402                      BEQ     1$
1737 013126  000137  036712              JMP     @#CPSPUR
1738
1739 013132  011637  001236        1$:   MOV     (SP),@#$TMP2
1740 013136  022626                      CMP     (SP)+,(SP)+
1741 013140  104235              2$:   ERROR   +235
1742 013142  000137  013400            JMP     @#HXDONE
1743
1744 013146  162701  000003      HXER10: SUB     #3,R1
1745 013152  006301                      ASL     R1
1746 013154  012702  012222              MOV     #HX165,R2
1747 013160  010237  001242              MOV     R2,@#$TMP4
1748 013164  160102                      SUB     R1,R2
1749 013166  010237  001240              MOV     R2,@#$TMP3
1750 013172  104236              1$:   ERROR   +236
1751 013174  000137  013400            JMP     @#HXDONE
1752
1753 013200  012737  013330  001244  HXER11: MOV     #HXP4,@#$TMP5
1754 013206  012737  013340  001250          MOV     #HXP5,@#$TMP7
1755 013214  000137  012732              JMP     @#HXER22
1756 013220  012737  013330  001244  HXER12: MOV     #HXP4,@#$TMP5
1757 013226  012737  013340  001250          MOV     #HXP5,@#$TMP7
1758 013234  012737  013270  001246          MOV     #HXDAT0,@#$TMP6
1759 013242  104245              1$:   ERROR   +245
1760 013244  000137  013400            JMP     @#HXDONE
1761
1762 013250  012737  013300  001244  HXER13: MOV     #HXP1,@#$TMP5
1763 013256  012737  013300  001250          MOV     #HXP1,@#$TMP7
1764 013264  000137  012732              JMP     @#HXER22
1765
1766 013270  000000            HXDAT0: 0
1767 013272  000000                    0
1768 013274  000000                    0
1769 013276  000000                    0
```

```
1770
1771 013300  000000                    HXP1:   0
1772 013302  000000                            0
1773 013304  000000                            0
1774 013306  000000                            0
1775
1776 013310  000577                    HXP2:   577
1777 013312  177776                            177776
1778 013314  177777                            177777
1779 013316  177776                            177776
1780 013320  005201                    HXP3:   5201
1781 013322  000000                            0
1782 013324  000000                            0
1783 013326  000000                            0
1784 013330  100577                    HXP4:   100577
1785 013332  177776                            177776
1786 013334  177777                            177777
1787 013336  177776                            177776
1788 013340  100577                    HXP5:   100577
1789 013342  177776                            177776
1790 013344  000000                            0
1791 013346  000000                            0
1792 013350  000252                    HXP6:   252
1793 013352  125252                            125252
1794 013354  125252                            125252
1795 013356  125252                            125252
1796
1797 013360  000577                    HXP7:   577
1798 013362  177776                            177776
1799 013364  000000                            0
1800 013366  000000                            0
1801 013370  000577                    HXP8:   577
1802 013372  177777                            177777
1803 013374  000000                            0
1804 013376  000000                            0
1805
1806 013400                            HXDONE:
     013400  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?).
1807
1808
1809
1817
1818                                    ;;**********************************************************************
                                        ;*TEST 4         CMPD TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE CMPD INSTRUCTION. NOTE THAT A SUBROUTINE
                                        ;*IS USED TO SET UP OPERANDS, EXECUTE THE INSTRUCTION AND CHECK THE
                                        ;*RESULTS
                                        ;*
                                        ;;**********************************************************************
     013402  000004                     TST4:   SCOPE
1819
1820                                    ;TEST THE CMPD INSTRUCTION WITH (FSRC=AC=0)
```

```
1821 013404                          AAA1:
     013404  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1822 013406  004737  014216                JSR     PC,@#CMPSUB
1823 013412  000000  000000  000000  1$:   .WORD   0,0,0,0                 ;AC0
     013420  000000
1824 013422  000000  000000  000000  2$:   .WORD   0,0,0,0                 ;FSRC
     013430  000000
1825 013432  000200                  3$:   200                             ;FPS BEFORE EXECUTION
1826 013434  000204                        204                             ;FPS AFTER EXECUTION
1827 013436  000200                        200                             ;ERROR FPS
1828 013440  104001                  4$:   ERROR   +1                      ;FPS ERROR
1829
1830
1831                                 ;TEST CMPD WITH (AC=0) AND FSRC POSITIVE.
1832 013442                          AAA2:
     013442  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1833 013444  004737  014216                JSR     PC,@#CMPSUB
1834 013450  000000  000000  000000  1$:   .WORD   0,0,0,0                 ;AC
     013456  000000
1835 013460  025252                  2$:   25252                           ;FSRC
1836 013462  052525                        52525
1837 013464  125252                        125252
1838 013466  052525                        52525
1839 013470  000200                  3$:   200                             ;FPS BEFORE EXECUTION
1840 013472  000200                        200                             ;FPS AFTER EXECUTION
1841 013474  000210                        210                             ;ERROR FPS
1842 013476  104003                  4$:   ERROR   +3                      ;FPS ERROR
1843
1844                                 ;TEST CMPD WITH (AC=0) AND FSRC NEGATIVE
1845 013500                          AAA3:
     013500  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1846 013502  004737  014216                JSR     PC,@#CMPSUB
1847 013506  000000  000000  000000  1$:   .WORD   0,0,0,0                 ;AC
     013514  000000
1848 013516  125252                  2$:   125252                          ;FSRC
1849 013520  125252                        125252
1850 013522  052525                        52525
1851 013524  125252                        125252
1852 013526  000200                  3$:   200                             ;FPS BEFORE EXECUTION
1853 013530  000210                        210                             ;FPS AFTER EXECUTION
1854 013532  000200                        200                             ;ERROR FPS
1855 013534  104004                  4$:   ERROR   +4                      ;FPS ERROR.
1856
1857                                 ;TEST CMPD WITH (FSRC=0) AND AC POSITIVE
1858 013536                          AAA4:
     013536  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1859 013540  004737  014216                JSR     PC,@#CMPSUB
1860 013544  025252                  1$:   25252                           ;AC
1861 013546  052525                        52525
1862 013550  125252                        125252
1863 013552  052525                        52525
1864 013554  000000  000000  000000  2$:   .WORD   0,0,0,0                 ;FSRC
     013562  000000
1865 013564  000200                  3$:   200                             ;FPS BEFORE EXECUTION
1866 013566  000210                        210                             ;FPS AFTER EXECUTION
1867 013570  000200                        200                             ;ERROR FPS
1868 013572  104005                  4$:   ERROR   +5                      ;FPS ERROR
```

```
1869
1870
1871                                    ;TEST CMPD WITH (FSRC=0) AND AC NEGATIVE
1872 013574                            AAA5:
     013574  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1873 013576  004737  014216                    JSR     PC,@#CMPSUB
1874 013602  125252                    1$:     125252                  ;AC
1875 013604  125252                            125252
1876 013606  052525                            52525
1877 013610  125252                            125252
1878 013612  000000  000000  000000  2$:      .WORD   0,0,0,0         ;FSRC
     013620  000000
1879 013622  000200                    3$:     200                     ;FPS BEFORE EXECUTION
1880 013624  000200                            200                     ;FPS AFTER EXECUTION
1881 013626  000210                            210                     ;ERROR FPS
1882 013630  104006                    4$:     ERROR   +6              ;FPS ERROR
1883
1884                                    ;TEST CMPD WITH AC POSITIVE AND FSRC NEGATIVE
1885 013632                            AAA6:
     013632  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1886 013634  004737  014216                    JSR     PC,@#CMPSUB
1887 013640  052525                    1$:     52525                   ;AC
1888 013642  125252                            125252
1889 013644  052525                            52525
1890 013646  125252                            125252
1891 013650  125252                    2$:     125252                  ;;FSRC
1892 013652  052525                            52525
1893 013654  125252                            125252
1894 013656  052525                            52525
1895 013660  000200                    3$:     200                     ;FPS BEFORE EXECUTION
1896 013662  000210                            210                     ;FPS AFTER EXECUTION
1897 013664  000200                            200                     ;ERROR FPS
1898 013666  104007                    4$:     ERROR   +7              ;FPS ERROR
1899
1900
1901                                    ;TEST CMPD WITH AC NEGATIVE AND FSRC POSITIVE
1902 013670                            AAA7:
     013670  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1903 013672  004737  014216                    JSR     PC,@#CMPSUB
1904 013676  125252                    1$:     125252                  ;AC
1905 013700  052525                            52525
1906 013702  125252                            125252
1907 013704  052525                            52525
1908 013706  052525                    2$:     52525                   ;FSRC
1909 013710  125252                            125252
1910 013712  052525                            52525
1911 013714  125252                            125252
1912 013716  000200                    3$:     200                     ;FPS BEFORE EXECUTION
1913 013720  000200                            200                     ;FPS AFTER EXECUTION
1914 013722  000210                            210                     ;ERROR FPS
1915 013724  104010                    4$:     ERROR   +10             ;FPS ERROR.
1916
1917                                    ;TEST CMPD WITH AC POSITIVE AND FSRC POSITIVE
1918                                    ;AND EAC LESS THAN EFSRC.
1919 013726                            AAA8:
     013726  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1920 013730  004737  014216                    JSR     PC,@#CMPSUB
```

```
 1921 013734  012345                1$:     12345                           ;AC
 1922 013736  067654                        67654
 1923 013740  032101                        32101
 1924 013742  023456                        23456
 1925 013744  023456                2$:     23456                           ;FSRC
 1926 013746  076543                        76543
 1927 013750  021012                        21012
 1928 013752  034567                        34567
 1929 013754  000200                3$:     200                             ;FPS BEFORE EXECUTION
 1930 013756  000200                        200                             ;FPS AFTER EXECUTION
 1931 013760  000210                        210                             ;ERROR FPS
 1932 013762  104011                4$:     ERROR   +11                     ;FPS ERROR
 1933
 1934
 1935                               ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE AND EAC GREATER THAN EFSRC
 1936 013764                        AAA9:
      013764  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 1937 013766  004737  014216                JSR     PC,@#CMPSUB
 1938 013772  045676                1$:     45676                           ;AC
 1939 013774  054321                        54321
 1940 013776  012345                        12345
 1941 014000  067654                        67654
 1942 014002  034567                2$:     34567                           ;FSRC
 1943 014004  065432                        65432
 1944 014006  101234                        101234
 1945 014010  056765                        56765
 1946 014012  000200                3$:     200                             ;FPS BEFORE EXECUTION
 1947 014014  000210                        210                             ;FPS AFTER EXECUTION
 1948 014016  000200                        200                             ;ERROR FPS
 1949 014020  104012                4$:     ERROR   +12
 1950
 1951                               ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE AND AC EQUAL TO FSRC
 1952 014022                        AAA10:
      014022  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 1953 014024  004737  014216                JSR     PC,@#CMPSUB
 1954 014030  012345                1$:     12345                           ;AC
 1955 014032  067012                        67012
 1956 014034  034567                        34567
 1957 014036  012345                        012345
 1958 014040  012345                2$:     12345                           ;FSRC
 1959 014042  067012                        67012
 1960 014044  034567                        34567
 1961 014046  012345                        012345
 1962 014050  000200                3$:     200                             ;FPS BEFORE EXECUTION
```

```
 1964 014052  000204                        204                             ;FPS AFTER EXECUTION
 1965 014054  000200                        200                             ;ERROR FPS
 1966 014056  104013              4$:       ERROR   +13                     ;FPS ERROR
 1967
 1968                             ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE, EAC EQUAL TO EFSRC,
 1969                             ;AND FSRC GREATER THAN AC.
 1970 014060                      AAA11:
      014060  104413                       LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 1971 014062  004737  014216               JSR     PC,@#CMPSUB
 1972 014066  012345              1$:       12345                           ;AC
 1973 014070  067012                        67012
 1974 014072  034567                        34567
 1975 014074  012345                        012345
 1976 014076  012345              2$:       12345                           ;FSRC
 1977 014100  070123                        70123
 1978 014102  045670                        45670
 1979 014104  123456                        123456
 1980 014106  000200              3$:       200                             ;FPS BEFORE EXECUTION
 1981 014110  000200                        200                             ;FPS AFTER EXECUTION
 1982 014112  000210                        210                             ;ERROR FPS
 1983 014114  104014              4$:       ERROR   +14                     ;FPS ERROR
 1984
 1985                             ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE, EAC EQUAL TO EFSRC,
 1986                             ;AND AC GREATER THAN FSRC.
 1987 014116                      AAA12:
      014116  104413                       LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 1988 014120  004737  014216               JSR     PC,@#CMPSUB
 1989 014124  054321              1$:       54321                           ;AC
 1990 014126  076543                        76543
 1991 014130  021076                        21076
 1992 014132  054321                        54321
 1993 014134  054321              2$:       54321                           ;FSRC
 1994 014136  065432                        65432
 1995 014140  107654                        107654
 1996 014142  032107                        32107
 1997 014144  000200              3$:       200                             ;FPS BEFORE EXECUTION
 1998 014146  000210                        210                             ;FPS AFTER EXECUTION
 1999 014150  000200                        200                             ;ERROR FPS
 2000 014152  104015              4$:       ERROR   +15                     ;FPS ERROR
 2001
 2002                             ;TEST CMPD WITH AC NEGATIVE, FSRC NEGATIVE, EAC EQUAL TO EFSRC,
 2003                             ;AND AC GREATER THAN FSRC
 2004 014154                      AAA13:
      014154  104413                       LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 2005 014156  004737  014216               JSR     PC,@#CMPSUB
 2006 014162  112345              1$:       112345                          ;AC
 2007 014164  043210                        43210
 2008 014166  076543                        76543
 2009 014170  021076                        21076
 2010 014172  112345              2$:       112345                          ;FSRC
 2011 014174  054321                        54321
 2012 014176  007654                        07654
 2013 014200  032107                        32107
 2014 014202  000200              3$:       200                             ;FPS BEFORE EXECUTION
 2015 014204  000210                        210                             ;FPS AFTER EXECUTION
 2016 014206  000200                        200                             ;ERROR FPS
 2017 014210  104016              4$:       ERROR   +16                     ;FPS ERROR
```

```
                                ;FINISHED CMPD TEST.
                        ;THIS SUBROUTINE, CMPSUB, IS CALLED TO SET UP, EXECUTE
                        ;AND CHECK THE RESULTS OF A CMPD INSTRUCTION.
                        ;IT IS CALLED THUS:
                        ;
                        ;       JSR     PC,@#CMPSUB
                        ;ACARG: .WORD   X,X,X,X          ;AC OPERAND
                        ;FSRCARG:.WORD  X,X,X,X          ;FSRC OPERAND
                        ;FPSB:  .WORD   X                ;FPS BEFORE EXECUTION
                        ;FPSA:  .WORD   X                ;FPS AFTER EXECUTION
                        ;FPSE:  .WORD   X                ;ERROR FPS
                        ;ERR:   ERROR   +X               ;FPS ERROR
                        ;CONT:                           ;RETURN ADDRESS
                        ;
                        ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
                        ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, CMPD, IS EXECUTED.
                        ;AFTER THE EXECUTION THE FPS IS CHECKED AGAINST FPSA. IF IT IS A MATCH
                        ;THEN THERE WAS NO ERROR AND CONTROL IS RETURNED TO CONT. IF
                        ;THE FPS IS INCORRECT IT IS COMPARED WITH FPSE IN AN ATTEMPT TO ANALYSE
                        ;THE FAILURE. IF THE FPS IS THE SAME AS FPSE THEN CONTROL IS
                        ;RETURNED TO THE ERROR CALL AT LOCATION ERR. IF THE FPS WAS
                        ;NOT CORRECT BUT DIDN'T MATCH FPSE A GENERAL ERROR IS REPORTED
                        ;AND CONTROL IS PASSED TO CONT.

014216  012601          CMPSUB: MOV     (SP)+,R1        ;PICK UP A POINTER TO THE
                                                        ;ARGUMENTS.
014220  016100  000020          MOV     20(R1),R0       ;GET THE FPS BEFORE EXECUTION.
014224  170100                  LDFPS   R0              ;LOAD IT INTO THE FPS.

014226  012737  014250  001236  MOV     #1$,@#$TMP2     ;SAVE ADDRESS OF CMPD INSTRUCTION.
014234  010100                  MOV     R1,R0           ;GET ADDRESS OF AC OPERAND.
014236  172410                  LDD     (R0),AC0        ;LOAD AC0 OPERAND

014240  010100                  MOV     R1,R0           ;COMPUTE FSRC OPERAND
014242  062700  000010          ADD     #10,R0          ;ADDRESS

014246  000240                  NOP                     ;FOR SCOPING.
014250  173410          1$:     CMPD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.

014252  170205                  STFPS   R5              ;SAVE FPS AFTER INSTRUCTION.

014254  016104  000022          MOV     22(R1),R4       ;GET EXPECTED FPS.
                                                        ;IF INCORRECT SET UP FOR
014260  010137  001240          MOV     R1,@#$TMP3      ;AN ERROR CALL.
014264  010137  001242          MOV     R1,@#$TMP4
014270  062737  000010  001242  ADD     #10,@#$TMP4
014276  010537  001244          MOV     R5,@#$TMP5
014302  010437  001246          MOV     R4,@#$TMP6
014306  020405                  CMP     R4,R5           ;WAS FPS CORRECT?
014310  001410                  BEQ     3$              ;BRANCH IF YES.


014312  026105  000024          CMP     24(R1),R5       ;WAS THE FPS THE SAME
```

```
2075
2076 014316  001003                    BNE     2$                      ;AS THE EXPECTED INCORRECT FPS?
2077                                                                    ;BRANCH IF NO MATCH.
2078 014320  062701  000026             ADD     #26,R1                  ;IF THE EXPECTED INCORRECT
2079                                                                ;FPS MATCHED THE RESULTANT FPS
2080 014324  000111                    JMP     (R1)                    ;RETURN TO THE ERROR CALL
2081                                                                    ;IN THE CALLING ROUTINE.
2082
2083 014326  104001          2$:       ERROR   +1                      ;OTHERWISE REPORT INCORRECT FPS
2084 014330  000411                    BR      5$
2085
2086 014332  012700  014376    3$:     MOV     #CMPTMP,R0              ;IF FPS WAS CORRECT MAKE SURE
2087 014336  174010                    STD     AC0,(R0)                ;AC0 WAS NOT AFFECTED BY CMPD.
2088 014340  010102                    MOV     R1,R2
2089 014342  012703  000004            MOV     #4,R3
2090 014346  022220          4$:       CMP     (R2)+,(R0)+
2091 014350  001003                    BNE     6$
2092 014352  077303                    SOB     R3,4$
2093
2094 014354  000161  000030    5$:     JMP     30(R1)                  ;RETURN
2095
2096 014360                    6$:                                     ;REPORT AC0 MODIFIED BY CMPD
2097 014360  010137  001240            MOV     R1,@#$TMP3
2098 014364  012737  014376  001242    MOV     #CMPTMP,@#$TMP4
2099 014372  104002          7$:       ERROR   +2
2100 014374  000767                    BR      5$                      ;RETURN
2101
2102 014376  000000  000000  000000  CMPTMP: .WORD   0,0,0,0
     014404  000000
2103
2104
2105
2106 014406                          AAADONE:
     014406  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?).
2107
2108
2109
2117
2118                                    ;;**********************************************************************
                                        ;*TEST 5        DIVD WITH (FSRC=0) AND (BUT FD) TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE DIVD INSTRUCTION WITH A
                                        ;*ZERO DIVISOR. THE CONDITION IS CHECKED WITH BOTH
                                        ;*TRAP ENABLED AND TRAPS DISABLED.
                                        ;*
                                        ;;**********************************************************************
     014410  000004                     TST5:   SCOPE
2119
2120                                    ;FIRST TEST DIVD WITH (FSRC=AC=0) AND TRAPS DISABLED.
2121 014412  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2122 014414  012704  040200            BBB0:   MOV     #40200,R4       ;SET UP FPS
2123                                                                    ;WITH INTERRUPTS
2124                                                                    ;DISABLED.
```

```
2125 014420  170104                        LDFPS   R4
2126 014422  012737  014664  000244        MOV     #BBBER1,@#FPVECT;SET UP FOR ANY FP INTERRUPTS.
2127 014430  012737  014450  001236        MOV     #BBB1,@#$TMP2
2128 014436  012700  015070                MOV     #BBBP1,R0       ;SET UP AC0 = 0
2129 014442  172410                        LDD     (R0),AC0
2130 014444  012701  015070                MOV     #BBBP1,R1       ;FSRC = 0
2131
2132 014450  174411                BBB1:   DIVD    (R1),AC0        ;TEST INSTRUCTION
2133
2134 014452  170205                        STFPS   R5              ;GET FPS
2135 014454  170303                        STST    R3              ;GET FEC
2136
2137 014456  012704  140204                MOV     #140204,R4      ;EXPECTED FPS.
2138 014462  020405                        CMP     R4,R5           ;IS FPS CORRECT.
2139 014464  001131                        BNE     BBBER2          ;IF INCORRECT BRANCH.
2140
2141 014466  012702  000004                MOV     #4,R2           ;EXPECTED FEC.
2142 014472  020203                        CMP     R2,R3           ;IS FEC CORRECT?
2143 014474  001140                        BNE     BBBER3          ;IF INCORRECT BRANCH.
2144
2145                               ;TEST DIVD WITH (FSRC=0) AND TRAPS DISABLED.
2146 014476                        BBB2:
     014476  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2147 014500  012704  040200                MOV     #40200,R4       ;LOAD FPS WITH TRAPS DISABLED.
2148 014504  170104                        LDFPS   R4
2149
2150 014506  012737  014526  001236        MOV     #BBB3,@#$TMP2
2151 014514  012700  015100                MOV     #BBBP2,R0       ;SET UP AC0 OPERAND (NON ZERO).
2152 014520  172410                        LDD     (R0),AC0
2153 014522  012700  015070                MOV     #BBBP1,R0       ;FSRC=0
2154 014526  174410                BBB3:   DIVD    (R0),AC0
2155
2156 014530  170205                        STFPS   R5              ;GET FPS.
2157 014532  170303                        STST    R3              ;GET FEC.
2158
2159 014534  012704  140200                MOV     #140200,R4              ;EXPECTED FPS.
2160 014540  020405                        CMP     R4,R5           ;IS FPS CORRECT?
2161 014542  001102                        BNE     BBBER2          ;IF INCORRECT BRANCH.
2162
2163 014544  012702  000004                MOV     #4,R2           ;EXPECTED FEC.
2164 014550  020203                        CMP     R2,R3           ;WAS FEC CORRECT?
2165 014552  001111                        BNE     BBBER3          ;IF INCORRECT BRANCH.
2166
2167                               ;TEST DIVD WITH FSRC=0) AND TRAPS ENABLED.
2168 014554                        BBB4:
     014554  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2169 014556  012704  000200                MOV     #200,R4         ;SET UP FPS. TRAP ENABLED.
2170 014562  170104                        LDFPS   R4
2171
2172 014564  012737  014612  001236        MOV     #BBB5,@#$TMP2
2173 014572  012700  015100                MOV     #BBBP2,R0       ;SET UP AC0 OPERAND (NON ZERO).
2174 014576  172410                        LDD     (R0),AC0
2175
2176 014600  012737  014620  000244        MOV     #BBB6,@#FPVECT  ;SET UP FOR THE EXPECTED INTERRUPT.
2177 014606  012700  015070                MOV     #BBBP1,R0       ;FSRC=0
2178
2179 014612  174410                BBB5:   DIVD    (R0),AC0        ;TEST INSTRUCTION (SHOULD RESULT IN TRAP).
```

```
2180 014614  170000                        CFCC
2181
2182 014616  000502                        BR      BBBER4          ;GO REPORT FAILURE, NO TRAP.
2183
2184 014620  022716  014614          BBB6: CMP     #BBB5+2,(SP)    ;TRAP TO HERE WHEN THE DIVISION BY 0
2185                                                                ;OCCURS. FIRST SEE IF THE ADDRESS OF
2186                                                                ;THE TRAP IS 2+THE ADDRESS OF THE TEST
2187                                                                ;DIVD INSTRUCTION.
2188 014624  001402                        BEQ     1$
2189 014626  000137  036660                JMP     @#FPSPUR        ;IF NOT THEN REPORT AN UNEXPECTED
2190                                                                ;FP TRAP.
2191 014632  170205                  1$:   STFPS   R5              ;GET FPS.
2192 014634  170303                        STST    R3              ;GET FEC.
2193 014636  022626                        CMP     (SP)+,(SP)+     ;RESET THE STACK.
2194
2195 014640  012704  100200                MOV     #100200,R4      ;EXPECTED FPS.
2196 014644  020405                        CMP     R4,R5           ;IS FPS CORRECT?
2197 014646  001040                        BNE     BBBER2          ;IF INCORRECT BRANCH.
2198
2199 014650  012702  000004                MOV     #4,R2           ;EXPECTED FEC.
2200 014654  020203                        CMP     R2,R3           ;IS FEC CORRECT?
2201 014656  001047                        BNE     BBBER3          ;IF INCORRECT BRANCH.
2202
2203 014660  000137  015110                JMP     @#BBBDONE       ;OTHERWISE GO TO NEXT TEST.
2204
2205
2206                              ;TRAP HERE IF AN UNEXPECTED INTERRUPT OCCURS.
2207 014664  062737  000002  001236 BBBER1: ADD     #2,@#$TMP2      ;SEE IF THE INTERRUPT OCCURRED
2208                                                                ;DURING THE EXECUTION OF THE DIVD
2209                                                                ;INSTRUCTION BEING TESTED.
2210 014672  021637  001236                CMP     (SP),@#$TMP2
2211 014676  001402                        BEQ     1$
2212 014700  000137  036660                JMP     @#FPSPUR        ;IF NOT REPORT UNEXPECTED FP TRAP.
2213
2214 014704  022626                  1$:   CMP     (SP)+,(SP)+     ;RESET THE STACK.
2215 014706  170303                        STST    R3              ;GET FEC.
2216 014710  170205                        STFPS   R5              ;GET FPS.
2217 014712  012737  000004  001240        MOV     #4,@#$TMP3      ;EXPECTED FEC.
2218 014720  010337  001242                MOV     R3,@#$TMP4
2219 014724  010537  001244                MOV     R5,@#$TMP5
2220 014730  010037  001250                MOV     R0,@#$TMP7
2221 014734  012737  140200  001246        MOV     #140200,@#$TMP6
2222 014742  104017                  2$:   ERROR   +17             ;REPORT (BUT FD) FAILED RESULTING IN AN FP TRAP
2223                                                                ;WITH TRAPS DISABLED.
2224 014744  000137  015110                JMP     @#BBBDONE
2225
2226                              ;REPORT FPS INCORRECT:
2227 014750  010537  001242       BBBER2: MOV     R5,@#$TMP4
2228 014754  010437  001244               MOV     R4,@#$TMP5
2229 014760  010037  001246               MOV     R0,@#$TMP6
2230 014764  010137  001250               MOV     R1,@#$TMP7
2231 014770  104020                  1$:   ERROR   +20
2232 014772  000137  015110               JMP     @#BBBDONE
2233
2234                              ;REPORT FEC INCORRECT:
2235 014776  010337  001242       BBBER3: MOV     R3,@#$TMP4
2236 015002  010237  001240               MOV     R2,@#$TMP3
```

```
2237 015006  010037  001246                    MOV     R0,@#$TMP6
2238 015012  010137  001250                    MOV     R1,@#$TMP7
2239 015016  104021                    1$:     ERROR   +21
2240 015020  000137  015110                    JMP     @#BBBDONE
2241
2242                                    ;REPORT NO TRAP OCCURRED AFTER TRYING TO DIVIDE
2243                                    ;BY ZERO WITH ALL TRAPS ENABLED.
2244 015024  170303                    BBBER4: STST    R3              ;GET FEC.
2245 015026  170205                            STFPS   R5              ;GET FPS.
2246 015030  012737  000004  001242            MOV     #4,@#$TMP4
2247 015036  010337  001240                    MOV     R3,@#$TMP3
2248 015042  010537  001244                    MOV     R5,@#$TMP5
2249 015046  012737  100200  001246            MOV     #100200,@#$TMP6
2250 015054  010037  001250                    MOV     R0,@#$TMP7
2251 015060  010137  001252                    MOV     R1,@#$TMP10
2252 015064  104022                    1$:     ERROR   +22
2253 015066  000410                            BR      BBBDONE
2254
2255 015070  000000  000000  000000    BBBP1:  .WORD   0,0,0,0
     015076  000000
2256 015100  012345  054321  023456    BBBP2:  .WORD   12345,54321,23456,76543
     015106  076543
2257
2258
2259
2260 015110                             BBBDONE:
     015110  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?).
2261
2262
2270
2271                                    ;;**********************************************************************
                                        ;*TEST 6        DIVF TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE DIVF INSTRUCTION. NOTE THAT A SUBROUTINE IS
                                        ;*USED TO SET UP THE OPERANDS, EXECUTE THE INSTRUCTION AND CHECK THE
                                        ;*RESULTS.
                                        ;*
                                        ;;**********************************************************************
     015112  000004                     TST6:   SCOPE
2272
2273                                    ;CHECK DIVF WITH (AC=0).
2274 015114                             CCC1:
     015114  104413                             LPERR                  ;SET UP THE LOOP ON ERROR ADDRESS.
2275 015116  004767  000552                     JSR     PC,DIVFSUB
2276 015122  000000  000000            1$:     .WORD   0,0             ;AC
2277 015126  012345  067012            2$:     .WORD   12345,67012     ;FSRC
2278 015132  000000  000000            3$:     .WORD   0,0             ;RES
2279 015136  000000                    4$:     0                       ;FPS BEFORE EXECUTION.
2280 015140  000004                            4                       ;FPS AFTER EXECUTION
2281 015142  012345  067012            5$:     .WORD   12345,67012     ;ERROR RESULT
2282 015146  104023                    6$:     ERROR   +23             ;RESULT BAD.
2283
2284                                    ;TEST DIVF WITH AC POSITIVE, FSRC POSITIVE AND IN ROUND MODE.
```

```
                                        CCC2:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2286 015152  004737  015674                    JSR     PC,@#DIVFSUB
2287 015156  065652  125252             1$:     .WORD   65652,125252    ;AC
2288 015162  065600  000000             2$:     .WORD   65600,0         ;FSRC
2289 015166  040252  125252             3$:     .WORD   40252,125252    ;RES
2290 015172  003000                     4$:     3000                    ;FPS BEFORE EXECUTION.
2291 015174  003000                             3000                    ;FPS AFTER EXECUTION.
2292 015176  040052  125252             5$:     .WORD   40052,125252    ;ERROR RESULT.
2293 015202  104024                     6$:     ERROR   +24             ;DIV NORMALIZE FAILURE.
2294
2295                                    ;TEST DIVF WITH AC POSITIVE, FSRC POSITIVE.
2296 015204                             CCC3:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2297 015206  004767  000462                    JSR     PC,DIVFSUB
2298 015212  076400  000000             1$:     .WORD   76400,0         ;AC
2299 015216  076400  000000             2$:     .WORD   76400,0         ;FSRC
2300 015222  040200  000000             3$:     .WORD   40200,0         ;RES
2301 015226  001000                     4$:     1000                    ;FPS BEFORE EXECUTION.
2302 015230  001000                             1000                    ;FPS AFTER EXECUTION.
2303 015232  140200  000000             5$:     .WORD   140200,0        ;ERROR RES.
2304
2305 015236  104025                     6$:     ERROR   +25             ;SIGN BAD.
2306
2307                                    ;TEST DIVF WITH BOTH OPERANDS POSITIVE.
2308 015240                             CCC4:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2309 015242  004737  015674                    JSR     PC,@#DIVFSUB
2310 015246  056777  177777             1$:     .WORD   56777,177777    ;AC
2311 015252  054200  000000             2$:     .WORD   54200,0         ;FSRC
2312 015256  042777  177777             3$:     .WORD   42777,177777    ;RES
2313 015262  000000                     4$:     0                       ;FPS BEFORE EXECUTION.
2314 015264  000000                             0                       ;FPS AFTER EXECUTION.
2315 015266  002000  002000             5$:     .WORD   2000,2000       ;ERROR RES.
2316 015272  104023                     6$:     ERROR   +23
2317
2318                                    ;TEST THE DIVF INSTRUCTION:
2319 015274                             CCC5:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2320 015276  004737  015674                    JSR     PC,@#DIVFSUB
2321 015302  012377  177777             1$:     .WORD   12377,177777    ;AC
2322 015306  012300  000000             2$:     .WORD   12300,0         ;FSRC
2323 015312  040252  125252             3$:     .WORD   40252,125252    ;RES
2324 015316  000000                     4$:     0                       ;FPS BEFORE EXECUTION.
2325 015320  000000                             0                       ;FPS AFTER EXECUTION.
2326 015322  177777  177777             5$:     .WORD   -1,-1           ;ERROR RES.
2327 015326  104023                     6$:     ERROR   +23
2328
2329                                    ;TEST DIVIDE ALGORITHM. TEST ROUND CONSTANT.
2330 015330                             CCC6:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2331 015332  004737  015674                    JSR     PC,@#DIVFSUB
2332 015336  064600  000001             1$:     .WORD   64600,1         ;AC
2333 015342  066600  000000             2$:     .WORD   66600,0         ;FSRC
2334 015346  036200  000001             3$:     .WORD   36200,1         ;RES
2335 015352  000000                     4$:     0                       ;FPS BEFORE EXECUTION.
2336 015354  000000                             0                       ;FPS AFTER EXECUTION.
```

```
2337 015356  003000  003000          5$:     .WORD   3000,3000        ;ERROR RES.
2338 015362  104023                  6$:     ERROR   +23
2339
2340                                  ;TEST DIVF.
2341 015364                           CCC7:
     015364  104413                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2342 015366  004737  015674                   JSR     PC,@#DIVFSUB
2343 015372  034577  177776          1$:     .WORD   34577,177776    ;AC
2344 015376  023400  000000          2$:     .WORD   23400,0         ;FSRC
2345 015402  051377  177776          3$:     .WORD   51377,177776    ;RES
2346 015406  000017                  4$:     17                      ;FPS BEFORE EXECUTION.
2347 015410  000000                          0                       ;FPS AFTER EXECUTION.
2348 015412  003400  003400          5$:     .WORD   3400,3400       ;ERROR RES.
2349 015416  104023                  6$:     ERROR   +23
2350
2351
2352                                  ;DIVF TEST.
2353 015420                           CCC8:
     015420  104413                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2354 015422  004737  015674                   JSR     PC,@#DIVFSUB
2355 015426  067652  125252          1$:     .WORD   67652,125252    ;AC
2356 015432  056500  000000          2$:     .WORD   56500,0         ;FSRC
2357 015436  051343  107070          3$:     .WORD   51343,107070    ;RES
2358 015442  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
2359 015444  000000                          0                       ;FPS AFTER EXECUTION.
2360 015446  051543  107070          5$:     .WORD   51543,107070    ;ERROR RES.
2361 015452  104026                  6$:     ERROR   +26             ;DIDN'T INCREMENT THE EXPONENT
2362                                                                  ;AFTER DIVID NORMALIZATION.
2363
2364                                  ;DIVF WITH AC NEGATIVE, FSRC NEGATIVE.
2365 015454                           CCC9:
     015454  104413                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2366 015456  004737  015674                   JSR     PC,@#DIVFSUB
2367 015462  140400  000000          1$:     .WORD   140400,0        ;AC
2368 015466  140500  000000          2$:     .WORD   140500,0        ;FSRC
2369 015472  040052  125253          3$:     .WORD   040052,125253   ;RES
2370 015476  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
2371 015500  000000                          0                       ;FPS AFTER EXECUTION.
2372 015502  140052  125253          5$:     .WORD   140052,125253   ;ERROR RES.
2373 015506  104027                  6$:     ERROR   +27             ;BAD SIGN.
2374
2375                                  ;DIVF WITH AC NEGATIVE AND FSRC POSITIVE.
2376 015510                           CCC10:
     015510  104413                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2377 015512  004737  015674                   JSR     PC,@#DIVFSUB
2378 015516  160077  000000          1$:     .WORD   160077,0        ;AC
2379 015522  040277  000000          2$:     .WORD   40277,0         ;FSRC
2380 015526  160000  000000          3$:     .WORD   160000,0        ;RES
2381 015532  000007                  4$:     7                       ;FPS BEFORE EXECUTION.
2382 015534  000010                          10                      ;FPS AFTER EXECUTION.
2383 015536  060000  000000          5$:     .WORD   60000,0         ;ERROR RES.
2384 015542  104027                  6$:     ERROR   +27             ;BAD SIGN.
2385
2386                                  ;DIVF WITH AC POSITIVE AND FSRC NEGATIVE.
2387 015544                           CCC11:
     015544  104413                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2388 015546  004737  015674                   JSR     PC,@#DIVFSUB
```

```
2389 015552  040400  000000          1$:     .WORD   40400,0          ;AC
2390 015556  140500  000000          2$:     .WORD   140500,0         ;FSRC
2391 015562  140052  125253          3$:     .WORD   140052,125253    ;RES
2392 015566  000017                  4$:     17                       ;FPS BEFORE EXECUTION.
2393 015570  000010                          10                       ;FPS AFTER EXECUTION.
2394 015572  040052  125253          5$:     .WORD   40052,125253     ;ERROR RES.
2395 015576  104027                  6$:     ERROR   +27              ;BAD SIGN.
2396
2397
2398                                  ;TEST DIVF BOTH OPERANDS POSITIVE AND TRUNCATE MODE.
2399 015600                          CCC12:
     015600  104413                          LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
2400 015602  004737  015674                  JSR     PC,@#DIVFSUB
2401 015606  060100  000001          1$:     .WORD   60100,1          ;AC
2402 015612  040300  000000          2$:     .WORD   40300,0          ;FSRC
2403 015616  060000  000000          3$:     .WORD   60000,0          ;RES
2404 015622  000052                  4$:     52                       ;FPS BEFORE EXECUTION.
2405 015624  000040                          40                       ;FPS AFTER EXECUTION.
2406 015626  060000  000001          5$:     .WORD   60000,1          ;ERROR RES.
2407 015632  104030                  6$:     ERROR   +30              ;TRUNCATION ERROR
2408
2409                                  ;DIVF WITH POSITIVE OPERANDS AND ROUND MODE.
2410 015634                          CCC13:
     015634  104413                          LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
2411 015636  004767  000032                  JSR     PC,DIVFSUB
2412 015642  060100  000001          1$:     .WORD   60100,1          ;AC
2413 015646  040300  000000          2$:     .WORD   40300,0          ;FSRC
2414 015652  060000  000001          3$:     .WORD   60000,1          ;RES
2415 015656  000005                  4$:     5                        ;FPS BEFORE EXECUTION.
2416 015660  000000                          0                        ;FPS AFTER EXECUTION.
2417 015662  060000  000000          5$:     .WORD   60000,0          ;ERROR RES.
2418 015666  104031                  6$:     ERROR   +31              ;ROUND ERROR.
2419
2420 015670  000137  016120                  JMP     @#CCCDONE        ;GO TO NEXT TEST.
2421
2422                                  ;THIS SUBROUTINE, DIVFSUB, IS CALLED TO SET UP, EXECUTE
2423                                  ;AND CHECK THE RESULT OF A DIVF INSTRUCTION. IT IS CALLED THUS:
2424                                  ;
2425                                  ;               JSR     PC,@#DIVFSUB
2426                                  ;       ACARG:  .WORD   X,X              ;AC OPERAND
2427                                  ;       FSRCARG:.WORD   X,X              ;FSRC OPERAND
2428                                  ;       RES:    .WORD   X,X              ;EXPECTED RESULT
2429                                  ;       FPSB:   .WORD   X                ;FPS BEFORE EXECUTION
2430                                  ;       FPSA:   .WORD   X                ;FPS AFTER EXECUTION
2431                                  ;       ERRES:  .WORD   X,X              ;ERROR RESULT
2432                                  ;       ERR:    ERROR   +X               ;RESULT ERROR
2433                                  ;       CONT:                            ;RETURN ADDRESS
2434                                  ;
2435                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
2436                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, DIVF IS EXECUTED.
2437                                  ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
2438                                  ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
2439                                  ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
2440                                  ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
2441                                  ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
2442                                  ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
2443                                  ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
```

```
                                        ;THEN THE FAILURE IS REPORTED IN DIVFSUB AND CONTROL IS PASSED TO
                                        ;CONT. IF NO ERRORS ARE DETECTED THEN DIVFSUB RETURNS CONTROL
                                        ;TO CONT.

015674  012601                  DIVFSUB:    MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
015676  012700  000200                      MOV     #200,R0         ;SET FD MODE.
015702  170100                              LDFPS   R0
015704  010100                              MOV     R1,R0           ;LOAD THE AC OPERAND.
015706  172410                              LDD     (R0),AC0
015710  016100  000014                      MOV     14(R1),R0       ;LOAD THE FPS
015714  170100                              LDFPS   R0
015716  012737  015732  001236              MOV     #1$,@#$TMP2
015724  010100                              MOV     R1,R0
015726  062700  000004                      ADD     #4,R0           ;ESTABLISH A POINTER TO FSRC.

015732  174410                  1$:         DIVF    (R0),AC0        ;TEST INSTRUCTION.

015734  170204                              STFPS   R4              ;GET THE FPS.
015736  012700  000200                      MOV     #200,R0         ;SET FD MODE
015742  170100                              LDFPS   R0

015744  012700  016110                      MOV     #DIVFT,R0       ;GET THE RESULT OF THE DIVF.
015750  174010                              STD     AC0,(R0)

015752  010102                              MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
015754  010237  001240                      MOV     R2,@#$TMP3
015760  062702  000004                      ADD     #4,R2
015764  010237  001242                      MOV     R2,@#$TMP4
015770  062702  000004                      ADD     #4,R2
015774  010237  001244                      MOV     R2,@#$TMP5
016000  012737  016110  001246              MOV     #DIVFT,@#$TMP6
016006  010437  001250                      MOV     R4,@#$TMP7
016012  016137  000016  001252              MOV     16(R1),@#$TMP10

016020  021061  000010                      CMP     (R0),10(R1)     ;IS THE RESULT CORRECT?
016024  001011                              BNE     10$             ;IF INCORRECT BRANCH.
016026  026061  000002  000012              CMP     2(R0),12(R1)
016034  001005                              BNE     10$

016036  026104  000016                      CMP     16(R1),R4       ;IS FPS CORRECT?
016042  001020                              BNE     15$             ;IF INCORRECT BRANCH.
016044  000161  000026                      JMP     26(R1)          ;IF NO ERRORS OCCURRED RETURN.

016050  021061  000020          10$:        CMP     (R0),20(R1)     ;DOES THE INCORRECT RESULT
016054  001010                              BNE     11$             ;MATCH THE ANTICIPATED INCORRECT RESULT.
016056  026061  000002  000022              CMP     2(R0),22(R1)
016064  001004                              BNE     11$             ;BRANCH IF NO.

016066  010102                              MOV     R1,R2           ;IT MATCHED SO RETURN TO THE ERROR
                                                                    ;REPORT AT THE CALLING ROUTINE.
016070  062702  000024                      ADD     #24,R2
016074  000112                              JMP     (R2)

016076                          11$:                                ;REPORT RESULT INCORRECT.
016076  104023                  12$:        ERROR   +23
016100  000161  000026          13$:        JMP     26(R1)
```

```
2501 016104                          15$:                          ;REPORT FPS INCORRECT.
2502 016104  104032                  16$:    ERROR   +32
2503 016106  000774                          BR      13$
2504
2505 016110  000000  000000  000000  DIVFT:  .WORD   0,0,0,0
     016116  000000
2506
2507 016120                          CCCDONE:
     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?).
2508
2509
2516
2517                                 ;;*****************************************************************
                                     ;*TEST 7        DIVD TEST
                                     ;*
                                     ;*THIS IS A TEST OF THE DIVD INSTRUCTION. NOTE THAT A SUBROUTINE IS
                                     ;*USED TO SET UP THE OPERANDS, EXECUTE THE INSTRUCTION AND CHECK THE RESULTS.
                                     ;*
                                     ;;*****************************************************************
     016122  000004                  TST7:   SCOPE
2518
2519                                 ;DIVD TEST WITH POSITIVE OPERANDS AND IN ROUND MODE.
2520 016124                          DDD1:
     016124  104413                          LPERR                 ;SET UP THE LOOP ON ERROR ADDRESS.
2521 016126  004737  016614                  JSR     PC,@#DIVDSUB
2522 016132  034277  000000  000000  1$:     .WORD   34277,0,0,0   ;AC
     016140  000000
2523 016142  040277  000000  000000  2$:     .WORD   40277,0,0,0   ;FSRC
     016150  000000
2524 016152  034200  000000  000000  3$:     .WORD   34200,0,0,0   ;RES
     016160  000000
2525 016162  000200                  4$:     200                   ;FPS BEFORE EXECUTION.
2526 016164  000200                          200                   ;FPS AFTER EXECUTION.
2527 016166  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1   ;ERROR RES.
     016174  177777
2528 016176  104033                  6$:     ERROR   +33
2529
2530                                 ;DIVD WITH AC NEGATIVE AND FSRC POSITIVE IN TRUNCATE MODE.
2531 016200                          DDD2:
     016200  104413                          LPERR                 ;SET UP THE LOOP ON ERROR ADDRESS.
2532 016202  004737  016614                  JSR     PC,@#DIVDSUB
2533 016206  134277  000000  000000  1$:     .WORD   134277,0,0,0  ;AC
     016214  000000
2534 016216  040277  000000  000000  2$:     .WORD   40277,0,0,0   ;FSRC
     016224  000000
2535 016226  134200  000000  000000  3$:     .WORD   134200,0,0,0  ;RES
     016234  000000
2536 016236  000207                  4$:     207                   ;FPS BEFORE EXECUTION.
2537 016240  000210                          210                   ;FPS AFTER EXECUTION.
2538 016242  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1   ;ERROR RESULT.
     016250  177777
2539 016252  104033                  6$:     ERROR   +33
2540
```

```
2541                                        ;DIVD TEST WITH OPERANDS BOTH NEGATIVE AND IN TRUNCATE MODE.
2542 016254                                 DDD3:
     016254  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2543 016256  004767  000332                        JSR     PC,DIVDSUB
2544 016262  134300  000000  000000  1$:          .WORD   134300,0,0,1    ;AC
     016270  000001
2545 016272  140300  000000  000000  2$:          .WORD   140300,0,0,0    ;FSRC
     016300  000000
2546 016302  034200  000000  000000  3$:          .WORD   34200,0,0,0     ;RES
     016310  000000
2547 016312  000250                   4$:          250                     ;FPS BEFORE EXECUTION.
2548 016314  000240                                240                     ;FPS AFTER EXECUTION.
2549 016316  034200  000000  000000  5$:          .WORD   34200,0,0,1     ;ERROR RES.
     016324  000001
2550 016326  104035                   6$:          ERROR   +35             ;TRUNCATION ERROR.
2551
2552                                        ;DIVD WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
2553 016330                                 DDD4:
     016330  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2554 016332  004737  016614                        JSR     PC,@#DIVDSUB
2555 016336  034300  000000  000000  1$:          .WORD   34300,0,0,1     ;AC
     016344  000001
2556 016346  140300  000000  000000  2$:          .WORD   140300,0,0,0    ;FSRC
     016354  000000
2557 016356  134200  000000  000000  3$:          .WORD   134200,0,0,1    ;RES
     016364  000001
2558 016366  000207                   4$:          207                     ;FPS BEFORE EXECUTION.
2559 016370  000210                                210                     ;FPS AFTER EXECUTION.
2560 016372  134200  000000  000000  5$:          .WORD   134200,0,0,0    ;ERROR RES.
     016400  000000
2561 016402  104036                   6$:          ERROR   +36             ;ROUND ERROR.
2562
2563                                        ;DIVD TEST.
2564 016404                                 DDD5:
     016404  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2565 016406  004737  016614                        JSR     PC,@#DIVDSUB
2566 016412  100400  000000  000000  1$:          .WORD   100400,0,0,0    ;AC
     016420  000000
2567 016422  000500  000000  000000  2$:          .WORD   500,0,0,0       ;FSRC
     016430  000000
2568 016432  140052  125252              3$:          .WORD   140052,125252   ;RES
2569 016436  125252  125252                           .WORD   125252,125252
2570 016442  007647                   4$:          7647                    ;FPS BEFORE EXECUTION.
2571 016444  007650                                7650                    ;FPS AFTER EXECUTION.
2572 016446  177777  177777  177777  5$:          .WORD   -1,-1,-1,-1     ;ERROR RES.
     016454  177777
2573 016456  104033                   6$:          ERROR   +33
2574
2575
2576                                        ;DIVD TEST WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
2577 016460                                 DDD6:
     016460  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2578 016462  004737  016614                        JSR     PC,@#DIVDSUB
2579 016466  000400  000000  000000  1$:          .WORD   400,0,0,0       ;AC
     016474  000000
2580 016476  100500  000000  000000  2$:          .WORD   100500,0,0,0    ;FSRC
     016504  000000
```

```
2581 016506 140052 125252          3$:   .WORD  140052,125252  ;RES
2582 016512 125252 125253                .WORD  125252,125253
2583 016516 007707                 4$:   7707                  ;FPS BEFORE EXECUTION.
2584 016520 007710                       7710                  ;FPS AFTER EXECUTION.
2585 016522 177777 177777 177777  5$:   .WORD  -1,-1,-1,-1     ;ERROR RES.
     016530 177777
2586 016532 104033                 6$:   ERROR  +33
2587
2588                               ;DIVD TEST.
2589 016534                        DDD7:
     016534 104413                       LPERR                 ;SET UP THE LOOP ON ERROR ADDRESS.
2590 016536 004737 016614                JSR    PC,@#DIVDSUB
2591 016542 170360 170360          1$:   .WORD  170360,170360  ;AC
2592 016546 170360 170360                .WORD  170360,170360
2593 016552 170360 170360          2$:   .WORD  170360,170360  ;FSRC
2594 016556 170360 170360                .WORD  170360,170360
2595 016562 040200 000000 000000   3$:   .WORD  40200,0,0,0     ;RES
     016570 000000
2596 016572 007717                 4$:   7717                  ;FPS BEFORE EXECUTION.
2597 016574 007700                       7700                  ;FPS AFTER EXECUTION.
2598 016576 177777 177777 177777  5$:   .WORD  -1,-1,-1,-1     ;ERROR RES.
     016604 177777
2599 016606 104033                 6$:   ERROR  +33
2600
2601 016610 000137 017054                JMP    @#DDDDONE      ;GO TO NEXT TEST.
2602
2603
2604                               ;THIS SUBROUTINE, DIVDSUB, IS CALLED TO SET UP, EXECUTE
2605                               ;AND CHECK THE RESULT OF A DIVD INSTRUCTION. IT IS CALLED THUS:
2606                               ;
2607                               ;        JSR     PC,@#DIVDSUB
2608                               ;ACARG:  .WORD   X,X,X,X          ;AC OPERAND
2609                               ;FSRCARG:.WORD   X,X,X,X          ;FSRC OPERAND
2610                               ;RES:    .WORD   X,X,X,X          ;EXPECTED RESULT
2611                               ;FPSB:   .WORD   X                ;FPS BEFORE EXECUTION
2612                               ;FPSA:   .WORD   X                ;FPS AFTER EXECUTION
2613                               ;ERRES:  .WORD   X,X,X,X          ;ERROR RESULT
2614                               ;ERR:    ERROR   +X               ;RESULT ERROR
2615                               ;CONT:                            ;RETURN ADDRESS
2616                               ;
2617                               ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
2618                               ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, DIVD IS EXECUTED.
```

```
                                        ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
                                        ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
                                        ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
                                        ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
                                        ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
                                        ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
                                        ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
                                        ;THEN THE FAILURE IS REPORTED IN DIVDSUB AND CONTROL IS PASSED TO
                                        ;CONT. IF NO ERRORS ARE DETECTED THEN DIVDSUB RETURNS CONTROL
                                        ;TO CONT.

016614  012601                  DIVDSUB:      MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
016616  012700  000200                        MOV     #200,R0         ;SET FD MODE.
016622  170100                                LDFPS   R0

016624  010100                                MOV     R1,R0           ;SET UP THE AC0 OPERAND.
016626  172410                                LDD     (R0),AC0
016630  016100  000030                        MOV     30(R1),R0       ;LOAD THE FPS.
016634  170100                                LDFPS   R0

016636  012737  016652  001236                MOV     #1$,@#$TMP2
016644  010100                                MOV     R1,R0           ;ESTABLISH A POINTER TO FSRC.
016646  062700  000010                        ADD     #10,R0

016652  174410                  1$:           DIVD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.

016654  170204                                STFPS   R4              ;GET THE FPS.
016656  012700  000200                        MOV     #200,R0         ;SET FD MODE.
016662  170100                                LDFPS   R0

016664  012700  017044                        MOV     #DIVDT,R0       ;GET THE RESULT.
016670  174010                                STD     AC0,(R0)

016672  010102                                MOV     R1,R2           ;SAVE DATA IN CASE OF ERROR.
016674  010237  001240                        MOV     R2,@#$TMP3
016700  062702  000010                        ADD     #10,R2
016704  010237  001242                        MOV     R2,@#$TMP4
016710  062702  000010                        ADD     #10,R2
016714  010237  001244                        MOV     R2,@#$TMP5
016720  012737  017044  001246                MOV     #DIVDT,@#$TMP6
016726  010437  001250                        MOV     R4,@#$TMP7
016732  016137  000032  001252                MOV     32(R1),@#$TMP10

016740  010102                                MOV     R1,R2           ;CHECK THE RESULT.
016742  062702  000020                        ADD     #20,R2
016746  012703  017044                        MOV     #DIVDT,R3
016752  012705  000004                        MOV     #4,R5
016756  022223                  2$:           CMP     (R2)+,(R3)+
016760  001006                                BNE     10$             ;BRANCH IF RESULT INCORRECT.
016762  077503                                SOB     R5,2$

016764  026104  000032                        CMP     32(R1),R4       ;IS FPS CORRECT?
016770  001023                                BNE     15$             ;BRANCH IF INCORRECT.
016772  000161  000046                        JMP     46(R1)          ;RETURN.

016776  010102                  10$:          MOV     R1,R2           ;WAS INCORRECT RESULT ANTICIPATED?
017000  062702  000034                        ADD     #34,R2
```

```
 2677 017004  012703  017044                    MOV     #DIVDT,R3
 2678 017010  012705  000004                    MOV     #4,R5
 2679 017014  022223                  11$:      CMP     (R2)+,(R3)+
 2680 017016  001005                            BNE     12$             ;BRANCH IF NO.
 2681 017020  077503                            SOB     R5,11$
 2682 017022  010102                            MOV     R1,R2           ;IF THE INCORRECT RESULT WAS
 2683 017024  062702  000044                    ADD     #44,R2          ;ANTICIPATED RETURN TO THE
 2684                                                                   ;ERROR REPORT IN THE CALLING
 2685 017030  000112                            JMP     (R2)            ;ROUTINE.
 2686
 2687 017032                          12$:                              ;REPORT RESULT INCORRECT.
 2688 017032  104033                  13$:      ERROR   +33
 2689 017034  000161  000046          14$:      JMP     46(R1)
 2690
 2691 017040                          15$:                              ;REPORT FPS INCORRECT.
 2692 017040  104034                  16$:      ERROR   +34
 2693 017042  000774                            BR      14$
 2694
 2695 017044  000000  000000  000000  DIVDT:    .WORD   0,0,0,0
      017052  000000
 2696
 2697 017054                          DDDDONE:
      017054  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?).
 2698
 2699
 2707
 2708                                 ;;*****************************************************************
                                      ;*TEST 10       MULF TEST
                                      ;*
                                      ;*THIS IS A TEST OF THE MULF INSTRUCTION. IT MAKES USE OF A SUBROUTINE
                                      ;*TO SET UP THE OPERANDS, EXECUTE THE MULF INSTRUCTION AND CHECK THE
                                      ;*RESULTS.
                                      ;*
                                      ;;*****************************************************************
      017056  000004                  TST10:  SCOPE
 2709
 2710                                 ;MULF WITH (FSRC=AC=0)
 2711 017060                          EEE1:
      017060  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 2712 017062  004737  017640                    JSR     PC,@#MULFSUB
 2713 017066  000000  000000          1$:       .WORD   0,0             ;AC
 2714 017072  000000  000000          2$:       .WORD   0,0             ;FSRC
 2715 017076  000000  000000          3$:       .WORD   0,0             ;RES
 2716 017102  007517                  4$:       7517                    ;FPS BEFORE EXECUTION.
 2717 017104  007504                            7504                    ;FPS AFTER EXECUTION.
 2718 017106  177777  177777          5$:       .WORD   -1,-1
 2719 017112  104037                  6$:       ERROR   +37
 2720
 2721                                 ;MULF WITH (FSRC=0).
 2722 017114                          EEE2:
      017114  104413                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 2723 017116  004737  017640                    JSR     PC,@#MULFSUB
 2724 017122  071625  034435          1$:       .WORD   71625,34435     ;AC
```

```
2725 017126  000000  000000          2$:   .WORD   0,0                ;FSRC
2726 017132  000000  000000          3$:   .WORD   0,0                ;RES
2727 017136  000013                  4$:   13                         ;FPS BEFORE EXECUTION.
2728 017140  000004                        4                          ;FPS AFTER EXECUTION.
2729 017142  177777  177777          5$:   .WORD   -1,-1              ;ERROR RES.
2730 017146  104037                  6$:   ERROR   +37
2731
2732                                 ;MULF WITH (AC=0)
2733 017150                          EEE3:
     017150  104413                        LPERR                      ;SET UP THE LOOP ON ERROR ADDRESS.
2734 017152  004737  017640                JSR     PC,@#MULFSUB
2735 017156  000000  000000          1$:   .WORD   0,0                ;AC
2736 017162  071625  153443          2$:   .WORD   071625,153443      ;FSRC
2737 017166  000000  000000          3$:   .WORD   0,0                ;RES
2738 017172  007500                  4$:   7500                       ;FPS BEFORE EXECUTION.
2739 017174  007504                        7504                       ;FPS AFTER EXECUTION.
2740 017176  177777  177777          5$:   .WORD   -1,-1              ;ERROR RES.
2741 017202  104037                  6$:   ERROR   +37
2742
2743                                 ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN ROUND MODE.
2744 017204                          EEE4:
     017204  104413                        LPERR                      ;SET UP THE LOOP ON ERROR ADDRESS.
2745 017206  004737  017640                JSR     PC,@#MULFSUB
2746 017212  040200  000000          1$:   .WORD   40200,0            ;AC
2747 017216  040177  177777          2$:   .WORD   40177,-1           ;FSRC
2748 017222  040177  177777          3$:   .WORD   40177,-1           ;RES
2749 017226  000017                  4$:   17                         ;FPS BEFORE EXECUTION.
2750 017230  000000                        0                          ;FPS AFTER EXECUTION.
2751 017232  140177  177777          5$:   .WORD   140177,-1          ;ERROR RES.
2752 017236  104041                  6$:   ERROR   +41                ;BAD SIGN.
2753
2754                                 ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN TRUNCATE MODE.
2755 017240                          EEE5:
     017240  104413                        LPERR                      ;SET UP THE LOOP ON ERROR ADDRESS.
2756 017242  004767  000372                JSR     PC,MULFSUB
2757 017246  040177  177777          1$:   .WORD   40177,-1           ;AC
2758 017252  040200  000000          2$:   .WORD   40200,0            ;FSRC
2759 017256  040177  177777          3$:   .WORD   40177,-1           ;RES
2760 017262  000040                  4$:   40                         ;FPS BEFORE EXECUTION.
2761 017264  000040                        40                         ;FPS AFTER EXECUTION.
2762 017266  037777  177777          5$:   .WORD   37777,-1           ;ERROR RES.
2763 017272  104042                  6$:   ERROR   +42                ;ST 252 TO 044 INTO 444 (BUT Y62)
2764                                                                   ;MUL. NORMALIZATION FAILURE.
2765
2766                                 ;MULF WITH BOTH OPERANDS POSITIVE NORMALIZE TEST.
2767 017274                          EEE6:
     017274  104413                        LPERR                      ;SET UP THE LOOP ON ERROR ADDRESS.
2768 017276  004737  017640                JSR     PC,@#MULFSUB
2769 017302  040100  000000          1$:   .WORD   40100,0            ;AC
2770 017306  040100  000000          2$:   .WORD   40100,0            ;FSRC
2771 017312  040020  000000          3$:   .WORD   40020,0            ;RES
2772 017316  000012                  4$:   12                         ;FPS BEFORE EXECUTION.
2773 017320  000000                        0                          ;FPS AFTER EXECUTION.
2774 017322  042040  000000          5$:   .WORD   42040,0            ;ERROR RES.
2775 017326  104043                  6$:   ERROR   +43                ;ST 252 TO 444 INTO 042 (BUT Y62)
2776                                                                   ;MUL. NORMALIZATION FAILURE.
2777
```

```
                                        ;MULF WITH BOTH OPERANDS POSITIVE IN ROUND MODE.
                                        EEE7:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
017330  104413
017332  004737  017640                            JSR     PC,@#MULFSUB
017336  017500  000000                    1$:     .WORD   17500,0         ;AC
017342  023652  125252                    2$:     .WORD   23652,125252    ;FSRC
017346  003177  177777                    3$:     .WORD   3177,-1         ;RES
017352  007417                            4$:     7417                    ;FPS BEFORE EXECUTION.
017354  007400                                    7400                    ;FPS AFTER EXECUTION.
017356  177777  177777                    5$:     .WORD   -1,-1
017362  104037                            6$:     ERROR   +37

                                        ;MULF WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
                                        EEE8:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
017364  104413
017366  004737  017640                            JSR     PC,@#MULFSUB
017372  040342  000000                    1$:     .WORD   40342,0         ;AC
017376  176542  000000                    2$:     .WORD   176542,0        ;FSRC
017402  176707  102000                    3$:     .WORD   176707,102000   ;RES
017406  000007                            4$:     7                       ;FPS BEFORE EXECUTION.
017410  000010                                    10                      ;FPS AFTER EXECUTION.
017412  076507  102000                    7$:     .WORD   76507,102000    ;ERROR RES.
017416  104041                            6$:     ERROR   +41             ;BAD SIGN.

                                        ;MULF WITH AC NEGATIVE AND FSRC POSITIVE IN ROUND MODE.
                                        EEE9:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
017420  104413
017422  004737  017640                            JSR     PC,@#MULFSUB
017426  140200  000000                    1$:     .WORD   140200,0        ;AC
017432  007417  007417                    2$:     .WORD   7417,7417       ;FSRC
017436  107417  007417                    3$:     .WORD   107417,7417     ;RES
017442  000000                            4$:     0                       ;FPS BEFORE EXECUTION.
017444  000010                                    10                      ;FPS AFTER EXECUTION.
017446  007417  007417                    5$:     .WORD   7417,7417       ;ERROR RES.
017452  104041                            6$:     ERROR   +41             ;BAD SIGN.

                                        ;MULF WITH BOTH OPERANDS NEGATIVE IN ROUND MODE.
                                        EEE10:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
017454  104413
017456  004737  017640                            JSR     PC,@#MULFSUB
017462  144600  000000                    1$:     .WORD   144600,0        ;AC
017466  154000  000000                    2$:     .WORD   154000,0        ;FSRC
017472  060400  000000                    3$:     .WORD   60400,0         ;RES
017476  000017                            4$:     17                      ;FPS BEFORE EXECUTION.
017500  000000                                    0                       ;FPS AFTER EXECUTION.
017502  160400  000000                    5$:     .WORD   160400,0        ;ERROR RES.
017506  104041                            6$:     ERROR   +41             ;BAD SIGN.

                                        ;MULF BOTH OPERANDS NEGATIVE IN ROUND MODE.
                                        EEE11:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
017510  104413
017512  004737  017640                            JSR     PC,@#MULFSUB
017516  140300  000000                    1$:     .WORD   140300,0        ;AC
017522  160000  000001                    2$:     .WORD   160000,1        ;FSRC
017526  060100  000002                    3$:     .WORD   60100,2         ;RES
017532  000010                            4$:     10                      ;FPS BEFORE EXECUTION.
017534  000000                                    0                       ;FPS AFTER EXECUTION.
```

```
2830 017536  060100  000001          5$:     .WORD   60100,1         ;ERROR RES.
2831 017542  104044                  6$:     ERROR   +44             ;ROUND FAILURE.
2832
2833                                  ;MULF WITH AC POSITIVE AND FSRC NEGATIVE IN TRUNCATE MODE.
2834 017544                           EEE12:
     017544  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2835 017546  004737  017640                  JSR     PC,@#MULFSUB
2836 017552  060000  000001          1$:     .WORD   60000,1         ;AC
2837 017556  140300  000000          2$:     .WORD   140300,0        ;FSRC
2838 017562  160100  000001          3$:     .WORD   160100,1        ;RES
2839 017566  007547                  4$:     7547                    ;FPS BEFORE EXECUTION.
2840 017570  007550                          7550                    ;FPS AFTER EXECUTION.
2841 017572  160100  000001          5$:     .WORD   160100,1        ;ERROR RES.
2842 017576  104045                  6$:     ERROR   +45             ;TRUNCATION ERROR.
2843
2844                                  ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN ROUND MODE.
2845 017600                           EEE13:
     017600  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2846 017602  004737  017640                  JSR     PC,@#MULFSUB
2847 017606  040277  000000          1$:     .WORD   40277,0         ;AC
2848 017612  060000  000001          2$:     .WORD   60000,1         ;FSRC
2849 017616  060077  000001          3$:     .WORD   60077,1         ;RES
2850 017622  000014                  4$:     14                      ;FPS BEFORE EXECUTION.
2851 017624  000000                          0                       ;FPS AFTER EXECUTION.
2852 017626  060077  000002          5$:     .WORD   60077,2         ;ERROR RES.
2853 017632  104044                  6$:     ERROR   +44             ;ROUND FAILURE. CONSTANT BAD.
2854
2855 017634  000167  000224                  JMP     EEEDONE         ;GO TO THE NEXT TEST.
2856
2857                                  ;THIS SUBROUTINE, MULFSUB, IS CALLED TO SET UP, EXECUTE
2858                                  ;AND CHECK THE RESULT OF A MULF INSTRUCTION. IT IS CALLED THUS:
2859                                  ;
2860                                  ;               JSR     PC,@#MULFSUB
2861                                  ;       ACARG:  .WORD   X,X             ;AC OPERAND
2862                                  ;       FSRCARG:.WORD   X,X             ;FSRC OPERAND
2863                                  ;       RES:    .WORD   X,X             ;EXPECTED RESULT
2864                                  ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
2865                                  ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
2866                                  ;       ERRES:  .WORD   X,X             ;ERROR RESULT
2867                                  ;       ERR:    ERROR   +X              ;RESULT ERROR
2868                                  ;       CONT:                           ;RETURN ADDRESS
2869                                  ;
2870                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
2871                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, MULF IS EXECUTED.
2872                                  ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
2873                                  ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
2874                                  ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
2875                                  ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
2876                                  ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
2877                                  ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
2878                                  ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
2879                                  ;THEN THE FAILURE IS REPORTED IN MULFSUB AND CONTROL IS PASSED TO
2880                                  ;CONT. IF NO ERRORS ARE DETECTED THEN MULFSUB RETURNS CONTROL
2881                                  ;TO CONT.
2882
2883 017640  012601                  MULFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
2884 017642  012700  000200                  MOV     #200,R0         ;SET FD MODE.
```

```
2885 017646  170100                        LDFPS   R0
2886 017650  010100                        MOV     R1,R0           ;LOAD THE AC OPERAND.
2887 017652  172410                        LDD     (R0),AC0
2888 017654  016100  000014                MOV     14(R1),R0       ;LOAD THE FPS
2889 017660  170100                        LDFPS   R0
2890 017662  012737  017676  001236        MOV     #1$,@#$TMP2
2891 017670  010100                        MOV     R1,R0
2892 017672  062700  000004                ADD     #4,R0           ;ESTABLISH A POINTER TO FSRC.
2893
2894 017676  171010              1$:       MULF    (R0),AC0        ;TEST INSTRUCTION.
2895
2896 017700  170204                        STFPS   R4              ;GET THE FPS.
2897 017702  012700  000200                MOV     #200,R0         ;SET FD MODE
2898 017706  170100                        LDFPS   R0
2899
2900 017710  012700  020054                MOV     #MULFT,R0       ;GET THE RESULT OF THE MULF.
2901 017714  174010                        STD     AC0,(R0)
2902
2903 017716  010102                        MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
2904 017720  010237  001240                MOV     R2,@#$TMP3
2905 017724  062702  000004                ADD     #4,R2
2906 017730  010237  001242                MOV     R2,@#$TMP4
2907 017734  062702  000004                ADD     #4,R2
2908 017740  010237  001244                MOV     R2,@#$TMP5
2909 017744  012737  020054  001246        MOV     #MULFT,@#$TMP6
2910 017752  010437  001250                MOV     R4,@#$TMP7
2911 017756  016137  000016  001252        MOV     16(R1),@#$TMP10
2912
2913 017764  021061  000010                CMP     (R0),10(R1)     ;IS THE RESULT CORRECT?
2914 017770  001011                        BNE     10$             ;IF INCORRECT BRANCH.
2915 017772  026061  000002  000012        CMP     2(R0),12(R1)
2916 020000  001005                        BNE     10$
2917
2918 020002  026104  000016                CMP     16(R1),R4       ;IS FPS CORRECT?
2919 020006  001020                        BNE     15$             ;IF INCORRECT BRANCH.
2920 020010  000161  000026                JMP     26(R1)          ;IF NO ERRORS OCCURRED RETURN.
2921
2922 020014  021061  000020      10$:      CMP     (R0),20(R1)     ;DOES THE INCORRECT RESULT
2923 020020  001010                        BNE     11$             ;MATCH THE ANTICIPATED INCORRECT RESULT.
2924 020022  026061  000002  000022        CMP     2(R0),22(R1)
2925 020030  001004                        BNE     11$             ;BRANCH IF NO.
2926
2927 020032  010102                        MOV     R1,R2           ;IT MATCHED SO RETURN TO THE ERROR
2928                                                              ;REPORT AT THE CALLING ROUTINE.
2929 020034  062702  000024                ADD     #24,R2
2930 020040  000112                        JMP     (R2)
2931
2932 020042                      11$:                             ;REPORT RESULT INCORRECT.
2933 020042  104037              12$:      ERROR   +37
2934 020044  000161  000026      13$:      JMP     26(R1)
2935
2936 020050                      15$:                             ;REPORT FPS INCORRECT.
2937 020050  104040              16$:      ERROR   +40
2938 020052  000774                        BR      13$
2939
2940 020054  000000  000000  000000  MULFT: .WORD   0,0,0,0
     020062  000000
```

```
                                EEEDONE:
020064  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 11       MULD TEST
                                ;*
                                ;*THIS IS A TEST OF THE MULD INSTRUCTION. NOTE THAT A SUBROUTINE IS
                                ;*USED TO SET UP THE OPERANDS, EXECUTE THE MULD INSTRUCTION AND
                                ;*CHECK THE RESULTS.
                                ;*
                                ;;*****************************************************************
020066  000004                  TST11:  SCOPE

                                ;MULD TEST WITH AC POSITIVE AND FSRC POSITIVE.
                                FFF1:
020070  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
020072  004737  020354                   JSR     PC,@#MULDSUB
020076  040200  000000  000000  1$:      .WORD   40200,0,0,0     ;AC
020104  000000
020106  023777  177777  177777  2$:      .WORD   23777,-1,-1,-1  ;FSRC
020114  177777
020116  023777  177777  177777  3$:      .WORD   23777,-1,-1,-1  ;RES
020124  177777
020126  000217                  4$:      217                     ;FPS BEFORE EXECUTION.
020130  000200                           200                     ;FPS AFTER ECECUTION.
020132  023777  177777  000000  5$:      .WORD   23777,-1,0,0    ;ERROR RES.
020140  000000
020142  104047                  6$:      ERROR   +47             ;BAD CONSTANT USED IN ALGORITHM
                                                                ;USED 24 INSTEAD OF 56.


                                ;MULD TEST WITH BOTH OPERANDS POSITIVE TRUNCATION TEST.
                                FFF2:
020144  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
020146  004767  000202                   JSR     PC,MULDSUB
020152  065400  000000  000000  1$:      .WORD   65400,0,0,1     ;AC
020160  000001
020162  037577  177777  177777  2$:      .WORD   37577,-1,-1,-2  ;FSRC
020170  177776
020172  064777  177777  177777  3$:      .WORD   64777,-1,-1,-1  ;RES
020200  177777
020202  000247                  4$:      247                     ;FPS BEFORE EXECUTION.
020204  000240                           240                     ;FPS AFTER EXECUTION.
020206  065000  000000  000000  5$:      .WORD   65000,0,0,0     ;ERROR RES.
020214  000000
020216  104050                  6$:      ERROR   +50             ;TRUNCATION ERROR.


                                ;MULD TEST WITH BOTH OPERANDS NEGATIVE IN ROUND MODE.
                                FFF3:
020220  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
020222  004737  020354                   JSR     PC,@#MULDSUB
```

```
 2981 020226  137577  177777  177777  1$:     .WORD   137577,-1,-1,-2 ;AC
      020234  177776
 2982 020236  165400  000000  000000  2$:     .WORD   165400,0,0,1    ;FSRC
      020244  000001
 2983 020246  065000  000000  000000  3$:     .WORD   65000,0,0,0     ;RES
      020254  000000
 2984 020256  007717                  4$:     7717                    ;FPS BEFORE EXECUTION.
 2985 020260  007700                          7700                    ;FPS AFTER EXECUTION.
 2986 020262  064777  177777  177777  5$:     .WORD   64777,-1,-1,-1  ;ERROR RES.
      020270  177777
 2987 020272  104051                  6$:     ERROR   +51             ;ROUND ERROR.
 2988
 2989                                 ;MULD   TEST WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
 2990 020274                          FFF4:
      020274  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 2991 020276  004737  020354                  JSR     PC,@#MULDSUB
 2992 020302  017500  000000  000000  1$:     .WORD   17500,0,0,0     ;AC
      020310  000000
 2993 020312  123652  125252          2$:     .WORD   123652,125252   ;FSRC
 2994 020316  125252  125252                  .WORD   125252,125252
 2995 020322  103177  177777  177777  3$:     .WORD   103177,-1,-1,-1 ;RES
      020330  177777
 2996 020332  000200                  4$:     200                     ;FPS BEFORE EXECUTION.
 2997 020334  000210                          210                     ;FPS AFTER EXECUTION.
 2998 020336  103200  000000  000000  5$:     .WORD   103200,0,0,0    ;ERROR RES.
      020344  000000
 2999 020346  104052                  6$:     ERROR   +52             ;ROUND ERROR (BAD CONSTANT).
 3000
 3001 020350  000167  000240                  JMP     FFFDONE
 3002
 3003                                 ;THIS SUBROUTINE, MULDSUB, IS CALLED TO SET UP, EXECUTE
 3004                                 ;AND CHECK THE RESULT OF A MULD INSTRUCTION. IT IS CALLED THUS:
 3005                                 ;
 3006                                 ;               JSR     PC,@#MULDSUB
 3007                                 ;       ACARG:  .WORD   X,X,X,X         ;AC OPERAND
 3008                                 ;       FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
 3009                                 ;       RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
 3010                                 ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
 3011                                 ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
 3012                                 ;       ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
 3013                                 ;       ERR:    ERROR   +X              ;RESULT ERROR
 3014                                 ;       CONT:                           ;RETURN ADDRESS
 3015                                 ;
 3016                                 ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
 3017                                 ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, MULD IS EXECUTED.
 3018                                 ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
 3019                                 ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
 3020                                 ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
 3021                                 ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
 3022                                 ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
 3023                                 ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
 3024                                 ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
 3025                                 ;THEN THE FAILURE IS REPORTED IN MULDSUB AND CONTROL IS PASSED TO
 3026                                 ;CONT. IF NO ERRORS ARE DETECTED THEN MULDSUB RETURNS CONTROL
 3027                                 ;TO CONT.
 3028
 3029 020354  012601                  MULDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
```

```
3030 020356  012700  000200                MOV     #200,R0         ;SET FD MODE.
3031 020362  170100                        LDFPS   R0
3032
3033 020364  010100                        MOV     R1,R0           ;SET UP THE AC0 OPERAND.
3034 020366  172410                        LDD     (R0),AC0
3035 020370  016100  000030                MOV     30(R1),R0       ;LOAD THE FPS.
3036 020374  170100                        LDFPS   R0
3037
3038 020376  012737  020412  001236        MOV     #1$,@#$TMP2
3039 020404  010100                        MOV     R1,R0           ;ESTABLISH A POINTER TO FSRC.
3040 020406  062700  000010                ADD     #10,R0
3041
3042 020412  171010              1$:       MULD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
3043
3044 020414  170204                        STFPS   R4              ;GET THE FPS.
3045 020416  012700  000200                MOV     #200,R0         ;SET FD MODE.
3046 020422  170100                        LDFPS   R0
3047
3048 020424  012700  020604                MOV     #MULDT,R0       ;GET THE RESULT.
3049 020430  174010                        STD     AC0,(R0)
3050
3051 020432  010102                        MOV     R1,R2           ;SAVE DATA IN CASE OF ERROR.
3052 020434  010237  001240                MOV     R2,@#$TMP3
3053 020440  062702  000010                ADD     #10,R2
3054 020444  010237  001242                MOV     R2,@#$TMP4
3055 020450  062702  000010                ADD     #10,R2
3056 020454  010237  001244                MOV     R2,@#$TMP5
3057 020460  012737  020604  001246        MOV     #MULDT,@#$TMP6
3058 020466  010437  001250                MOV     R4,@#$TMP7
3059 020472  016137  000032  001252        MOV     32(R1),@#$TMP10
3060
3061 020500  010102                        MOV     R1,R2           ;CHECK THE RESULT.
3062 020502  062702  000020                ADD     #20,R2
3063 020506  012703  020604                MOV     #MULDT,R3
3064 020512  012705  000004                MOV     #4,R5
3065 020516  022223              2$:       CMP     (R2)+,(R3)+
3066 020520  001006                        BNE     10$             ;BRANCH IF RESULT INCORRECT.
3067 020522  077503                        SOB     R5,2$
3068
3069 020524  026104  000032                CMP     32(R1),R4       ;IS FPS CORRECT?
3070 020530  001023                        BNE     15$             ;BRANCH IF INCORRECT.
3071 020532  000161  000046                JMP     46(R1)          ;RETURN.
3072
3073 020536  010102              10$:      MOV     R1,R2           ;WAS INCORRECT RESULT ANTICIPATED?
3074 020540  062702  000034                ADD     #34,R2
3075 020544  012703  020604                MOV     #MULDT,R3
3076 020550  012705  000004                MOV     #4,R5
3077 020554  022223              11$:      CMP     (R2)+,(R3)+
3078 020556  001005                        BNE     12$             ;BRANCH IF NO.
3079 020560  077503                        SOB     R5,11$
3080 020562  010102                        MOV     R1,R2           ;IF THE INCORRECT RESULT WAS
3081 020564  062702  000044                ADD     #44,R2          ;ANTICIPATED RETURN TO THE
3082                                                               ;ERROR REPORT IN THE CALLING
3083 020570  000112                        JMP     (R2)            ;ROUTINE.
3084
3085 020572              12$:                                      ;REPORT RESULT INCORRECT.
3086 020572  104246      13$:      ERROR   +246
```

```
 3087 020574  000161  000046          14$:    JMP     46(R1)
 3088
 3089 020600                          15$:                            ;REPORT FPS INCORRECT.
 3090 020600  104046                  16$:    ERROR   +46
 3091 020602  000774                          BR      14$
 3092
 3093 020604  000000  000000  000000  MULDT:  .WORD   0,0,0,0
      020612  000000
 3094
 3095 020614                          FFFDONE:
      020614  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?).
 3096
 3097
 3106
 3107                                         ;TEST TITLE:UNDER/OVERFLOW, USING MULF WITH TRAPS DISABLED
 3108                                 ;;*****************************************************************
                                      ;*TEST 12       SEE COMMENT ABOVE FOR TEST TITLE
                                      ;*
                                      ;*THIS IS A TEST OF THE OVERFLOW AND UNDERFLOW CONDITIONS USING
                                      ;*THE MULF INSTRUCTION WITH TRAPS DISABLED. NOTE THAT A SUBROUTINE
                                      ;*IS USED TO SET UP THE OPERANDS, EXECUTE THE MULF INSTRUCTION AND
                                      ;*CHECK THE RESULTS.
                                      ;*
                                      ;;*****************************************************************
      020616  000004                  TST12:  SCOPE
 3109
 3110                                 ;UNDERFLOW, WITH EXPONENT OF RESULT = -129
 3111 020620                          III1:
      020620  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 3112 020622  004737  021044                  JSR     PC,@#OVUNFNT
 3113 020626  020200  000000          1$:     .WORD   20200,0         ;AC
 3114 020632  020000  000000          2$:     .WORD   20000,0         ;FSRC
 3115 020636  000000  000000          3$:     .WORD   0,0             ;RES
 3116 020642  177777  177777          4$:     .WORD   -1,-1           ;ERROR RES.
 3117 020646  000000                  5$:     0                       ;FPS BEFORE EXECUTION.
 3118 020650  000004                          4                       ;FPS AFTER EXECUTION.
 3119 020652  000012                  6$:     12                      ;FEC
 3120 020654  177777                          -1                      ;FLAG
 3121 020656  104117                  7$:     ERROR   +117            ;ST 331 TO 155 INTO 115 (BUT FIU)
 3122 020660  000401                          BR      8$
 3123 020662  104114                          ERROR   +114
 3124 020664                          8$:
 3125
 3126                                 ;UNDERFLOW, WITH EXPONENT OF RESULT = -193
 3127 020664                          III2:
      020664  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 3128 020666  004737  021044                  JSR     PC,@#OVUNFNT
 3129 020672  010200  000000          1$:     .WORD   10200,0         ;AC
 3130 020676  010000  000000          2$:     .WORD   10000,0         ;FSRC
 3131 020702  000000  000000          3$:     .WORD   0,0             ;RES
 3132 020706  010000  000000          4$:     .WORD   10000,0         ;ERROR RES.
 3133 020712  005013                  5$:     5013                    ;FPS BEFORE EXECUTION.
 3134 020714  005004                          5004                    ;FPS AFTER EXECUTION.
```

```
3135 020716  000012                  6$:     12                      ;FEC
3136 020720  177777                          -1                      ;FLAG
3137 020722  104120                  7$:     ERROR   +120            ;SETTING FIUV OR FIV CAUSES TRAP
3138                                                                 ;WITH FIU CLEAR.
3139 020724  000401                          BR      8$
3140 020726  104114                          ERROR   +114
3141 020730                          8$:
3142
3143                                 ;OVERFLOW, EXPONENT OF RESULT = 128
3144 020730                          III3:
     020730  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3145 020732  004737  021044                  JSR     PC,@#OVUNFNT
3146 020736  060200  000000          1$:     .WORD   60200,0         ;AC
3147 020742  060000  000000          2$:     .WORD   60000,0         ;FSRC
3148 020746  000000  000000          3$:     .WORD   0,0             ;RES
3149 020752  060000  000000          4$:     .WORD   60000,0         ;ERROR RES.
3150 020756  000000                  5$:     0                       ;FPS BEFORE EXECUTION.
3151 020760  000006                          6                       ;FPS AFTER EXECUTION.
3152 020762  000010                  6$:     10                      ;FEC
3153 020764  000000                          0                       ;FLAG
3154 020766  104121                  7$:     ERROR   +121            ;ST 333 TO 136 INTO 116 (BUT FIV).
3155 020770  000401                          BR      8$
3156 020772  104113                          ERROR   +113
3157 020774                          8$:
3158
3159                                 ;OVERFLOW, EXPONENT OF RESULT = 130
3160 020774                          III4:
     020774  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3161 020776  004737  021044                  JSR     PC,@#OVUNFNT
3162 021002  060200  000000          1$:     .WORD   60200,0         ;AC
3163 021006  060200  000000          2$:     .WORD   60200,0         ;FSRC
3164 021012  000000  000000          3$:     .WORD   0,0             ;RES
3165 021016  177777  177777          4$:     .WORD   -1,-1           ;ERROR RES.
3166 021022  006011                  5$:     6011                    ;FPS BEFORE EXECUTION.
3167 021024  006006                          6006                    ;FPS AFTER EXECUTION.
3168 021026  000010                  6$:     10                      ;FEC
3169 021030  000000                          0                       ;FLAG
3170 021032  104122                  7$:     ERROR   +122            ;SETTING FIUV OR FIU WITH
3171                                                                 ;FIV CLEAR CAUSES TRAP.
3172 021034  000401                          BR      8$
3173 021036  104113                          ERROR   +113
3174 021040  000167  000410          8$:     JMP     IIIDONE         ;GO TO NEXT TEST.
3175
3176
3177
3178                                 ;THIS SUBROUTINE, OVUNFNT, IS USED TO SET UP THE OPERANDS, EXECUTE
3179                                 ;THE MULF INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
3180                                 ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
3181                                 ;TO IT IS MADE THUS:
3182                                 ;
3183                                 ;               ACARG:  .WORD   X,X             ;AC OPERAND
3184                                 ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
3185                                 ;               RES:    .WORD   X,X             ;EXPECTED RESULT
3186                                 ;               ERRES:  .WORD   X,X             ;ERROR RESULT
3187                                 ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
3188                                 ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
3189                                 ;               FEC:    .WORD   X               ;EXPECTED FEC
```

```
                                        ;              FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
                                        ;              ERR1:   ERROR   +X              ;TRAP ERROR.
                                        ;              BR      CONT
                                        ;              ERR2:   ERROR   +X              ;DATA, RESULT ERROR
                                        ;              CONT:                           ;RETURN ADDRESS
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                        ;THE MULF INSTRUCTION IS EXECUTED. IF NO TRAP OCCURS THEN THE
                                        ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                        ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNFNT RETURNS CONTROL
                                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNFNT
                                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                        ;MULF 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 OVUNFNT
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNFNT WILL
                                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
                                        ;IF A TRAP OCCURS (IT SHOULD NOT) THEN OVUNFNT WILL READ THE FEC.
                                        ;SHOULD THE FEC MATCH THE ANTICIPATED FEC OVUNFNT WILL
                                        ;STORE ALL DATA AND TRANSFER CONTROL TO THE ERROR CALL AT ERR1. IF THE
                                        ;FEC IS NOT THE SAME AS THE ANTICIPATED FEC OVUNFNT WILL REPORT
                                        ;THE ERROR AND RETURN TO CONT. NOTE THAT OVUNFNT USES THE FLAG
                                        ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
                                        ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).

021044  012601                          OVUNFNT:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
021046  012700  000200                          MOV     #200,R0         ;SET FD MODE.
021052  170100                          MOV     LDFPS   R0

021054  010100                          MOV     R1,R0           ;LOAD AC0, OPERAND.
021056  172410                          MOV     LDD     (R0),AC0

021060  010102                          MOV     R1,R2           ;SAVE THE DATA PATTERNS IN CASE OF
021062  010237  001240                  MOV     R2,@#$TMP3      ;ERROR.
021066  062702  000004                  MOV     ADD     #4,R2
021072  010237  001242                  MOV     R2,@#$TMP4
021076  062702  000004                  MOV     ADD     #4,R2
021102  010237  001244                  MOV     R2,@#$TMP5
021106  016137  000022  001252          MOV     22(R1),@#$TMP10
021114  012737  021444  001246          MOV     #OVFNTT,@#$TMP6

021122  016100  000020                  MOV     20(R1),R0       ;LOAD THE FPS.
021126  170100                          MOV     LDFPS   R0
021130  012737  021152  001236          MOV     #1$,@#$TMP2
021136  012737  021336  000244          MOV     #25$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
                                                                ;OF ERROR.
021144  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
021146  062700  000004                  MOV     ADD     #4,R0

021152  171010                  1$:     MULF    (R0),AC0        ;TEST INSTRUCTION.

021154  170204                  2$:     STFPS   R4              ;GET FPS.
021156  170305                          STST    R5              ;GET FEC.
021160  012700  000200                  MOV     #200,R0         ;SET FD MODE.
021164  170100                          MOV     LDFPS   R0
021166  012700  021444                  MOV     #OVFNTT,R0      ;GET THE RESULT.
```

```
3247 021172  174010                        STD     AC0,(R0)
3248 021174  010437  001250                MOV     R4,@#$TMP7
3249 021200  010537  001254                MOV     R5,@#$TMP11
3250
3251 021204  012700  021444                MOV     #OVFNTT,R0      ;CHECK THE RESULT.
3252 021210  010102                        MOV     R1,R2
3253 021212  062702  000010                ADD     #10,R2
3254 021216  012703  000002                MOV     #2,R3
3255 021222  022022              3$:       CMP     (R0)+,(R2)+
3256 021224  001015                        BNE     15$             ;BRANCH IF INCORRECT.
3257 021226  077303                        SOB     R3,3$
3258
3259 021230  026104  000022                CMP     22(R1),R4       ;WAS FPS CORRECT?
3260 021234  001002                        BNE     10$             ;BRANCH IF FPS IS INCORRECT.
3261
3262 021236  000161  000036          4$:   JMP     36(R1)          ;RETURN, TEST COMPLETED.
3263
3264                                  ;REPORT INCORRECT FPS.
3265 021242  005761  000026          10$:  TST     26(R1)          ;WAS THE RESULT OVER OR UNDER FLOW?
3266 021246  001002                        BNE     12$             ;BRANCH IF UNDERFLOW.
3267
3268                                                              ;REPORT FPS BAD AFTER OVERFLOW.
3269 021250  104111              11$:      ERROR   +111
3270 021252  000771                        BR      4$
3271
3272 021254                      12$:                            ;REPORT FPS BAD AFTER UNDERFLOW.
3273 021254  104112              13$:      ERROR   +112
3274 021256  000767                        BR      4$
3275
```

```
                                        ;RESULT INCORRECT.
021260  012700  021444          15$:    MOV     #OVFNTT,R0      ;SEE IF FAILURE IS ANTICIPATED
021264  010102                          MOV     R1,R2           ;FAILURE.
021266  062702  000014                  ADD     #14,R2
021272  012703  000002                  MOV     #2,R3
021276  022022                  16$:    CMP     (R0)+,(R2)+
021300  001007                          BNE     17$             ;BRANCH IF NOT ANTICIPATED.
021302  077303                          SOB     R3,16$
021304  010102                          MOV     R1,R2           ;ERROR WAS ANTICIPATED SO RETURN
021306  062702  000034                  ADD     #34,R2          ;TO THE ERROR REPORT IN THE CALLING
021312  010237  001236                  MOV     R2,@#$TMP2      ;ROUTINE.
021316  000112                          JMP     (R2)
021320  005761  000026          17$:    TST     26(R1)          ;RESULT WAS NOT ANTICIPATED
                                                                ;SO ERROR MUST BE REPORTED HERE.
                                                                ;FIRST SEE IF ARGUMENTS SHOULD
                                                                ;HAVE RESULTED IN OVERFLOW OR UNDER
                                                                ;FLOW BY LOOKING AT THE FLAG.
021324  001002                          BNE     19$             ;BRANCH IF UNDERFLOW EXPECTED.

                                                                ;REPORT RESULT INCORRECT, EXPECTING
021326  104113                  18$:    ERROR   +113            ;OVERFLOW.
021330  000742                          BR      4$
                                19$:                            ;REPORT RESULT INCORRECT, EXPECTING
021332  104114                  20$:    ERROR   +114            ;UNDERFLOW.
021334  000740                          BR      4$

                                        ;IF AN FP TRAP OCCURS COME HERE.
021336  011602                  25$:    MOV     (SP),R2         ;GET ADDRESS OF TRAP.
021340  022702  021154                  CMP     #2$,R2          ;WAS THE TRAP DURING THE MULF INSTRUCTION?
021344  001402                          BEQ     26$             ;BRANCH IF YES.
021346  000137  036660                  JMP     @#FPSPUR        ;OTHERWISE GO REPORT A SPURIOUS
                                                                ;FP TRAP.
021352  022626                  26$:    CMP     (SP)+,(SP)+     ;RESET THE STACK.
021354  010237  001236                  MOV     R2,@#$TMP2      ;SAVE DATA FOR ERROR REPORT.
021360  170204                          STFPS   R4              ;GET FPS.
021362  170305                          STST    R5              ;GET FEC.
021364  012700  000200                  MOV     #200,R0         ;SET FD MODE.
021370  170100                          LDFPS   R0
021372  012700  021444                  MOV     #OVFNTT,R0      ;GET THE RESULT.
021376  174010                          STD     AC0,(R0)
021400  010537  001254                  MOV     R5,@#$TMP11
021404  020561  000024                  CMP     R5,24(R1)       ;WAS THE FEC ANTICIPATED?
021410  001004                          BNE     27$             ;BRANCH IF NOT ANTICIPATED.

021412  010102                          MOV     R1,R2           ;ERROR WAS ANTICIPATED SO
021414  062702  000030                  ADD     #30,R2          ;RETURN TO THE ERROR REPORT OF THE
                                                                ;CALLING ROUTINE.
021420  000112                          JMP     (R2)

021422  005761  000026          27$:    TST     26(R1)          ;THE ERROR WAS NOT ANTICIPATED SO
                                                                ;IT MUST BE REPORTED HERE. FIRST SEE IF EXPECTED
                                                                ;OVERFLOW OR UNDER FLOW.
021426  001003                          BNE     29$             ;BRANCH IF EXPECTING UNDERFLOW
```

```
3334                                                    ;REPORT TRAPPED ON OVERFLOW WITH FIV=0
3335 021430  104115                    28$:    ERROR   +115
3336 021432  000161  000036                    JMP     36(R1)
3337
3338 021436                            29$:                                    ;REPORT TRAPPED ON UNDER FLOW WITH FIU=0
3339 021436  104116                    30$:    ERROR   +116
3340 021440  000161  000036                    JMP     36(R1)
3341
3342 021444  000000  000000  000000  .QVFNTT: .WORD   0,0,0,0
     021452  000000
3343
3344 021454                            IIIDONE:
     021454  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?).
3345
3346
3347
3356
3357                                    ;TEST TITLE:UNDER/OVERFLOW, USING MULD WITH TRAP DISABLED
3358                                    ;;*****************************************************************
                                        ;*TEST 13       SEE COMMENT ABOVE FOR TEST TITLE
                                        ;*
                                        ;*THIS IS A TEST OF THE OVERFLOW AND UNDERFLOW CONDITIONS THAT CAN
                                        ;*ARRISE USING THE MULD INSTRUCTION WITH TRAPS DISABLED. A SUBROUTINE IS
                                        ;*USED TO SET UP THE OPERANDS, EXECUTE THE MULD INSTRUCTION AND
                                        ;*CHECK THE RESULTS.
                                        ;*
                                        ;;*****************************************************************
     021456  000004                     TST13:  SCOPE
3359
3360                                    ;UNDERFLOW, EXPONENT OF RESULT=-129
3361 021460                             JJJ1:
     021460  104413                            LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
3362 021462  004737  022004                    JSR     PC,@#OVUNDNT
3363 021466  020200  000000            1$:     .WORD   20200,0                 ;AC
3364 021472  127272  000000                    .WORD   127272,0
3365 021476  020000  000000  000000    2$:     .WORD   20000,0,0,0             ;FSRC
     021504  000000
3366 021506  000000  000000  000000    3$:     .WORD   0,0,0,0                 ;RES
     021514  000000
3367 021516  000000  000000            4$:     .WORD   0,0                     ;ERROR RES.
3368 021522  127272  000000                    .WORD   127272,0
3369 021526  000200                    5$:     200                             ;FPS BEFORE EXECUTION.
3370 021530  000204                            204                             ;FPS AFTER EXECUTION.
3371 021532  000012                    6$:     12                              ;FEC
3372 021534  177777                            -1                              ;FLAG
3373 021536  104131                    7$:     ERROR   +131                    ;ST 331 TO 155 INTO 115 (BUT FIU)
3374 021540  000401                            BR      8$
3375 021542  104132                            ERROR   +132                    ;ST 115 (BUT FD)
3376 021544                            8$:
3377
3378                                    ;UNDERFLOW, EXPONENT OF RESULT = -193
3379 021544                             JJJ2:
     021544  104413                            LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
3380 021546 004737 022004                    JSR     PC,@#OVUNDNT
3381 021552 010200 000000          1$:       .WORD   10200,0          ;AC
3382 021556 123456 000000                    .WORD   123456,0
3383 021562 010000 000000 000000   2$:       .WORD   10000,0,0,0      ;FSRC
     021570 000000
3384 021572 000000 000000 000000   3$:       .WORD   0,0,0,0          ;RES
     021600 000000
3385 021602 000000 000000 123456   4$:       .WORD   0,0,123456,0     ;ERROR RES
     021610 000000
3386 021612 005213                 5$:       5213                     ;FPS BEFORE EXECUTION.
3387 021614 005204                           5204                     ;FPS AFTER EXECUTION.
3388 021616 000012                 6$:       12                       ;FEC
3389 021620 177777                           -1                       ;FLAG
3390 021622 104133                 7$:       ERROR   +133             ;SETTING FIUV OR FIV BAD.
3391 021624 000401                           BR      8$
3392 021626 104132                           ERROR   +132             ;ST 115 (BUT FD)
3393 021630                        8$:
3394
3395                               ;OVERFLOW, EXPONENT OF RESULT = 128
3396 021630                        JJJ3:
     021630 104413                           LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
3397 021632 004737 022004                    JSR     PC,@#OVUNDNT
3398 021636 060200 000000          1$:       .WORD   60200,0          ;AC
3399 021642 065432 000000                    .WORD   65432,0
3400 021646 060000 000000 000000   2$:       .WORD   60000,0,0,0      ;FSRC
     021654 000000
3401 021656 000000 000000 000000   3$:       .WORD   0,0,0,0          ;RES
     021664 000000
3402 021666 000000 000000 065432   4$:       .WORD   0,0,65432,0      ;ERROR RES.
     021674 000000
3403 021676 000200                 5$:       200                      ;FPS BEFORE EXECUTION.
3404 021700 000206                           206                      ;FPS AFTER EXECUTION.
3405 021702 000010                 6$:       10                       ;FEC
3406 021704 000000                           0                        ;FLAG
3407 021706 104134                 7$:       ERROR   +134             ;ST 333 TO 136 INTO 116 (BUT FIV)
3408 021710 000401                           BR      8$
3409 021712 104135                           ERROR   +135             ;ST 116 (BUT FD)
3410 021714                        8$:
3411
3412
3413                               ;OVERFLOW, EXPONENT OF RESULT = 130
3414 021714                        JJJ4:
     021714 104413                           LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
3415 021716 004737 022004                    JSR     PC,@#OVUNDNT
3416 021722 060200 000000          1$:       .WORD   60200,0          ;AC
3417 021726 125252 000000                    .WORD   125252,0
3418 021732 060200 000000 000000   2$:       .WORD   60200,0,0,0      ;FSRC
     021740 000000
3419 021742 000000 000000 000000   3$:       .WORD   0,0,0,0          ;RES
     021750 000000
3420 021752 000000 000000 125252   4$:       .WORD   0,0,125252,0     ;ERROR RES.
     021760 000000
3421 021762 006211                 5$:       6211                     ;FPS BEFORE EXECUTION.
3422 021764 006206                           6206                     ;FPS AFTER EXECUTION.
3423 021766 000010                 6$:       10                       ;FEC
3424 021770 000000                           0                        ;FLAG
3425 021772 104136                 7$:       ERROR   +136             ;SETTING FIUV OR FIV BAD.
```

```
3426 021774  000401                        BR      8$
3427 021776  104135                        ERROR   +135            ;ST 116 (BUT FD)
3428 022000  000137  022414          8$:    JMP     @#JJJDONE       ;GO TO NEXT TEST.
3429
3430                                ;THIS SUBROUTINE, OVUNDNT, IS USED TO SET UP THE OPERANDS, EXECUTE
3431                                ;THE MULD INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
3432                                ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
3433                                ;TO IT IS MADE THUS:
3434                                ;
3435                                ;       ACARG:   .WORD   X,X,X,X        ;AC OPERAND
3436                                ;       FSRCARG:.WORD    X,X,X,X        ;FSRC OPERAND
3437                                ;       RES:     .WORD   X,X,X,X        ;EXPECTED RESULT
3438                                ;       ERRES:   .WORD   X,X,X,X        ;ERROR RESULT
3439                                ;       FPSB:    .WORD   X              ;FPS BEFORE EXECUTION
3440                                ;       FPSA:    .WORD   X              ;FPS AFTER EXECUTION
3441                                ;       FEC:     .WORD   X              ;EXPECTED FEC
3442                                ;       FLAG:    .WORD   X              ;0/-1,OVER/UNDER FLOW FLAG
3443                                ;       ERR1:    ERROR   +X             ;TRAP ERROR.
3444                                ;       BR       CONT
3445                                ;       ERR2:    ERROR   +X             ;DATA, RESULT ERROR
3446                                ;       CONT:                           ;RETURN ADDRESS
3447                                ;
3448                                ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
3449                                ;THE MULD INSTRUCTION IS EXECUTED. IF NO TRAP OCCURS THEN THE
3450                                ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
3451                                ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNDNT RETURNS CONTROL
3452                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNDNT
3453                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
3454                                ;MULD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
3455                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
3456                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNDNT
3457                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
3458                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNDNT WILL
3459                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
3460                                ;IF A TRAP OCCURS (IT SHOULD NOT) THEN OVUNDNT WILL READ THE FEC.
3461                                ;SHOULD THE FEC MATCH THE ANTICIPATED FEC OVUNDNT WILL
3462                                ;STORE ALL DATA AND TRANSFER CONTROL TO THE ERROR CALL AT ERR1. IF THE
3463                                ;FEC IS NOT THE SAME AS THE ANTICIPATED FEC OVUNDNT WILL REPORT
3464                                ;THE ERROR AND RETURN TO CONT. NOTE THAT OVUNDNT USES THE FLAG
3465                                ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
3466                                ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).
3467
3468 022004  012601                  OVUNDNT:      MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
3469 022006  012700  000200                        MOV     #200,R0         ;SET FD MODE.
3470 022012  170100                              LDFPS   R0
3471
3472 022014  010100                              MOV     R1,R0           ;LOAD AC0, OPERAND.
3473 022016  172410                              LDD     (R0),AC0
3474
3475 022020  010102                              MOV     R1,R2           ;SAVE THE DATA PATTERNS IN CASE OF
3476 022022  010237  001240                      MOV     R2,@#$TMP3      ;ERROR.
3477 022026  062702  000010                      ADD     #10,R2
3478 022032  010237  001242                      MOV     R2,@#$TMP4
3479 022036  062702  000010                      ADD     #10,R2
3480 022042  010237  001244                      MOV     R2,@#$TMP5
3481 022046  016137  000042  001252              MOV     42(R1),@#$TMP10
3482 022054  012737  022404  001246              MOV     #OVDNTT,@#$TMP6
```

```
3483
3484 022062  016100  000040                    MOV     40(R1),R0       ;LOAD THE FPS.
3485 022066  170100                            LDFPS   R0
3486 022070  012737  022112  001236            MOV     #1$,@#$TMP2
3487 022076  012737  022276  000244            MOV     #25$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
3488                                                                    ;OF ERROR.
3489 022104  010100                            MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
3490 022106  062700  000010                    ADD     #10,R0
3491
3492 022112  171010                  1$:       MULD    (R0),AC0        ;TEST INSTRUCTION.
3493
3494 022114  170204                  2$:       STFPS   R4              ;GET FPS.
3495 022116  170305                            STST    R5              ;GET FEC.
3496 022120  012700  000200                    MOV     #200,R0         ;SET FD MODE.
3497 022124  170100                            LDFPS   R0
3498 022126  012700  022404                    MOV     #OVDNTT,R0      ;GET THE RESULT.
3499 022132  174010                            STD     AC0,(R0)
3500 022134  010437  001250                    MOV     R4,@#$TMP7
3501 022140  010537  001254                    MOV     R5,@#$TMP11
3502
3503 022144  012700  022404                    MOV     #OVDNTT,R0      ;CHECK THE RESULT.
3504 022150  010102                            MOV     R1,R2
3505 022152  062702  000020                    ADD     #20,R2
3506 022156  012703  000004                    MOV     #4,R3
3507 022162  022022                  3$:       CMP     (R0)+,(R2)+
3508 022164  001015                            BNE     15$             ;BRANCH IF INCORRECT.
3509 022166  077303                            SOB     R3,3$
3510
3511 022170  026104  000042                    CMP     42(R1),R4       ;WAS FPS CORRECT?
3512 022174  001002                            BNE     10$             ;BRANCH IF FPS IS INCORRECT.
3513
3514 022176  000161  000056            4$:     JMP     56(R1)          ;RETURN, TEST COMPLETED.
3515
3516                                    ;REPORT INCORRECT FPS.
3517 022202  005761  000046            10$:    TST     46(R1)          ;WAS THE RESULT OVER OR UNDER FLOW?
3518 022206  001002                            BNE     12$             ;BRANCH IF UNDERFLOW.
3519
3520                                                                    ;REPORT FPS BAD AFTER OVERFLOW.
3521 022210  104123                  11$:      ERROR   +123
3522 022212  000771                            BR      4$
3523
3524 022214                           12$:                              ;REPORT FPS BAD AFTER UNDERFLOW.
3525 022214  104124                  13$:      ERROR   +124
3526 022216  000767                            BR      4$
3527
3528                                    ;RESULT INCORRECT.
3529 022220  012700  022404            15$:    MOV     #OVDNTT,R0      ;SEE IF FAILURE IS ANTICIPATED
3530 022224  010102                            MOV     R1,R2           ;FAILURE.
3531 022226  062702  000030                    ADD     #30,R2
3532 022232  012703  000004                    MOV     #4,R3
3533 022236  022022                  16$:      CMP     (R0)+,(R2)+
3534 022240  001007                            BNE     17$             ;BRANCH IF NOT ANTICIPATED.
3535 022242  077303                            SOB     R3,16$
3536
3537 022244  010102                            MOV     R1,R2           ;ERROR WAS ANTICIPATED SO RETURN
3538 022246  062702  000054                    ADD     #54,R2          ;TO THE ERROR REPORT IN THE CALLING
3539 022252  010237  001236                    MOV     R2,@#$TMP2      ;ROUTINE.
```

```
3540 022256  000112                          JMP     (R2)
3541
3542 022260  005761  000046          17$:    TST     46(R1)          ;RESULT WAS NOT ANTICIPATED
3543                                                                 ;SO ERROR MUST BE REPORTED HERE.
3544                                                                 ;FIRST SEE IF ARGUMENTS SHOULD
3545                                                                 ;HAVE RESULTED IN OVERFLOW OR UNDER
3546                                                                 ;FLOW BY LOOKING AT THE FLAG.
3547 022264  001002                          BNE     19$             ;BRANCH IF UNDERFLOW EXPECTED.
3548
3549                                                                 ;REPORT RESULT INCORRECT, EXPECTING
3550 022266  104125          18$:    ERROR   +125                    ;OVERFLOW.
3551 022270  000742                          BR      4$
3552
3553 022272                  19$:                                    ;REPORT RESULT INCORRECT, EXPECTING
3554 022272  104126          20$:    ERROR   +126                    ;UNDERFLOW.
3555 022274  000740                          BR      4$
3556
3557                         ;IF AN FP TRAP OCCURS COME HERE.
3558 022276  011602          25$:    MOV     (SP),R2                 ;GET ADDRESS OF TRAP.
3559 022300  022702  022114            CMP     #2$,R2                  ;WAS THE TRAP DURING THE MULF INSTRUCTION?
3560 022304  001402                    BEQ     26$                     ;BRANCH IF YES.
3561 022306  000137  036660            JMP     @#FPSPUR                ;OTHERWISE GO REPORT A SPURIOUS
3562                                                                 ;FP TRAP.
3563 022312  022626          26$:    CMP     (SP)+,(SP)+             ;RESET THE STACK.
3564 022314  010237  001236            MOV     R2,@#$TMP2              ;SAVE DATA FOR ERROR REPORT.
3565 022320  170204                    STFPS   R4                      ;GET FPS.
3566 022322  170305                    STST    R5                      ;GET FEC.
3567 022324  012700  000200            MOV     #200,R0                 ;SET FD MODE.
3568 022330  170100                    LDFPS   R0
3569 022332  012700  022404            MOV     #OVDNTT,R0              ;GET THE RESULT.
3570 022336  174010                    STD     AC0,(R0)
3571 022340  010537  001254            MOV     R5,@#$TMP11
3572 022344  020561  000044            CMP     R5,44(R1)               ;WAS THE FEC ANTICIPATED?
3573 022350  001004                    BNE     27$                     ;BRANCH IF NOT ANTICIPATED.
3574
3575 022352  010102                    MOV     R1,R2                   ;ERROR WAS ANTICIPATED SO
3576 022354  062702  000050            ADD     #50,R2                  ;RETURN TO THE ERROR REPORT OF THE
3577                                                                 ;CALLING ROUTINE.
3578 022360  000112                    JMP     (R2)
3579
3580 022362  005761  000026    27$:    TST     26(R1)                  ;THE ERROR WAS NOT ANTICIPATED SO
3581                                                                 ;IT MUST BE REPORTED HERE. FIRST SEE IF EXPECTED
3582                                                                 ;OVERFLOW OR UNDER FLOW.
3583 022366  001003                    BNE     29$                     ;BRANCH IF EXPECTING UNDERFLOW
3584
3585                                                                 ;REPORT TRAPPED ON OVERFLOW WITH FIV=0
3586 022370  104127          28$:    ERROR   +127
3587 022372  000161  000056            JMP     56(R1)
3588
3589 022376                  29$:                                    ;REPORT TRAPPED ON UNDER FLOW WITH FIU=0
3590 022376  104130          30$:    ERROR   +130
3591 022400  000161  000056            JMP     56(R1)
3592
3593 022404  000000  000000  000000  OVDNTT: .WORD   0,0,0,0
     022412  000000
3594
3595 022414                  JJJDONE:
```

```
        022414  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?).
   3596
   3597
   3598
   3610
   3611                                         ;TEST TITLE:UNDER/OVERFLOW, USING MULF WITH TRAPS ENABLED
   3612                                 ;;*****************************************************************
                                        ;*TEST 14       SEE COMMENT ABOVE FOR TEST TITLE
                                        ;*
                                        ;*THIS IS A TEST OF THE UNDERFLOW AND OVERFLOW
                                        ;*CONDITIONS THAT CAN OCCUR USING THE MULF INSTRUCTION.
                                        ;*A SUBROUTINE IS CALLED TO SET UP THE OPERANDS,
                                        ;* EXECUTE THE MULF INSTRUCTION AND CHECK
                                        ;*THE RESULTS. HERE THE PARTICULAR INTERRUPT,
                                        ;*EITHER OVERFLOW OR UNDERFLOW, IS ENABLED SO A TRAP SHOULD
                                        ;*OCCUR.
                                        ;*
                                        ;*****************************************************************
        022416  000004                  TST14:  SCOPE
   3613
   3614                                 ;UNDERFLOW, EXPONENT OF RESULT = -129
   3615 022420                          KKK1:
        022420  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
   3616 022422  004737  022644                  JSR     PC,@#OVUNFT
   3617 022426  020123  045676          1$:     .WORD   20123,45676     ;AC
   3618 022432  020200  000000          2$:     .WORD   20200,0         ;FSRC
   3619 022436  000123  045676          3$:     .WORD   123,45676       ;RES
   3620 022442  177777  177777          4$:     .WORD   -1,-1           ;ERROR RES.
   3621 022446  002000                  5$:     2000                    ;FPS BEFORE EXECUTION.
   3622 022450  102004                          102004                  ;FPS AFTER EXECUTION.
   3623 022452  000012                  6$:     12                      ;FEC
   3624 022454  177777                          -1                      ;FLAG
   3625 022456  104145                  7$:     ERROR   +145            ;ST 331 (BUT FIU) NO TRAP.
   3626 022460  000401                          BR      8$
   3627 022462  104144                          ERROR   +144
   3628 022464                          8$:
   3629
   3630                                 ;UNDERFLOW, EXPONENT OF THE RESULT = -193
   3631 022464                          KKK3:
        022464  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
   3632 022466  004737  022644                  JSR     PC,@#OVUNFT
   3633 022472  010127  127272          1$:     .WORD   10127,127272    ;AC
   3634 022476  010200  000000          2$:     .WORD   10200,0         ;FSRC
   3635 022502  060127  127272          3$:     .WORD   60127,127272    ;RES
   3636 022506  177777  177777          4$:     .WORD   -1,-1           ;ERROR RES.
   3637 022512  007017                  5$:     7017                    ;FPS BEFORE EXECUTION.
   3638 022514  107000                          107000                  ;FPS AFTER EXECUTION.
   3639 022516  000012                  6$:     12                      ;FEC
   3640 022520  177777                          -1
   3641 022522  104146                  7$:     ERROR   +146            ;ST 137 (BUT FIU) NO TRAP.
   3642 022524  000401                          BR      8$
   3643 022526  104144                          ERROR   +144
   3644 022530                          8$:
```

```
                                        ;OVERFLOW, EXPONENT OF THE RESULT = 128
022530                                  KKK4:
022530  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
022532  004737  022644                          JSR     PC,@#OVUNFT
022536  060252  125252                  1$:     .WORD   60252,125252    ;AC
022542  060000  000000                  2$:     .WORD   60000,0         ;FSRC
022546  000052  125252                  3$:     .WORD   000052,125252   ;RES
022552  177777  177777                  4$:     .WORD   -1,-1           ;ERROR RES.
022556  001000                          5$:     1000                    ;FPS BEFORE EXECUTION.
022560  101006                                  101006                  ;FPS AFTER EXECUTION.
022562  000010                          6$:     10                      ;FEC
022564  000000                                  0                       ;FLAG
022566  104147                          7$:     ERROR   +147            ;ST 333 (BUT FIV) NO TRAP
022570  000401                                  BR      8$
022572  104143                                  ERROR   +143
022574                                  8$:

                                        ;OVERFLOW, EXPONENT OF RESULT = 130
022574                                  KKK5:
022574  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
022576  004737  022644                          JSR     PC,@#OVUNFT
022602  060345  067654                  1$:     .WORD   60345,67654     ;AC
022606  060200  000000                  2$:     .WORD   60200,0         ;FSRC
022612  000345  067654                  3$:     .WORD   345,67654       ;RES
022616  177777  177777                  4$:     .WORD   -1,-1           ;ERROR RES.
022622  007015                          5$:     7015                    ;FPS BEFORE EXECUTION.
022624  107002                                  107002                  ;FPS AFTER EXECUTION.
022626  000010                          6$:     10                      ;FEC
022630  000000                                  0                       ;FLAG
022632  104150                          7$:     ERROR   +150            ;ST 133 (BUT FIV) NO TRAP
022634  000401                                  BR      8$
022636  104143                                  ERROR   +143
022640  000167  000412                  8$:     JMP     KKKDONE

                                        ;THIS SUBROUTINE, OVUNFT, IS USED TO SET UP THE OPERANDS, EXECUTE
                                        ;THE MULF INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
                                        ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
                                        ;TO IT IS MADE THUS:
                                        ;
                                        ;               ACARG:  .WORD   X,X             ;AC OPERAND
                                        ;               FSRCARG:.WORD   X,X             ;FSRC 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
                                        ;               FEC:    .WORD   X               ;EXPECTED FEC
                                        ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
                                        ;               ERR1:   ERROR   +X              ;TRAP ERROR.
                                        ;               BR      CONT
                                        ;               ERR2:   ERROR   +X              ;DATA, RESULT ERROR
                                        ;               CONT:                           ;RETURN ADDRESS
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                        ;THE MULF INSTRUCTION IS EXECUTED. IF THE TRAP OCCURS THEN THE
                                        ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                        ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNFT RETURNS CONTROL
```

```
                                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNFT
                                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. THE FEC IS TREATED
                                        ;IN THE SAME WAY. IF THE RESULT OF THE
                                        ;MULF 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 OVUNFT
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNFT WILL
                                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
                                        ;IF NO TRAP OCCURS CONTROL IS PASSED TO ERR1.
                                        ;NOTE THAT OVUNFNT USES THE FLAG
                                        ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
                                        ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).

022644  012601                  OVUNFT: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
022646  012700  000200                  MOV     #200,R0         ;SET FD MODE.
022652  170100                          LDFPS   R0

022654  010100                          MOV     R1,R0           ;LOAD AC0, OPERAND.
022656  172410                          LDD     (R0),AC0

022660  010102                          MOV     R1,R2           ;SAVE THE DATA PATTERNS IN CASE OF
022662  010237  001240                  MOV     R2,@#$TMP3      ;ERROR.
022666  062702  000004                  ADD     #4,R2
022672  010237  001242                  MOV     R2,@#$TMP4
022676  062702  000004                  ADD     #4,R2
022702  010237  001244                  MOV     R2,@#$TMP5
022706  016137  000022  001252          MOV     22(R1),@#$TMP10
022714  012737  023246  001246          MOV     #OVFTT,@#$TMP6

022722  016100  000020                  MOV     20(R1),R0       ;LOAD THE FPS.
022726  170100                          LDFPS   R0
022730  012737  022752  001236          MOV     #1$,@#$TMP2
022736  012737  022762  000244          MOV     #50$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
                                                                ;OF ERROR.
022744  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
022746  062700  000004                  ADD     #4,R0

022752  171010                  1$:     MULF    (R0),AC0        ;TEST INSTRUCTION. SHOULD CAUSE TRAP.
022754  170000                  2$:     CFCC

022756  000137  023206                  JMP     @#25$           ;FAILURE, NO TRAP.

022762  011602                  50$:    MOV     (SP),R2         ;TRAP TO HERE AND SEE IF THE PC OF THE
022764  020227  022754                  CMP     R2,#2$          ;TRAP WAS THAT OF THE MULF INSTRUCTION.
022770  001402                          BEQ     51$             ;BRANCH IF YES.
022772  000137  036660                  JMP     @#FPSPUR        ;OTHERWISE REPORT SPURIOUS FP ERROR.

022776  022626                  51$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
023000  170204                          STFPS   R4              ;GET FPS.
023002  170305                          STST    R5              ;GET FEC.
023004  012700  000200                  MOV     #200,R0         ;SET FD MODE.
023010  170100                          LDFPS   R0
023012  012700  023246                  MOV     #OVFTT,R0       ;GET THE RESULT.
023016  174010                          STD     AC0,(R0)
023020  010437  001250                  MOV     R4,@#$TMP7
023024  010537  001254                  MOV     R5,@#$TMP11
```

```
3757
3758 023030  012700  023246            MOV     #OVFTT,R0       ;CHECK THE RESULT.
3759 023034  010102                    MOV     R1,R2
3760 023036  062702  000010            ADD     #10,R2
3761 023042  012703  000002            MOV     #2,R3
3762 023046  022022              3$:   CMP     (R0)+,(R2)+
3763 023050  001027                    BNE     15$             ;BRANCH IF INCORRECT.
3764 023052  077303                    SOB     R3,3$
3765
3766 023054  026104  000022            CMP     22(R1),R4       ;WAS FPS CORRECT?
3767 023060  001014                    BNE     10$             ;BRANCH IF FPS IS INCORRECT.
3768
3769 023062  026105  000024            CMP     24(R1),R5       ;IS FEC CORRECT?
3770 023066  001002                    BNE     5$              ;IF INCORRECT BRANCH.
3771 023070  000161  000036      4$:   JMP     36(R1)          ;RETURN, TEST COMPLETED.
3772
3773                           ;REPORT INCORRECT FEC.
3774 023074  005761  000026      5$:   TST     26(R1)          ;WAS THE RESULT OVERFLOW OR UNDERFLOW?
3775 023100  001002                    BNE     7$              ;BRANCH IF UNDERFLOW.
3776
3777                                                           ;REPORT BAD FEC ON EXPECTED OVERFLOW.
3778 023102  104137              6$:   ERROR   +137
3779 023104  000771                    BR      4$
3780
3781 023106                      7$:                           ;REPORT BAD FEC ON EXPECTED UNDERFLOW.
3782 023106  104140              8$:   ERROR   +140
3783 023110  000767                    BR      4$
3784
3785                           ;REPORT INCORRECT FPS.
3786 023112  005761  000026      10$:  TST     26(R1)          ;WAS THE RESULT OVER OR UNDER FLOW?
3787 023116  001002                    BNE     12$             ;BRANCH IF UNDERFLOW.
3788
3789                                                           ;REPORT FPS BAD AFTER OVERFLOW.
3790 023120  104141              11$:  ERROR   +141
3791 023122  000762                    BR      4$
3792
3793 023124                      12$:                          ;REPORT FPS BAD AFTER UNDERFLOW.
3794 023124  104142              13$:  ERROR   +142
3795 023126  000760                    BR      4$
3796
3797                           ;RESULT INCORRECT.
3798 023130  012700  023246      15$:  MOV     #OVFTT,R0       ;SEE IF FAILURE IS ANTICIPATED
3799 023134  010102                    MOV     R1,R2           ;FAILURE.
3800 023136  062702  000014            ADD     #14,R2
3801 023142  012703  000002            MOV     #2,R3
3802 023146  022022              16$:  CMP     (R0)+,(R2)+
3803 023150  001007                    BNE     17$             ;BRANCH IF NOT ANTICIPATED.
3804 023152  077303                    SOB     R3,16$
3805
3806 023154  010102                    MOV     R1,R2           ;ERROR WAS ANTICIPATED SO RETURN
3807 023156  062702  000034            ADD     #34,R2          ;TO THE ERROR REPORT IN THE CALLING
3808 023162  010237  001236            MOV     R2,@#$TMP2      ;ROUTINE.
3809 023166  000112                    JMP     (R2)
3810
3811 023170  005761  000026      17$:  TST     26(R1)          ;RESULT WAS NOT ANTICIPATED
3812                                                           ;SO ERROR MUST BE REPORTED HERE.
3813                                                           ;FIRST SEE IF ARGUMENTS SHOULD
```

```
3814                                                    ;HAVE RESULTED IN OVERFLOW OR UNDER
3815                                                    ;FLOW BY LOOKING AT THE FLAG.
3816 023174  001002                    BNE     19$      ;BRANCH IF UNDERFLOW EXPECTED.
3817
3818                                                    ;REPORT RESULT INCORRECT, EXPECTING
3819 023176  104143            18$:    ERROR   +143     ;OVERFLOW.
3820 023200  000733                    BR      4$
3821
3822 023202                    19$:                     ;REPORT RESULT INCORRECT, EXPECTING
3823 023202  104144            20$:    ERROR   +144     ;UNDERFLOW.
3824 023204  000731                    BR      4$
3825
3826                           ;IF NO FP TRAP OCCURS COME HERE.
3827 023206  170204            25$:    STFPS   R4       ;GET FPS.
3828 023210  170305                    STST    R5       ;GET FEC.
3829 023212  012700  000200            MOV     #200,R0  ;SET FD MODE.
3830 023216  170100                    LDFPS   R0
3831 023220  012700  023246            MOV     #OVFTT,R0 ;GET THE RESULT.
3832 023224  174010                    STD     AC0,(R0)
3833 023226  010437  001250            MOV     R4,@#$TMP7
3834 023232  010537  001254            MOV     R5,@#$TMP11
3835 023236  010102                    MOV     R1,R2    ;ERROR WAS ANTICIPATED SO
3836 023240  062702  000030            ADD     #30,R2   ;RETURN TO THE ERROR REPORT OF THE
3837                                                    ;CALLING ROUTINE.
3838 023244  000112                    JMP     (R2)
3839
3840 023246  000000  000000  000000  OVFTT:  .WORD   0,0,0,0
     023254  000000
3841
3842 023256                    KKKDONE:
     023256  104412                    RSETUP           ;GO INITIALIZE THE FPS AND STACK; AND
                                                        ;SEE IF THE USER HAS EXPRESSED
                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                        ;THE USER TYPED CONTROL G?).
3843
3844
3845
3853
3854                                    ;TEST TITLE:UNDER/OVERFLOW, USING MULD WITH TRAPS ENABLED
3855                           ;;*****************************************************************
                               ;*TEST 15       SEE COMMENT ABOVE FOR TEST TITLE
                               ;*
                               ;*THIS IS A TEST OF THE OVER FLOW AND UNDER FLOW CONDITIONS USING THE
                               ;*MULD INSTRUCTION WITH TRAPS ENABLED. A SUBROUTINE IS USED TO SET UP
                               ;*THE OPERANDS, EXECUTE THE MULD INSTRUCTION AND CHECK THE RESULTS.
                               ;*
                               ;;*****************************************************************
     023260  000004            TST15:  SCOPE
3856
3857                           ;UNDERFLOW, EXPONENT OF RESULT = -129
3858 023262                    LLL1:
     023262  104413                    LPERR            ;SET UP THE LOOP ON ERROR ADDRESS.
3859 023264  004737  023606            JSR     PC,@#OVUNDT
3860 023270  020052  125252    1$:     .WORD   20052,125252    ;AC
3861 023274  125252  125252            .WORD   125252,125252
3862 023300  020300  000000  000000  2$:     .WORD   20300,0,0,0     ;FSRC
```

```
        023306  000000
3863    023310  000177  177777  177777  3$:     .WORD   177,-1,-1,-1      ;RES
        023316  177777
3864    023320  000177  177777          4$:     .WORD   177,-1            ;ERROR RES.
3865    023324  125252  125252                  .WORD   125252,125252
3866    023330  002200                  5$:     2200                      ;FPS BEFORE EXECUTION.
3867    023332  102204                          102204                    ;FPS AFTER EXECUTION.
3868    023334  000012                  6$:     12                        ;FEC
3869    023336  177777                          -1                        ;FLAG
3870    023340  104157                  7$:     ERROR   +157              ;ST 331 (BUT FIU) NO TRAP.
3871    023342  000401                          BR      8$
3872    023344  104160                          ERROR   +160              ;ST 155 (BUT FD)
3873    023346                          8$:
3874
3875                                    ;UNDERFLOW, EXPONENT OF THE RESULT = -193
3876    023346                          LLL2:
        023346  104413                          LPERR                     ;SET UP THE LOOP ON ERROR ADDRESS.
3877    023350  004737  023606                  JSR     PC,@#OVUNDT
3878    023354  010327  127272          1$:     .WORD   10327,127272      ;AC
3879    023360  036363  045454                  .WORD   36363,45454
3880    023364  010000  000000  000000  2$:     .WORD   10000,0,0,0       ;FSRC
        023372  000000
3881    023374  060127  127272          3$:     .WORD   60127,127272      ;RES
3882    023400  036363  045454                  .WORD   36363,45454
3883    023404  177777  177777  177777  4$:     .WORD   -1,-1,-1,-1       ;ERROR RES.
        023412  177777
3884    023414  007217                  5$:     7217                      ;FPS BEFORE EXECUTION.
3885    023416  107200                          107200                    ;FPS AFTER EXECUTION.
3886    023420  000012                  6$:     12                        ;FEC
3887    023422  177777                          -1                        ;FLAG
3888    023424  104161                  7$:     ERROR   +161              ;ST 137 (BUT FIU) NO TRAP.
3889    023426  000401                          BR      8$
3890    023430  104156                          ERROR   +156
3891    023432                          8$:
3892
3893                                    ;OVERFLOW, EXPONENT OF THE RESULT = 128
3894    023432                          LLL3:
        023432  104413                          LPERR                     ;SET UP THE LOOP ON ERROR ADDRESS.
3895    023434  004737  023606                  JSR     PC,@#OVUNDT
3896    023440  060252  125252          1$:     .WORD   60252,125252      ;AC
3897    023444  125252  125252                  .WORD   125252,125252     ;FSRC
3898    023450  160100  000000  000000  2$:     .WORD   160100,0,0,0      ;FSRC
        023456  000000
3899    023460  100177  177777  177777  3$:     .WORD   100177,-1,-1,-1   ;RES
        023466  177777
3900    023470  100177  177777          4$:     .WORD   100177,-1         ;ERROR RES.
3901    023474  125252  125252                  .WORD   125252,125252
3902    023500  001200                  5$:     1200                      ;FPS BEFORE EXECUTION.
3903    023502  101216                          101216                    ;FPS AFTER EXECUTION.
3904    023504  000010                  6$:     10                        ;FEC
3905    023506  000000                          0                         ;FLAG
3906    023510  104162                  7$:     ERROR   +162              ;ST 333 (BUT FIV) NO TRAP.
3907    023512  000401                          BR      8$
3908    023514  104163                          ERROR   +163              ;ST 700 (BUT FD).
3909    023516                          8$:
3910
3911                                    ;OVERFLOW, EXPONENT OF THE RESULT = 130
```

```
 3912 023516                          LLL4:
      023516  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 3913 023520  004737  023606                 JSR     PC,@#OVUNDT
 3914 023524  060345  067654          1$:    .WORD   60345,67654     ;AC
 3915 023530  056765  045676                 .WORD   56765,45676
 3916 023534  060200  000000  000000 2$:    .WORD   60200,0,0,0     ;FSRC
      023542  000000
 3917 023544  000345  067654          3$:    .WORD   345,67654       ;RES
 3918 023550  056765  045676                 .WORD   56765,45676
 3919 023554  177777  177777  177777 4$:    .WORD   -1,-1,-1,-1     ;ERROR RES.
      023562  177777
 3920 023564  007215                  5$:    7215                    ;FPS BEFORE EXECUTION.
 3921 023566  107202                         107202                  ;FPS AFTER EXECUTION.
 3922 023570  000010                  6$:    10                      ;FEC
 3923 023572  000000                         0                       ;FLAG
 3924 023574  104164                  7$:    ERROR   +164            ;ST 133 (BUT FIV) NO TRAP
 3925 023576  000401                         BR      8$
 3926 023600  104155                         ERROR   +155
 3927 023602  000137  024220          8$:    JMP     @#LLLDONE
 3928
 3929                                 ;THIS SUBROUTINE, OVUNDT, IS USED TO SET UP THE OPERANDS, EXECUTE
 3930                                 ;THE MULD INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
 3931                                 ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
 3932                                 ;TO IT IS MADE THUS:
 3933                                 ;
 3934                                 ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
```

```
                                        FSRCARG:.WORD   X,X,X,X         ;FSRC 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
                                        FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
                                        ERR1:   ERROR   +X              ;TRAP ERROR.
                                        BR      CONT
                                        ERR2:   ERROR   +X              ;DATA, RESULT ERROR
                                        CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                ;THE MULD INSTRUCTION IS EXECUTED. IF THE TRAP OCCURS THEN THE
                                ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNDT RETURNS CONTROL
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNDT
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. THE FEC IS TREATED
                                ;IN THE SAME WAY. IF THE RESULT OF THE
                                ;MULF 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 OVUNDT
                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNDT WILL
                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
                                ;IF NO TRAP OCCURS CONTROL IS PASSED TO ERR1.
                                ;NOTE THAT OVUNDNT USES THE FLAG
                                ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
                                ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).

023606  012601                  OVUNDT: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
023610  012700  000200                  MOV     #200,R0         ;SET FD MODE.
023614  170100                          LDFPS   R0

023616  010100                          MOV     R1,R0           ;LOAD AC0, OPERAND.
023620  172410                          LDD     (R0),AC0

023622  010102                          MOV     R1,R2           ;SAVE THE DATA PATTERNS IN CASE OF
023624  010237  001240                  MOV     R2,@#$TMP3      ;ERROR.
023630  062702  000010                  ADD     #10,R2
023634  010237  001242                  MOV     R2,@#$TMP4
023640  062702  000010                  ADD     #10,R2
023644  010237  001244                  MOV     R2,@#$TMP5
023650  016137  000042  001252          MOV     42(R1),@#$TMP10
023656  012737  024210  001246          MOV     #OVDTT,@#$TMP6

023664  016100  000040                  MOV     40(R1),R0       ;LOAD THE FPS.
023670  170100                          LDFPS   R0
023672  012737  023714  001236          MOV     #1$,@#$TMP2
023700  012737  023724  000244          MOV     #50$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
                                                                ;OF ERROR.
023706  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
023710  062700  000010                  ADD     #10,R0

023714  171010                  1$:     MULD    (R0),AC0        ;TEST INSTRUCTION. SHOULD CAUSE TRAP.
023716  170000                  2$:     CFCC
```

```
 3993 023720  000137  024150                    JMP     @#25$           ;FAILURE, NO TRAP.
 3994
 3995 023724  011602                    50$:    MOV     (SP),R2         ;TRAP TO HERE AND SEE IF THE PC OF THE
 3996 023726  020227  023716                    CMP     R2,#2$          ;TRAP WAS THAT OF THE MULF INSTRUCTION.
 3997 023732  001402                            BEQ     51$             ;BRANCH IF YES.
 3998 023734  000137  036660                    JMP     @#FPSPUR        ;OTHERWISE REPORT SPURIOUS FP ERROR.
 3999
 4000 023740  022626                    51$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
 4001 023742  170204                            STFPS   R4              ;GET FPS.
 4002 023744  170305                            STST    R5              ;GET FEC.
 4003 023746  012700  000.00                    MOV     #200,R0         ;SET FD MODE.
 4004 023752  170100                            LDFPS   R0
 4005 023754  012700  024210                    MOV     #OVDTT,R0       ;GET THE RESULT.
 4006 023760  174010                            STD     AC0,(R0)
 4007 023762  010437  001250                    MOV     R4,@#$TMP7
 4008 023766  010537  001254                    MOV     R5,@#$TMP11
 4009
 4010 023772  012700  024210                    MOV     #OVDTT,R0       ;CHECK THE RESULT.
 4011 023776  010102                            MOV     R1,R2
 4012 024000  062702  000020                    ADD     #20,R2
 4013 024004  012703  000004                    MOV     #4,R3
 4014 024010  022022                    3$:     CMP     (R0)+,(R2)+
 4015 024012  001027                            BNE     15$             ;BRANCH IF INCORRECT.
 4016 024014  077303                            SOB     R3,3$
 4017
 4018 024016  026104  000042                    CMP     42(R1),R4       ;WAS FPS CORRECT?
 4019 024022  001014                            BNE     10$             ;BRANCH IF FPS IS INCORRECT.
 4020
 4021 024024  026105  000044                    CMP     44(R1),R5       ;IS FEC CORRECT?
 4022 024030  001002                            BNE     5$              ;IF INCORRECT BRANCH.
 4023 024032  000161  000056            4$:     JMP     56(R1)          ;RETURN, TEST COMPLETED.
 4024
 4025                                    ;REPORT INCORRECT FEC.
 4026 024036  005761  000046            5$:     TST     46(R1)          ;WAS THE RESULT OVERFLOW OR UNDERFLOW?
 4027 024042  001002                            BNE     7$              ;BRANCH IF UNDERFLOW.
 4028
 4029                                                                    ;REPORT BAD FEC ON EXPECTED OVERFLOW.
 4030 024044  104151                    6$:     ERROR   +151
 4031 024046  000771                            BR      4$
 4032
 4033 024050                            7$:                             ;REPORT BAD FEC ON EXPECTED UNDERFLOW.
 4034 024050  104152                    8$:     ERROR   +152
 4035 024052  000767                            BR      4$
 4036
 4037                                    ;REPORT INCORRECT FPS.
 4038 024054  005761  000046            10$:    TST     46(R1)          ;WAS THE RESULT OVER OR UNDER FLOW?
 4039 024060  001002                            BNE     12$             ;BRANCH IF UNDERFLOW.
 4040
 4041                                                                    ;REPORT FPS BAD AFTER OVERFLOW.
 4042 024062  104153                    11$:    ERROR   +153
 4043 024064  000762                            BR      4$
 4044
 4045 024066                            12$:                            ;REPORT FPS BAD AFTER UNDERFLOW.
 4046 024066  104154                    13$:    ERROR   +154
 4047 024070  000760                            BR      4$
 4048
 4049                                    ;RESULT INCORRECT.
```

```
4050 024072  012700  024210          15$:   MOV    #OVDTT,R0        ;SEE IF FAILURE IS ANTICIPATED
4051 024076  010102                         MOV    R1,R2            ;FAILURE.
4052 024100  062702  000030                 ADD    #30,R2
4053 024104  012703  000004                 MOV    #4,R3
4054 024110  022022                 16$:   CMP    (R0)+,(R2)+
4055 024112  001007                         BNE    17$              ;BRANCH IF NOT ANTICIPATED.
4056 024114  077303                         SOB    R3,16$
4057
4058 024116  010102                         MOV    R1,R2            ;ERROR WAS ANTICIPATED SO RETURN
4059 024120  062702  000054                 ADD    #54,R2           ;TO THE ERROR REPORT IN THE CALLING
4060 024124  010237  001236                 MOV    R2,@#$TMP2       ;ROUTINE.
4061 024130  000112                         JMP    (R2)
4062
4063 024132  005761  000046          17$:   TST    46(R1)           ;RESULT WAS NOT ANTICIPATED
4064                                                                 ;SO ERROR MUST BE REPORTED HERE.
4065                                                                 ;FIRST SEE IF ARGUMENTS SHOULD
4066                                                                 ;HAVE RESULTED IN OVERFLOW OR UNDER
4067                                                                 ;FLOW BY LOOKING AT THE FLAG.
4068 024136  001002                         BNE    19$              ;BRANCH IF UNDERFLOW EXPECTED.
4069
4070                                                                 ;REPORT RESULT INCORRECT, EXPECTING
4071 024140  104155                 18$:   ERROR  +155              ;OVERFLOW.
4072 024142  000733                         BR     4$
4073
4074 024144                         19$:                            ;REPORT RESULT INCORRECT, EXPECTING
4075 024144  104156                 20$:   ERROR  +156              ;UNDERFLOW.
4076 024146  000731                         BR     4$
4077
4078                                ;IF NO FP TRAP OCCURS COME HERE.
4079 024150  170204                 25$:   STFPS  R4                ;GET FPS.
4080 024152  170305                        STST   R5                ;GET FEC.
4081 024154  012700  000200                MOV    #200,R0          ;SET FD MODE.
4082 024160  170100                        LDFPS  R0
4083 024162  012700  024210                MOV    #OVDTT,R0        ;GET THE RESULT.
4084 024166  174010                        STD    AC0,(R0)
4085 024170  010437  001250                MOV    R4,@#$TMP7
4086 024174  010537  001254                MOV    R5,@#$TMP11
4087 024200  010102                        MOV    R1,R2            ;ERROR WAS ANTICIPATED SO
4088 024202  062702  000050                ADD    #50,R2           ;RETURN TO THE ERROR REPORT OF THE
4089                                                                 ;CALLING ROUTINE.
4090 024206  000112                        JMP    (R2)
4091
4092 024210  000000  000000  000000  OVDTT: .WORD  0,0,0,0
     024216  000000
4093
4094 024220                         LLLDONE:
     024220  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?).
4095
4096
4097
4098
4106
4107                                ;;*****************************************************************
```

```
                                ;*TEST 16       MODF TEST
                                ;*
                                ;*THIS IS A TEST OF THE MODF INSTRUCTION, WHICH MAKES USE OF
                                ;*A SUBROUTINE TO SET UP THE OPERANDS, EXECUTE THE MODF INSTRUCTION
                                ;*AND CHECK THE RESULTS.
                                ;*
                                ;;*****************************************************************
     024222  000004             TST16:  SCOPE
 4108
 4109                           ;MODF WITH (FSRC=AC=0)
 4110 024224                    GGG1:
      024224  104413                    LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4111 024226  004737  025310             JSR     PC,@#MODFSUB
 4112 024232  000000  000000    1$:      .WORD   0,0             ;AC
 4113 024236  000000  000000    2$:      .WORD   0,0             ;FSRC
 4114 024242  000000  000000    3$:      .WORD   0,0             ;FRACTIONAL RES.
 4115 024246  000000  000000    4$:      .WORD   0,0             ;INTEGER RES.
 4116 024252  177777  177777    5$:      .WORD   -1,-1           ;ERROR FRACTIONAL RES.
 4117 024256  177777  177777    6$:      .WORD   -1,-1           ;ERROR INGETER RES.
 4118 024262  000013             7$:      13                      ;FPS BEFORE EXECUTION.
 4119 024264  000004                      4                       ;FPS AFTER EXECUTION.
 4120 024266  104056             8$:      ERROR   +56             ;STORE SINGLE ZERO BAD.
 4121 024270  000401                      BR      9$
 4122 024272  104057                      ERROR   +57             ;AC V 1 <= ZERO FAILED.
 4123 024274                    9$:
 4124
 4125                           ;MODF TEST, WITH (FSRC=0)
 4126 024274                    GGG2:
      024274  104413                    LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4127 024276  004737  025310             JSR     PC,@#MODFSUB
 4128 024302  123456  076543    1$:      .WORD   123456,76543    ;AC
 4129 024306  000000  000000    2$:      .WORD   0,0             ;FSRC
 4130 024312  000000  000000    3$:      .WORD   0,0             ;FRACTIONAL RES.
 4131 024316  000000  000000    4$:      .WORD   0,0             ;INTEGER RESULT.
 4132 024322  123456  076543    5$:      .WORD   123456,76543    ;ERROR FRACTIONAL RES.
 4133 024326  177777  177777    6$:      .WORD   -1,-1           ;ERROR INTEGER RES.
 4134 024332  000000             7$:      0                       ;FPS BEFORE EXECUTION.
 4135 024334  000004                      4                       ;FPS AFTER EXECUTION.
 4136 024336  104056             8$:      ERROR   +56             ;STORE ZERO FAILURE.
 4137 024340  000401                      BR      9$
 4138 024342  104057                      ERROR   +57
 4139 024344                    9$:
 4140
 4141                           ;MODF TEST WITH (AC=0)
 4142 024344                    GGG3:
      024344  104413                    LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4143 024346  004737  025310             JSR     PC,@#MODFSUB
 4144 024352  000000  000000    1$:      .WORD   0,0             ;AC
 4145 024356  076543  021234    2$:      .WORD   76543,21234     ;FSRC
 4146 024362  000000  000000    3$:      .WORD   0,0             ;FRACTIONAL RES.
 4147 024366  000000  000000    4$:      .WORD   0,0             ;INTEGER RES.
 4148 024372  000000  000000    5$:      .WORD   0,0             ;ERROR FRACTIONAL RES.
 4149 024376  177777  177777    6$:      .WORD   -1,-1           ;ERROR INTEGER RES.
 4150 024402  000003             7$:      3                       ;FPS BEFORE EXECUTION.
 4151 024404  000004                      4                       ;FPS AFTER EXECUTION.
 4152 024406  104053             8$:      ERROR   +53             ;RES.BAD
 4153 024410  000401                      BR      9$
```

```
4154 024412  104057                        ERROR   +57
4155 024414                          9$:
4156
4157                                 ;MODF TEST WITH EXPONENT OF THE RESULT = 25
4158 024414                          GGG4:
     024414  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4159 024416  004737  025310                JSR     PC,@#MODFSUB
4160 024422  046252  125252          1$:   .WORD   46252,125252    ;AC
4161 024426  040300  000000          2$:   .WORD   40300,0         ;FSRC
4162 024432  000000  000000          3$:   .WORD   0,0             ;FRACTIONAL RES.
4163 024436  046377  177777          4$:   .WORD   46377,-1        ;INTEGER RES.
4164 024442  046252  125252          5$:   .WORD   46252,125252    ;ERROR FRACTIONAL RES.
4165 024446  040300  000000          6$:   .WORD   40300,0         ;ERROR INTEGER RES.
4166 024452  000013                  7$:   13                      ;FPS BEFORE EXECUTION.
4167 024454  000004                        4                       ;FPS AFTER EXECUTION.
4168 024456  104053                  8$:   ERROR   +53             ;ST 134
4169 024460  000401                        BR      9$
4170 024462  104060                        ERROR   +60
4171 024464                          9$:
4172
4173                                 ;MODF TEST WITH EXPONENT OF THE RESULT = 127
4174 024464                          GGG5:
     024464  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4175 024466  004737  025310                JSR     PC,@#MODFSUB
4176 024472  077652  125252          1$:   .WORD   77652,125252    ;AC
4177 024476  040300  000000          2$:   .WORD   40300,0         ;FSRC
4178 024502  000000  000000          3$:   .WORD   0,0             ;FRACTIONAL RES.
4179 024506  077777  177777          4$:   .WORD   77777,-1        ;INTEGER RES.
4180 024512  077652  125252          5$:   .WORD   77652,125252    ;ERROR FRACTIONAL RES.
4181 024516  040300  000000          6$:   .WORD   40300,0         ;ERROR INTEGER RES.
4182 024522  000000                  7$:   0                       ;FPS BEFORE EXECUTION.
4183 024524  000004                        4                       ;FPS AFTER EXECUTION.
4184 024526  104053                  8$:   ERROR   +53
4185 024530  000401                        BR      9$
4186 024532  104060                        ERROR   +60
4187 024534                          9$:
4188
4189                                 ;MODF TEST WITH EXPONENT OF RESULT = 25
4190 024534                          GGG6:
     024534  104413                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4191 024536  004737  025310                JSR     PC,@#MODFSUB
4192 024542  046200  000001          1$:   .WORD   46200,1         ;AC
4193 024546  040340  000000          2$:   .WORD   40340,0         ;FSRC
4194 024552  000000  000000          3$:   .WORD   0,0             ;FRACTIONAL RES.
4195 024556  046340  000001          4$:   .WORD   46340,1         ;INTEGER RES.
4196 024562  040000  000000          5$:   .WORD   40000,0         ;ERROR FRACTIONAL RES.
4197 024566  177777  177777          6$:   .WORD   -1,-1           ;ERROR INTEGER RES.
4198 024572  000013                  7$:   13                      ;FPS BEFORE EXECUTION.
4199 024574  000004                        4                       ;FPS AFTER EXECUTION.
4200 024576  104061                  8$:   ERROR   +61             ;BAD CONSTANT (NOT 24),
4201                                                               ;OR ST 525 TO 050 INTO 150.
4202 024600  000401                        BR      9$
4203 024602  104054                        ERROR   +54
4204 024604                          9$:
4205
4206                                 ;MODF   TEST WITH EXPONENT OF THE RESULT = 24
4207 024604                          GGG7:
```

```
                024604  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4208            024606  004737  025310                  JSR     PC,@#MODFSUB
4209            024612  046000  000001          1$:     .WORD   46000,1         ;AC
4210            024616  040340  000000          2$:     .WORD   40340,0         ;FSRC
4211            024622  040100  000000          3$:     .WORD   40100,0         ;FRACTIONAL RES.
4212            024626  046140  000001          4$:     .WORD   46140,1         ;INTEGER RESULT.
4213            024632  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
4214            024636  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
4215            024642  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
4216            024644  000000                          0                       ;FPS AFTER EXECUTION.
4217            024646  104062                  8$:     ERROR   +62             ;BAD CONSTANT USED (NOT 24)
4218                                                                            ;OR ST 525 TO 150 INTO 050
4219            024650  000401                          BR      9$
4220            024652  104054                          ERROR   +54
4221            024654                          9$:
4222
4223                                            ;MODF TEST WITH EXPONENT OF THE RESULT = 10
4224            024654                          GGG8:
                024654  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4225            024656  004737  025310                  JSR     PC,@#MODFSUB
4226            024662  042577  177777          1$:     .WORD   42577,-1        ;AC
4227            024666  040200  000000          2$:     .WORD   40200,0         ;FSRC
4228            024672  040177  176000          3$:     .WORD   40177,176000    ;FRACTIONAL RES.
4229            024676  042577  140000          4$:     .WORD   42577,140000    ;INTEGER RES.
4230            024702  177777  177777          5$:     .WORD   -1,-1           ;ERROR FRACTIONAL RES.
4231            024706  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
4232            024712  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
4233            024714  000000                          0                       ;FPS AFTER EXECUTION.
4234            024716  104053                  8$:     ERROR   +53
4235            024720  000401                          BR      9$
4236            024722  104054                          ERROR   +54
4237            024724                          9$:
4238
4239                                            ;MODF TEST WITH THE EXPONENT OF THE RESULT = 10
4240            024724                          GGG9:
                024724  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4241            024726  004737  025310                  JSR     PC,@#MODFSUB
4242            024732  042577  140001          1$:     .WORD   42577,140001    ;AC
4243            024736  040200  000000          2$:     .WORD   40200,0         ;FSRC
4244            024742  034600  000000          3$:     .WORD   34600,0         ;FRACTIONAL RES.
4245            024746  042577  140000          4$:     .WORD   42577,140000    ;INTEGER RES.
4246            024752  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
4247            024756  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
4248            024762  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
4249            024764  000000                          0                       ;FPS AFTER EXECUTION.
4250            024766  104063                  8$:     ERROR   +63             ;ST 532 TO 122 INTO NORMALIZE.
4251            024770  000401                          BR      9$
4252            024772  104054                          ERROR   +54
4253            024774                          9$:
4254
4255                                            ;MODF TEST WITH EXPONENT OF THE RESULT = 9
4256            024774                          GGG10:
                024774  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4257            024776  004737  025310                  JSR     PC,@#MODFSUB
4258            025002  042377  100000          1$:     .WORD   42377,100000    ;AC
4259            025006  040200  000000          2$:     .WORD   40200,0         ;FSRC
4260            025012  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
```

```
4261 025016  042377  100000          4$:     .WORD   42377,100000     ;INTEGER RES.
4262 025022  177777  177777          5$:     .WORD   -1,-1           ;ERROR FRACTIONAL RES.
4263 025026  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
4264 025032  000013                  7$:     13                      ;FPS BEFORE EXECUTION.
4265 025034  000004                          4                       ;FPS AFTER EXECUTION.
4266 025036  104053                  8$:     ERROR   +53
4267 025040  000401                          BR      9$
4268 025042  104054                          ERROR   +54
4269 025044                          9$:
4270
4271                                 ;MODF TEST WITH EXPONENT OF THE RESULT = 0
4272 025044                          GGG11:
     025044  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4273 025046  004737  025310                  JSR     PC,@#MODFSUB
4274 025052  040177  177777          1$:     .WORD   40177,-1        ;AC
4275 025056  040200  000000          2$:     .WORD   40200,0         ;FSRC
4276 025062  040177  177777          3$:     .WORD   40177,-1        ;FRACTIONAL RES.
4277 025066  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
4278 025072  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
4279 025076  040177  177777          6$:     .WORD   40177,-1        ;ERROR INTEGER RES.
4280 025102  000017                  7$:     17                      ;FPS BEFORE EXECUTION.
4281 025104  000000                          0                       ;FPS AFTER EXECUTION.
4282 025106  104064                  8$:     ERROR   +64             ;ST 041 TO 046 INTO 246.
4283 025110  000401                          BR      9$
4284 025112  104064                          ERROR   +64
4285 025114                          9$:
4286
4287                                 ;MODF TEST WITH EXPONENT OF THE RESULT = -15
4288 025114                          GGG12:
     025114  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4289 025116  004737  025310                  JSR     PC,@#MODFSUB
4290 025122  034377  177777          1$:     .WORD   34377,-1        ;AC
4291 025126  040200  000000          2$:     .WORD   40200,0         ;FSRC
4292 025132  034377  177777          3$:     .WORD   34377,-1        ;FRACTIONAL RES.
4293 025136  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
4294 025142  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
4295 025146  034377  177777          6$:     .WORD   34377,-1        ;ERROR INTEGER RES.
4296 025152  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
4297 025154  000000                          0                       ;FPS AFTER EXECUTION.
4298 025156  104064                  8$:     ERROR   +64
4299 025160  000401                          BR      9$
4300 025162  104064                          ERROR   +64
4301 025164                          9$:
4302
4303                                 ;MODF TEST WITH EXPONENT OF RESULT = -64, IN ROUND MODE
4304 025164                          GGG13:
     025164  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4305 025166  004737  025310                  JSR     PC,@#MODFSUB
4306 025172  020000  000001          1$:     .WORD   20000,1         ;AC
4307 025176  040300  000000          2$:     .WORD   40300,0         ;FSRC
4308 025202  020100  000002          3$:     .WORD   20100,2         ;FRACTIONAL RES.
4309 025206  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
4310 025212  020100  000001          5$:     .WORD   20100,1         ;ERROR FRACTIONAL RES.
4311 025216  000000  000000          6$:     .WORD   0,0             ;ERROR INTEGER RES.
4312 025222  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
4313 025224  000000                          0                       ;FPS AFTER EXECUTION.
4314 025226  104065                  8$:     ERROR   +65             ;ROUND TRUNK, ST 126 INTO ROUND.
```

```
 4315 025230  000401                          BR      9$
 4316 025232  104054                          ERROR   +54
 4317 025234                          9$:
 4318
 4319                                 ;MODF TEST WITH EXPONENT OF RESULT = 11
 4320 025234                          GGG14:
      025234  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4321 025236  004737  025310                  JSR     PC,@#MODFSUB
 4322 025242  142777  170000          1$:     .WORD   142777,170000   ;AC
 4323 025246  040200  000000          2$:     .WORD   40200,0         ;FSRC
 4324 025252  140000  000000          3$:     .WORD   140000,0        ;FRACTIONAL RES.
 4325 025256  142777  160000          4$:     .WORD   142777,160000   ;INTEGER RES.
 4326 025262  040000  000000          5$:     .WORD   40000,0         ;ERROR FRACTIONAL RES.
 4327 025266  042777  160000          6$:     .WORD   42777,160000    ;ERROR INTEGER RES.
 4328 025272  000007                  7$:     7                       ;FPS BEFORE EXECUTION.
 4329 025274  000010                          10                      ;FPS AFTER EXECUTION.
 4330 025276  104066                  8$:     ERROR   +66             ;SIGN OF FRACTION.
 4331 025300  000401                          BR      9$
 4332 025302  104067                          ERROR   +67             ;SIGN OF INTEGER.
 4333 025304  000167  000366          9$:     JMP     GGGDONE         ;GO TO NEXT TEST.
 4334
 4335
 4336                                 ;THIS SUBROUTINE, MODFSUB, IS CALLED TO SETUP THE
 4337                                 ;OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK THE RESULTS.
 4338                                 ;IT IS CALLED THUS:
 4339                                 ;
 4340                                 ;       ACARG:   .WORD   X,X            ;AC OPERAND
 4341                                 ;       FSRCARG:.WORD    X,X            ;FSRC OPERAND
 4342                                 ;       FRES:    .WORD   X,X            ;FRACTIONAL RESULT
 4343                                 ;       INTRES:  .WORD   X,X            ;INTEGER RESULT
 4344                                 ;       ERFRES:  .WORD   X,X            ;ERROR FRACTION RESULT
 4345                                 ;       ERINTRES:.WORD   X,X            ;ERROR INTEGER RESULT
 4346                                 ;       FPSB:    .WORD   X              ;FPS BEFORE EXECUTION
 4347                                 ;       FPSA:    .WORD   X              ;FPS AFTER EXECUTION
 4348                                 ;       ERR1:    ERROR   +X             ;FRACTION ERROR
 4349                                 ;                BR      CONT
 4350                                 ;       ERR2:    ERROR   +X             ;INTEGER ERROR
 4351                                 ;       CONT:                           ;RETURN ADDRESS
 4352                                 ;
 4353                                 ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODF
 4354                                 ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
 4355                                 ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
 4356                                 ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
 4357                                 ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
 4358                                 ;THEN MODFSUB WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
 4359                                 ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
 4360                                 ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
 4361                                 ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
 4362                                 ;FAILURE MATCHES THE TRUE RESULT THEN  MODFSUB PASSES CONTROL TO THE
 4363                                 ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
 4364                                 ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
 4365                                 ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
 4366                                 ;IF A MATCH IS MADE HOWEVER, MODFSUB WILL RETURN CONTROL TO THE ERROR
 4367                                 ;CALL AT ERR2.
 4368
 4369 025310  012601                  MODFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
 4370 025312  012700  000200                  MOV     #200,R0         ;SET FD MODE.
```

```
4371 025316 170100                      LDFPS   R0
4372 025320 010100                      MOV     R1,R0           ;SET UP AC0
4373 025322 172410                      LDD     (R0),AC0
4374 025324 012700 025666               MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
4375 025330 172510                      LDD     (R0),AC1
4376 025332 016100 000030               MOV     30(R1),R0       ;SET UP THE FPS.
4377 025336 170100                      LDFPS   R0
4378 025340 012737 025354 001236        MOV     #1$,@#$TMP2
4379 025346 010100                      MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
4380 025350 062700 000004               ADD     #4,R0
4381
4382 025354 171410              1$:     MODF    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
4383
4384 025356 170204                      STFPS   R4              ;GET THE FPS.
4385 025360 012700 000200               MOV     #200,R0         ;SET FD MODE.
4386 025364 170100                      LDFPS   R0
4387 025366 012700 025646               MOV     #MODFT0,R0      ;GET THE FRACTIONAL RESULT.
4388 025372 174010                      STD     AC0,(R0)
4389 025374 012700 025656               MOV     #MODFT1,R0      ;GET THE INTEGER RESULT.
4390 025400 174110                      STD     AC1,(R0)
4391
4392 025402 010102                      MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
4393 025404 010237 001240               MOV     R2,@#$TMP3
4394 025410 062702 000004               ADD     #4,R2
4395 025414 010237 001242               MOV     R2,@#$TMP4
4396 025420 062702 000004               ADD     #4,R2
4397 025424 010237 001244               MOV     R2,@#$TMP5
4398 025430 062702 000004               ADD     #4,R2
4399 025434 010237 001246               MOV     R2,@#$TMP6
4400 025440 012737 025646 001250        MOV     #MODFT0,@#$TMP7
4401 025446 012737 025656 001252        MOV     #MODFT1,@#$TMP10
4402 025454 010437 001254               MOV     R4,@#$TMP11
4403 025460 016137 000032 001256        MOV     32(R1),@#$TMP12
4404
4405 025466 012702 025646               MOV     #MODFT0,R2      ;CHECK THE FRACTIONAL RESULT.
4406 025472 026112 000010               CMP     10(R1),(R2)
4407 025476 001022                      BNE     10$             ;BRANCH IF INCORRECT.
4408 025500 026162 000012 000002        CMP     12(R1),2(R2)
4409 025506 001016                      BNE     10$
4410
4411 025510 012702 025656               MOV     #MODFT1,R2      ;CHECK THE INTEGER RESULT.
4412 025514 026112 000014               CMP     14(R1),(R2)
4413 025520 001026                      BNE     15$             ;BRANCH IF INCORRECT.
4414 025522 026162 000016 000002        CMP     16(R1),2(R2)
4415 025530 001022                      BNE     15$
4416
4417 025532 026104 000032               CMP     32(R1),R4       ;CHECK THE FPS.
4418 025536 001034                      BNE     20$             ;BRANCH IF INCORRECT.
4419
4420 025540 000161 000042       9$:     JMP     42(R1)          ;RETURN.
4421
4422                             ;FRACTIONAL ERROR.
4423 025544 026112 000020       10$:    CMP     20(R1),(R2)     ;WAS THE ERROR ANTICIPATED?
4424 025550 001010                      BNE     11$             ;BRANCH IF NOT ANTICIPATED.
4425 025552 026162 000022 000002        CMP     22(R1),2(R2)
4426 025560 001004                      BNE     11$
4427 025562 010102                      MOV     R1,R2           ;THE ERROR WAS ANTICIPATED SO
```

```
4428 025564  062702  000034                    ADD     #34,R2          ;RETURN TO THE ERROR REPORT AT THE
4429                                                                     ;CALLING ROUTINE.
4430 025570  000112                            JMP     (R2)
4431
4432 025572                            11$:                             ;THE ERROR WAS NOT ANTICIPATED SO
4433 025572  104053                    12$:    ERROR   +53              ;REPORT THE INCORRECT FRACTION HERE.
4434 025574  000761                            BR      9$
4435
4436                                   ;INTEGER ERROR.
4437 025576  026112  000024            15$:    CMP     24(R1),(R2)      ;WAS THIS ERROR ANTICIPATED?
4438 025602  001010                            BNE     16$              ;BRANCH IF NOT.
4439 025604  026162  000026  000002            CMP     26(R1),2(R2)
4440 025612  001004                            BNE     16$
4441 025614  010102                            MOV     R1,R2            ;THE ERROR WAS ANTICIAPTED SO RETURN
4442 025616  062702  000040                    ADD     #40,R2           ;TO THE ERROR REPORT IN THE CALLING
4443                                                                     ;ROUTINE.
4444 025622  000112                            JMP     (R2)
4445
4446 025624                            16$:                             ;THE ERROR WAS NOT ANTICIPATED SO REPORT
4447 025624  104054                    17$:    ERROR   +54              ;THE INTEGER FAILURE HERE.
4448 025626  000744                            BR      9$
4449
4450                                   ;FPS INCORRECT.
4451 025630  010437  001254            20$:    MOV     R4,@#$TMP11      ;REPORT INCORRECT FPS.
4452 025634  016137  000032  001256            MOV     32(R1),@#$TMP12
4453 025642  104055                    21$:    ERROR   +55
4454 025644  000735                            BR      9$
4455
4456 025646  000000  000000  000000  MODFT0: .WORD   0,0,0,0
     025654  000000
4457
4458 025656  000000  000000  000000  MODFT1: .WORD   0,0,0,0
     025664  000000
4459
4460 025666  177777  177777  177777  MODP1:  .WORD   -1,-1,-1,-1
     025674  177777
4461
4462 025676                            GGGDONE:
     025676  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?).
4463
4464
4465
4473
4474                                   ;;*****************************************************************
                                       ;*TEST 17       MODD TEST
                                       ;*
                                       ;*THIS IS A TEST OF THE MODD INSTRUCTION. IT MAKES USE OF A SUBROUTINE
                                       ;*TO SET UP THE ARGUMENTS, EXECUTE THE INSTRUCTION AND CHECK THE
                                       ;*RESULTS.
                                       ;*
                                       ;;*****************************************************************
     025700  000004                    TST17:  SCOPE
4475
```

```
4476                                        ;MODD WITH (FSRC=AC=0)
4477 025702                                 HHH1:
     025702  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4478 025704  004737  027406                        JSR     PC,@#MODDSUB
4479 025710  000000  000000  000000  1$:    .WORD   0,0,0,0                 ;AC
     025716  000000
4480 025720  000000  000000  000000  2$:    .WORD   0,0,0,0                 ;FSRC
     025726  000000
4481 025730  000000  000000  000000  3$:    .WORD   0,0,0,0                 ;FRACTIONAL RES.
     025736  000000
4482 025740  000000  000000  000000  4$:    .WORD   0,0,0,0                 ;INTEGER RES.
     025746  000000
4483 025750  000000  000000  000000  5$:    .WORD   0,0,0,0                 ;ERROR FRACTIONAL RES.
     025756  000000
4484 025760  000000  000000  177777  6$:    .WORD   0,0,-1,-1               ;ERROR INTEGER RES.
     025766  177777
4485 025770  000200                 7$:    200                             ;FPS BEFORE EXECUTION.
4486 025772  000204                        204                             ;FPS AFTER EXECUTION.
4487 025774  104070                 8$:    ERROR   +70
4488 025776  000401                        BR      9$
4489 026000  104074                        ERROR   +74                     ;ST 231 TO 142 INTO 143
4490 026002                         9$:
4491
4492                                        ;MODD TEST WITH FSRC=0
4493 026002                                 HHH2:
     026002  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4494 026004  004737  027406                        JSR     PC,@#MODDSUB
4495 026010  012345  067012         1$:    .WORD   012345,67012            ;AC
4496 026014  034567  012345                .WORD   34567,012345
4497 026020  000000  000000  000000  2$:    .WORD   0,0,0,0                 ;FSRC
     026026  000000
4498 026030  000000  000000  000000  3$:    .WORD   0,0,0,0                 ;FRACTIONAL RES.
     026036  000000
4499 026040  000000  000000  000000  4$:    .WORD   0,0,0,0                 ;INTEGER RES.
     026046  000000
4500 026050  012345  067012         5$:    .WORD   012345,67012            ;ERROR FRACTIONAL RES.
4501 026054  034567  012345                .WORD   34567,012345
4502 026060  177777  177777  177777  6$:    .WORD   -1,-1,-1,-1             ;ERROR INTEGER RES.
     026066  177777
4503 026070  000213                 7$:    213                             ;FPS BEFORE EXECUTION.
4504 026072  000204                        204                             ;FPS AFTER EXECUTION.
4505 026074  104075                 8$:    ERROR   +75                     ;STORE DOUBLE ZERO
4506 026076  000401                        BR      9$
4507 026100  104076                        ERROR   +76                     ;AC V 1 <= ZERO ST 143
4508 026102                         9$:
4509
4510                                        ;MODD TEST WITH (AC=0)
4511 026102                                 HHH3:
     026102  104413                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4512 026104  004737  027406                        JSR     PC,@#MODDSUB
4513 026110  000000  000000  000000  1$:    .WORD   0,0,0,0                 ;AC
     026116  000000
4514 026120  072727  127272         2$:    .WORD   72727,127272            ;FSRC
4515 026124  072727  127272                .WORD   72727,127272
4516 026130  000000  000000  000000  3$:    .WORD   0,0,0,0                 ;FRACTIONAL RES.
     026136  000000
4517 026140  000000  000000  000000  4$:    .WORD   0,0,0,0                 ;INTEGER RES.
```

```
        026146  000000
4518    026150  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1        ;ERROR FRACTIONAL RES.
        026156  177777
4519    026160  177777  177777  177777  6$:     .WORD   -1,-1,-1,-1        ;ERROR INTEGER RES.
        026166  177777
4520    026170  000213                    7$:     213                        ;FPS BEFORE EXECUTION.
4521    026172  000204                            204                        ;FPS AFTER EXECUTION.
4522    026174  104070                    8$:     ERROR   +70
4523    026176  000401                            BR      9$
4524    026200  104071                            ERROR   +71
4525    026202                          9$:
4526
4527                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 57
4528    026202                          HHH4:
        026202  104413                            LPERR                      ;SET UP THE LOOP ON ERROR ADDRESS.
4529    026204  004737  027406                    JSR     PC,@#MODDSUB
4530    026210  056252  125252          1$:     .WORD   56252,125252       ;AC
4531    026214  125252  125250                    .WORD   125252,125250
4532    026220  040300  000000  000000  2$:     .WORD   40300,0,0,0        ;FSRC
        026226  000000
4533    026230  000000  000000  000000  3$:     .WORD   0,0,0,0            ;FRACTIONAL RES.
        026236  000000
4534    026240  056377  177777  177777  4$:     .WORD   56377,-1,-1,-4     ;INTEGER RES.
        026246  177774
4535    026250  000000  000000          5$:     .WORD   0,0                ;ERROR FRACTIONAL RES.
4536    026254  125252  125252                    .WORD   125252,125252
4537    026260  056377  177777  177777  6$:     .WORD   56377,-1,-1,-1     ;ERROR INTEGER RES.
        026266  177777
4538    026270  000213                    7$:     213                        ;FPS BEFORE EXECUTION.
4539    026272  000204                            204                        ;FPS AFTER EXECUTION.
4540    026274  104077                    8$:     ERROR   +77                ;ST 526 TO 134 INTO 135
4541    026276  000401                            BR      9$
4542    026300  104077                            ERROR   +77
4543    026302                          9$:
4544
4545                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 79
4546    026302                          HHH5:
        026302  104413                            LPERR                      ;SET UP THE LOOP ON ERROR ADDRESS.
4547    026304  004737  027406                    JSR     PC,@#MODDSUB
4548    026310  140240  000000  000000  1$:     .WORD   140240,0,0,0       ;AC
        026316  000000
4549    026320  063714  146314          2$:     .WORD   63714,146314       ;FSRC
4550    026324  133572  167737                    .WORD   133572,167737
4551    026330  000000  000000  000000  3$:     .WORD   0,0,0,0            ;FRACTIONAL RES.
        026336  000000
4552    026340  163777  177777          4$:     .WORD   163777,-1          ;INTEGER RES.
4553    026344  162531  125726                    .WORD   162531,125726
4554    026350  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1        ;ERROR FRACTIONAL RES.
        026356  177777
4555    026360  063777  177777          6$:     .WORD   63777,-1           ;ERROR INTEGER RES.
4556    026364  162531  125726                    .WORD   162531,125726
4557    026370  000210                    7$:     210                        ;FPS BEFORE EXECUTION.
4558    026372  000204                            204                        ;FPS AFTER EXECUTION.
4559    026374  104070                    8$:     ERROR   +70
4560    026376  000401                            BR      9$
4561    026400  104100                            ERROR   +100               ;ST 526 BAD SIGN
4562    026402                          9$:
```

```
                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 57
026402                                  HHH6:
026402  104413                              LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
026404  004737  027406                      JSR     PC,@#MODDSUB
026410  056200  000000  000000  1$:         .WORD   56200,0,0,1     ;AC
026416  000001
026420  040340  000000  000000  2$:         .WORD   40340,0,0,0     ;FSRC
026426  000000
026430  000000  000000  000000  3$:         .WORD   0,0,0,0         ;FRACTIONAL RES.
026436  000000
026440  056340  000000  000000  4$:         .WORD   56340,0,0,1     ;INTEGER RES.
026446  000001
026450  040000  000000  000000  5$:         .WORD   40000,0,0,0     ;ERROR FRACTIONAL RES.
026456  000000
026460  056340  000000  000000  6$:         .WORD   56340,0,0,1     ;ERROR INTEGER RES.
026466  000001
026470  000213                  7$:         213                     ;FPS BEFORE EXECUTION.
026472  000204                              204                     ;FPS AFTER EXECUTION.
026474  104101                  8$:         ERROR   +101            ;CONSTANT BAD (NOT 56)
                                                                    ;OR ST 525 TO 050 INTO 150
026476  000401                              BR      9$
026500  104101                              ERROR   +101
026502                          9$:

                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 56
026502                                  HHH7:
026502  104413                              LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
026504  004737  027406                      JSR     PC,@#MODDSUB
026510  056000  000000  000000  1$:         .WORD   56000,0,0,1     ;AC
026516  000001
026520  040340  000000  000000  2$:         .WORD   40340,0,0,0     ;FSRC
026526  000000
026530  040100  000000  000000  3$:         .WORD   40100,0,0,0     ;FRACTIONAL RES.
026536  000000
026540  056140  000000  000000  4$:         .WORD   56140,0,0,1     ;INTEGER RES.
026546  000001
026550  000000  000000  000000  5$:         .WORD   0,0,0,0         ;ERROR FRACTIONAL RES.
026556  000000
026560  056140  000000  000000  6$:         .WORD   56140,0,0,1     ;ERROR INTEGER RES.
026566  000001
026570  000213                  7$:         213                     ;FPS BEFORE EXECUTION.
```

```
4592 026572  000200                          200                    ;FPS AFTER EXECUTION.
4593 026574  104102                8$:       ERROR   +102           ;BAD CONSTANT (NOT 56) OR
4594                                                                 ;ST 525 TO 150 INTO 050
4595 026576  000401                          BR      9$
4596 026600  104102                          ERROR   +102
4597 026602                        9$:
4598
4599                               ;MODD TEST WITH EXPONENT OF THE RESULT = 36
4600 026602                        HHH8:
     026602  104413                          LPERR                  ;SET UP THE LOOP ON ERROR ADDRESS.
4601 026604  004737  027406                  JSR     PC,@#MODDSUB
4602 026610  051177  177777  177777  1$:     .WORD   51177,-1,-1,-1 ;AC
     026616  177777
4603 026620  040200  000000  000000  2$:     .WORD   40200,0,0,0    ;FSRC
     026626  000000
4604 026630  040177  177760  000000  3$:     .WORD   40177,-20,0,0  ;FRACTIONAL RES.
     026636  000000
4605 026640  051177  177777  177760  4$:     .WORD   51177,-1,-20,0 ;INTEGER RES.
     026646  000000
4606 026650  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1    ;ERROR FRACTIONAL RES.
     026656  177777
4607 026660  177777  177777  177777  6$:     .WORD   -1,-1,-1,-1    ;ERROR INTEGER RES.
     026666  177777
4608 026670  000217                    7$:     217                    ;FPS BEFORE EXECUTION.
4609 026672  000200                            200                    ;FPS AFTER EXECUTION.
4610 026674  104070                    8$:     ERROR   +70
4611 026676  000401                            BR      9$
4612 026700  104071                            ERROR   +71
4613 026702                        9$:
4614
4615                               ;MODD TEST WITH EXPONENT OF THE RESULT = 30
4616 026702                        HHH9:
     026702  104413                          LPERR                  ;SET UP THE LOOP ON ERROR ADDRESS.
4617 026704  004737  027406                  JSR     PC,@#MODDSUB
4618 026710  040200  000000  000000  1$:     .WORD   40200,0,0,0    ;AC
     026716  000000
4619 026720  047577  177777            2$:     .WORD   47577,-1       ;FSRC
4620 026724  176000  000001                    .WORD   176000,1
4621 026730  031600  000000  000000  3$:     .WORD   31600,0,0,0    ;FRACTIONAL RES.
     026736  000000
4622 026740  047577  177777            4$:     .WORD   47577,-1       ;INTEGER RES.
4623 026744  176000  000000                    .WORD   176000,0
4624 026750  000000  000000  000000  5$:     .WORD   0,0,0,0        ;ERROR FRACTIONAL RES.
     026756  000000
4625 026760  047577  177777  177777  6$:     .WORD   47577,-1,-1,-1 ;ERROR INTEGER RES.
     026766  177777
4626 026770  000200                    7$:     200                    ;FPS BEFORE EXECUTION.
4627 026772  000200                            200                    ;FPS AFTER EXECUTION.
4628 026774  104103                    8$:     ERROR   +103           ;(NORMALIZE) ST 532 TO 122
4629                                                                 ;INTO NORM.
4630 026776  000401                            BR      9$
4631 027000  104104                            ERROR   +104           ;AC V 1 <= X14
4632                                                                 ;OR ST 733 TO 156 INTO 157.
4633 027002                        9$:
4634
4635                               ;MODD TEST WITH EXPONENT OF THE RESULT = 31
4636 027002                        HHH10:
```

```
                027002  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4637            027004  004737  027406                  JSR     PC,@#MODDSUB
4638            027010  047777  177777          1$:     .WORD   47777,-1        ;AC
4639            027014  177000  000000                  .WORD   177000,0
4640            027020  040200  000000  000000  2$:     .WORD   40200,0,0,0     ;FSRC
                027026  000000
4641            027030  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
                027036  000000
4642            027040  047777  177777          4$:     .WORD   47777,-1        ;INTEGER RES.
4643            027044  177000  000000                  .WORD   177000,0
4644            027050  000000  000000  177000  5$:     .WORD   0,0,177000,0    ;ERROR FRACTIONAL RES.
                027056  000000
4645            027060  177777  177777  177777  6$:     .WORD   -1,-1,-1,-1     ;ERROR INTEGER RES.
                027066  177777
4646            027070  000213                  7$:     213                     ;FPS BEFORE EXECUTION.
4647            027072  000204                          204                     ;FPS AFTER EXECUTION.
4648            027074  104105                  8$:     ERROR   +105            ;(BUT FD) STORE X10
4649            027076  000401                          BR      9$
4650            027100  104071                          ERROR   +71
4651            027102                          9$:
4652
4653                                            ;MODD TEST WITH EXPONENT OF THE RESULT = 0
4654            027102                          HHH11:
                027102  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4655            027104  004737  027406                  JSR     PC,@#MODDSUB
4656            027110  040200  000000  000000  1$:     .WORD   40200,0,0,0     ;AC
                027116  000000
4657            027120  040177  072727          2$:     .WORD   40177,72727     ;FSRC
4658            027124  127272  072727                  .WORD   127272,72727
4659            027130  040177  072727          3$:     .WORD   40177,72727     ;FRACTIONAL RES.
4660            027134  127272  072727                  .WORD   127272,72727
4661            027140  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
                027146  000000
4662            027150  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR FRACTIONAL RES.
                027156  177777
4663            027160  000000  000000  177777  6$:     .WORD   0,0,-1,-1       ;ERROR INTEGER RES.
                027166  177777
4664            027170  000200                  7$:     200                     ;FPS BEFORE EXECUTION.
4665            027172  000200                          200                     ;FPS AFTER EXECUTION.
4666            027174  104070                  8$:     ERROR   +70
4667            027176  000401                          BR      9$
4668            027200  104106                          ERROR   +106            ;ST 246 TO 126 INTO 127 (BUT FD)
4669            027202                          9$:
4670
4671                                            ;MODD TEST WITH EXPONENT OF THE RESULT = -115
4672            027202                          HHH12:
                027202  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4673            027204  004737  027406                  JSR     PC,@#MODDSUB
4674            027210  003377  177777          1$:     .WORD   3377,-1         ;AC
4675            027214  177777  052525                  .WORD   -1,52525
4676            027220  040200  000000  000000  2$:     .WORD   40200,0,0,0     ;FSRC
                027226  000000
4677            027230  003377  177777          3$:     .WORD   3377,-1         ;FRACTIONAL RES.
4678            027234  177777  052525                  .WORD   -1,52525
4679            027240  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
                027246  000000
4680            027250  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR FRACTIONAL RES.
```

```
        027256  177777
4681    027260  000000  000000  177777  6$:     .WORD   0,0,-1,-1          ;ERROR INTEGER RES.
        027266  177777
4682    027270  000200                  7$:     200                         ;FPS BEFORE EXECUTION.
4683    027272  000200                          200                         ;FPS AFTER EXECUTION.
4684    027274  104070                  8$:     ERROR   +70
4685    027276  000401                          BR      9$
4686    027300  104107                          ERROR   +107                ;ST 446 TO 126 INTO 127 (BUT FD)
4687    027302                          9$:
4688
4689                                    ;MODD TEST WITH EXPONENT OF THE RESULT = -63, IN ROUND MODE.
4690    027302                          HHH13:
        027302  104413                          LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
4691    027304  004737  027406                  JSR     PC,@#MODDSUB
4692    027310  040300  000000  000000  1$:     .WORD   40300,0,0,0         ;AC
        027316  000000
4693    027320  020200  000000  000000  2$:     .WORD   20200,0,0,1         ;FSRC
        027326  000001
4694    027330  020300  000000  000000  3$:     .WORD   20300,0,0,2         ;FRACTIONAL RES.
        027336  000002
4695    027340  000000  000000  000000  4$:     .WORD   0,0,0,0             ;INTEGER RES.
        027346  000000
4696    027350  000000  000000  177777  5$:     .WORD   0,0,-1,-1           ;ERROR FRACTIONAL RES.
        027356  177777
4697    027360  177777  177777  177777  6$:     .WORD   -1,-1,-1,-1         ;ERROR INTEGER RES.
        027366  177777
4698    027370  000200                  7$:     200                         ;FPS BEFORE EXECUTION.
4699    027372  000200                          200                         ;FPS AFTER EXECUTION.
4700    027374  104110                  8$:     ERROR   +110                ;ST 127 INTO RND/TR
4701    027376  000401                          BR      9$
4702    027400  104071                          ERROR   +71
4703    027402  000137  030004          9$:     JMP     @#HHHDONE           ;GO TO THE NEXT TEST.
4704
4705                                    ;THIS SUBROUTINE, MODDSUB, IS CALLED TO SETUP THE
4706                                    ;OPERANDS, EXECUTE THE MODD INSTRUCTION AND CHECK THE RESULTS.
4707                                    ;IT IS CALLED THUS:
4708                                    ;
4709                                    ;               ACARG:   .WORD   X,X,X,X        ;AC OPERAND
4710                                    ;               FSRCARG: .WORD   X,X,X,X        ;FSRC OPERAND
4711                                    ;               FRES:    .WORD   X,X,X,X        ;FRACTIONAL RESULT
4712                                    ;               INTRES:  .WORD   X,X,X,X        ;INTEGER RESULT
4713                                    ;               ERFRES:  .WORD   X,X,X,X        ;ERROR FRACTION RESULT
4714                                    ;               ERINTRES:.WORD   X,X,X,X        ;ERROR INTEGER RESULT
4715                                    ;               FPSB:    .WORD   X              ;FPS BEFORE EXECUTION
4716                                    ;               FPSA:    .WORD   X              ;FPS AFTER EXECUTION
4717                                    ;               ERR1:    ERROR   +X             ;FRACTION ERROR
4718                                    ;                        BR      CONT
4719                                    ;               ERR2:    ERROR   +X             ;INTEGER ERROR
4720                                    ;               CONT:                           ;RETURN ADDRESS
4721                                    ;
4722                                    ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODD
4723                                    ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
4724                                    ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
4725                                    ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
4726                                    ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
4727                                    ;THEN MODDSUB WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
4728                                    ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
```

```
                                        ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
                                        ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
                                        ;FAILURE MATCHES THE TRUE RESULT THEN  MODDSUB PASSES CONTROL TO THE
                                        ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
                                        ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
                                        ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
                                        ;IF A MATCH IS MADE HOWEVER, MODDSUB WILL RETURN CONTROL TO THE ERROR
                                        ;CALL AT ERR2.

027406  012601                  MODDSUB:      MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
027410  012700  000200                        MOV     #200,R0         ;SET FD MODE.
027414  170100                              LDFPS   R0
027416  010100                              MOV     R1,R0           ;SET UP AC0
027420  172410                              LDD     (R0),AC0
027422  012700  025666                      MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
027426  172510                              LDD     (R0),AC1
027430  016100  000060                      MOV     60(R1),R0       ;SET UP THE FPS.
027434  170100                              LDFPS   R0
027436  012737  027452  001236              MOV     #1$,@#$TMP2
027444  010100                              MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
027446  062700  000010                      ADD     #10,R0

027452  171410                  1$:         MODD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.

027454  170204                              STFPS   R4              ;GET THE FPS.
027456  012700  000200                      MOV     #200,R0         ;SET FD MODE.
027462  170100                              LDFPS   R0
027464  012700  027764                      MOV     #MODDT0,R0      ;GET THE FRACTIONAL RESULT.
027470  174010                              STD     AC0,(R0)
027472  012700  027774                      MOV     #MODDT1,R0      ;GET THE INTEGER RESULT.
027476  174110                              STD     AC1,(R0)

027500  010102                              MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
027502  010237  001240                      MOV     R2,@#$TMP3
027506  062702  000010                      ADD     #10,R2
027512  010237  001242                      MOV     R2,@#$TMP4
027516  062702  000010                      ADD     #10,R2
027522  010237  001244                      MOV     R2,@#$TMP5
027526  062702  000010                      ADD     #10,R2
027532  010237  001246                      MOV     R2,@#$TMP6
027536  012737  027764  001250              MOV     #MODDT0,@#$TMP7
027544  012737  027774  001252              MOV     #MODDT1,@#$TMP10
027552  016137  000062  001256              MOV     62(R1),@#$TMP12
027560  010437  001254                      MOV     R4,@#$TMP11

027564  012702  027764                      MOV     #MODDT0,R2      ;CHECK THE FRACTIONAL RESULT.
027570  010103                              MOV     R1,R3
027572  062703  000020                      ADD     #20,R3
027576  012705  000004                      MOV     #4,R5
027602  022223                  2$:         CMP     (R2)+,(R3)+
027604  001020                              BNE     10$             ;BRANCH IF INCORRECT.
027606  077503                              SOB     R5,2$

027610  012702  027774                      MOV     #MODDT1,R2      ;CHECK THE INTEGER RESULT.
027614  010103                              MOV     R1,R3
027616  062703  000030                      ADD     #30,R3
027622  012705  000004                      MOV     #4,R5
```

```
4786 027626  022223                3$:     CMP     (R2)+,(R3)+
4787 027630  001026                        BNE     15$             ;BRANCH IF INCORRECT.
4788 027632  077503                        SOB     R5,3$
4789
4790
4791 027634  026104  000062                CMP     62(R1),R4       ;CHECK THE FPS.
4792 027640  001042                        BNE     20$             ;BRANCH IF INCORRECT.
4793
4794 027642  000161  000072          9$:     JMP     72(R1)          ;RETURN.
4795
4796                                ;FRACTIONAL ERROR.
4797 027646  012702  027764          10$:    MOV     #MODDT0,R2      ;WAS THE FRACTIONAL ERROR ANTICIPATED?
4798 027652  010103                        MOV     R1,R3
4799 027654  062703  000040                ADD     #40,R3
4800 027660  012705  000004                MOV     #4,R5
4801 027664  022223                50$:    CMP     (R2)+,(R3)+
4802 027666  001005                        BNE     11$             ;BRANCH IF NOT ANTICIPATED.
4803 027670  077503                        SOB     R5,50$
4804 027672  010102                        MOV     R1,R2           ;THE ERROR WAS ANTICIPATED SO
4805 027674  062702  000064                ADD     #64,R2          ;RETURN TO THE ERROR REPORT AT THE
4806                                                               ;CALLING ROUTINE.
4807 027700  000112                        JMP     (R2)
4808
4809 027702                        11$:                            ;THE ERROR WAS NOT ANTICIPATED SO
4810 027702  104070                12$:    ERROR   +70             ;REPORT THE INCORRECT FRACTION HERE.
4811 027704  000756                        BR      9$
4812
4813                                ;INTEGER ERROR.
4814 027706  012702  027774          15$:    MOV     #MODDT1,R2      ;WAS THE INTEGER ERROR ANTICIPATED?
4815 027712  010103                        MOV     R1,R3
4816 027714  062703  000050                ADD     #50,R3
4817 027720  012705  000004                MOV     #4,R5
4818 027724  022223                60$:    CMP     (R2)+,(R3)+
4819 027726  001005                        BNE     17$             ;BRANCH IF NOT ANTICIPATED.
4820 027730  077503                        SOB     R5,60$
4821 027732  010102                        MOV     R1,R2           ;THE ERROR WAS ANTICIAPTED SO RETURN
4822 027734  062702  000070                ADD     #70,R2          ;TO THE ERROR REPORT IN THE CALLING
4823                                                               ;ROUTINE.
4824 027740  000112                        JMP     (R2)
4825
4826 027742                        16$:                            ;THE ERROR WAS NOT ANTICIPATED SO REPORT
4827 027742  104071                17$:    ERROR   +71             ;THE INTEGER FAILURE HERE.
4828 027744  000736                        BR      9$
4829
4830                                ;FPS INCORRECT.
4831 027746  010437  001254          20$:    MOV     R4,@#$TMP11     ;REPORT INCORRECT FPS.
4832 027752  016137  000062  001256        MOV     62(R1),@#$TMP12
4833 027760  104072                21$:    ERROR   +72
4834 027762  000727                        BR      9$
4835
4836 027764  000000  000000  000000  MODDT0: .WORD   0,0,0,0
     027772  000000
4837
4838 027774  000000  000000  000000  MODDT1: .WORD   0,0,0,0
     030002  000000
4839
4840 030004                        HHHDONE:
```

```
        030004  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?).
  4841
  4842
  4843
  4851
  4852                                          ;TEST TITLE:UNDER/OVERFLOW, USING MODF WITH TRAPS DISABLED
  4853                                  ;;*****************************************************************
                                        ;*TEST 20        SEE COMMENT ABOVE FOR TEST TITLE
                                        ;*
                                        ;*THIS IS A TEST OF THE MODF OVERFLOW AND UNDERFLOW CONDITIONS. IT MAKES
                                        ;*USE OF A SUBROUTINE TO SETUP THE OPERANDS, EXECUTE THE MODF INSTRUCTION
                                        ;*AND CHECK THE RESULTS. TRAPS ARE DISABLED DURING THIS TEST.
                                        ;*
                                        ;;*****************************************************************
        030006  000004                  TST20:  SCOPE
  4854
  4855                                  ;UNDERFLOW TEST, WITH EXPONENT OF THE RESULT = -129, FIU = 1, FID = 1
  4856 030010                           MMM1:
        030010  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
  4857 030012  004767  000320                  JSR     PC,MODFOV
  4858 030016  020123  045676           1$:     .WORD   20123,45676     ;AC
  4859 030022  020200  000000           2$:     .WORD   20200,0         ;FSRC
  4860 030026  000123  045676           3$:     .WORD   123,45676       ;FRACTIONAL RES.
  4861 030032  000000  000000           4$:     .WORD   0,0             ;INTEGER RES.
  4862 030036  177777  177777           5$:     .WORD   -1,-1           ;ERROR FRACTIONAL RES.
  4863 030042  177777  177777           6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
  4864 030046  042000                   7$:     42000                   ;FPS BEFORE EXECUTION.
  4865 030050  142004                           142004                  ;FPS AFTER EXECUTION.
  4866 030052  000012                           12                      ;FEC
  4867 030054  104170                   8$:     ERROR   +170            ;FEC INCORRECT, UNDERFLOW.
  4868 030056  000401                           BR      9$
  4869 030060  104171                           ERROR   +171            ;AC V 1 (2,3) <= ZERO, ST 126.
  4870 030062                           9$:
  4871
  4872                                  ;UNDERFLOW EXP OF RESULT = -193, FIU = 0, FID = 1
  4873 030062                           MMM2:
        030062  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
  4874 030064  004737  030336                  JSR     PC,@#MODFOV
  4875 030070  010200  000000           1$:     .WORD   10200,0         ;AC
  4876 030074  010000  000000           2$:     .WORD   10000,0         ;FSRC
  4877 030100  000000  000000           3$:     .WORD   0,0             ;FRACTIONAL RES.
  4878 030104  000000  000000           4$:     .WORD   0,0             ;INTEGER RES.
  4879 030110  177777  177777           5$:     .WORD   -1,-1           ;ERROR FRACTIONAL RES.
  4880 030114  177777  177777           6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
  4881 030120  005013                   7$:     5013                    ;FPS BEFORE EXECUTION.
  4882 030122  005004                           5004                    ;FPS AFTER EXECUTION.
  4883 030124  000012                           12                      ;FEC
  4884 030126  000240                   8$:     NOP
  4885 030130  000401                           BR      9$
  4886 030132  104171                           ERROR   +171
  4887 030134                           9$:
  4888
  4889                                  ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, FIV = 1, FID = 1
```

```
4890 030134                          MMM3:
     030134  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4891 030136  004737  030336                  JSR     PC,@#MODFOV
4892 030142  060052  125252          1$:     .WORD   60052,125252    ;AC
4893 030146  060200  000000          2$:     .WORD   60200,0         ;FSRC
4894 030152  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
4895 030156  000052  125252          4$:     .WORD   52,125252       ;INTEGER RES.
4896 030162  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
4897 030166  000000  000000          6$:     .WORD   0,0             ;ERROR INTEGER RES.
4898 030172  041000                  7$:     41000                   ;FPS BEFORE EXECUTION.
4899 030174  141006                          141006                  ;FPS AFTER EXECUTION.
4900 030176  000010                          10                      ;FEC
4901 030200  104172                  8$:     ERROR   +172            ;BAD FEC ON OVERFLOW.
4902 030202  000401                          BR      9$
4903 030204  104173                          ERROR   +173            ;ST 520 TO STORE ZERO TWICE
4904                                                                 ;INTO 162
4905 030206                          9$:
4906
4907                                 ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 130, FIV = 0, FID = 1
4908 030206                          MMM4:
     030206  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4909 030210  004737  030336                  JSR     PC,@#MODFOV
4910 030214  060345  067654          1$:     .WORD   60345,67654     ;AC
4911 030220  060200  000000          2$:     .WORD   60200,0         ;FSRC
4912 030224  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
4913 030230  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
4914 030234  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
4915 030240  000345  067654          6$:     .WORD   345,67654       ;ERROR INTEGER RES.
4916 030244  006011                  7$:     6011                    ;FPS BEFORE EXECUTION.
4917 030246  006006                          6006                    ;FPS AFTER EXECUTION.
4918 030250  000010                          10                      ;FEC
4919 030252  000240                  8$:     NOP
4920 030254  000401                          BR      9$
4921 030256  104174                          ERROR   +174            ;ST 520 TO 162 INTO STORE ZERO TWICE.
4922 030260                          9$:
4923
4924                                 ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, RESULT NEGATIVE
4925                                 ;AND FIV = 1, FID = 1
4926 030260                          MMM5:
     030260  104413                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4927 030262  004737  030336                  JSR     PC,@#MODFOV
4928 030266  160252  125252          1$:     .WORD   160252,125252   ;AC
4929 030272  060000  000000          2$:     .WORD   60000,0         ;FSRC
4930 030276  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
4931 030302  100052  125252          4$:     .WORD   100052,125252   ;INTEGER RES.
4932 030306  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
4933 030312  000052  125252          6$:     .WORD   52,125252       ;ERROR INTEGER RES.
4934 030316  041000                  7$:     41000                   ;FPS BEFORE EXECUTION.
4935 030320  141006                          141006                  ;FPS AFTER EXECUTION.
4936 030322  000010                          10                      ;FEC
4937 030324  104172                  8$:     ERROR   +172
4938 030326  000401                          BR      9$
4939 030330  104175                          ERROR   +175            ;ST 517, BAD SIGN.
4940 030332  000137  030732          9$:     JMP     @#MMMDONE       ;GO TO THE NEXT TEST.
4941
4942                                 ;THIS SUBROUTINE, MODFOV, IS CALLED TO SETUP THE
4943                                 ;OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK THE RESULTS.
```

```
                        ;IT IS CALLED THUS:
                        ;
                        ;               ACARG:   .WORD   X,X            ;AC OPERAND
                        ;               FSRCARG:.WORD    X,X            ;FSRC OPERAND
                        ;               FRES:    .WORD   X,X            ;FRACTIONAL RESULT
                        ;               INTRES:  .WORD   X,X            ;INTEGER RESULT
                        ;               ERFRES:  .WORD   X,X            ;ERROR FRACTION RESULT
                        ;               ERINTRES:.WORD   X,X            ;ERROR INTEGER RESULT
                        ;               FPSB:    .WORD   X              ;FPS BEFORE EXECUTION
                        ;               FPSA:    .WORD   X              ;FPS AFTER EXECUTION
                        ;               FEC:     .WORD   X              ;FEC
                        ;               ERR1:    ERROR   +X             ;FEC ERROR
                        ;                        BR      CONT
                        ;               ERR2:    ERROR   +X             ;INTEGER ERROR
                        ;               CONT:                           ;RETURN ADDRESS
                        ;
                        ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODF
                        ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
                        ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
                        ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
                        ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
                        ;THEN MODFOV WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
                        ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
                        ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
                        ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
                        ;FAILURE MATCHES THE TRUE RESULT THEN  MODFOV PASSES CONTROL TO THE
                        ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
                        ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
                        ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
                        ;IF A MATCH IS MADE HOWEVER, MODFOV WILL RETURN CONTROL TO THE ERROR
                        ;CALL AT ERR2.

030336  012601          MODFOV: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
030340  012700  000200          MOV     #200,R0         ;SET FD MODE.
030344  170100                  LDFPS   R0
030346  010100                  MOV     R1,R0           ;SET UP AC0
030350  172410                  LDD     (R0),AC0
030352  012700  025666          MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
030356  172510                  LDD     (R0),AC1
030360  016100  000030          MOV     30(R1),R0       ;SET UP THE FPS.
030364  170100                  LDFPS   R0
030366  012737  030402  001236  MOV     #1$,@#$TMP2
030374  010100                  MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
030376  062700  000004          ADD     #4,R0

030402  171410          1$:     MODF    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.

030404  170204                  STFPS   R4              ;GET THE FPS.
030406  170305                  STST    R5              ;GET FEC.
030410  012700  000200          MOV     #200,R0         ;SET FD MODE.
030414  170100                  LDFPS   R0
030416  012700  030712          MOV     #MODFD0,R0      ;GET THE FRACTIONAL RESULT.
030422  174010                  STD     AC0,(R0)
030424  012700  030722          MOV     #MODFD1,R0      ;GET THE INTEGER RESULT.
030430  174110                  STD     AC1,(R0)

030432  010102                  MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
```

```
5001 030434 010237 001240          MOV   R2,@#$TMP3
5002 030440 062702 000004          ADD   #4,R2
5003 030444 010237 001242          MOV   R2,@#$TMP4
5004 030450 062702 000004          ADD   #4,R2
5005 030454 010237 001244          MOV   R2,@#$TMP5
5006 030460 062702 000004          ADD   #4,R2
5007 030464 010237 001246          MOV   R2,@#$TMP6
5008 030470 012737 030712 001250   MOV   #MODFD0,@#$TMP7
5009 030476 012737 030722 001252   MOV   #MODFD1,@#$TMP10
5010 030504 010437 001254          MOV   R4,@#$TMP11
5011 030510 016137 000032 001256   MOV   32(R1),@#$TMP12
5012 030516 010537 001260          MOV   R5,@#$TMP13
5013 030522 016137 000034 001262   MOV   34(R1),@#$TMP14
5014
5015 030530 012702 030712          MOV   #MODFD0,R2      ;CHECK THE FRACTIONAL RESULT.
5016 030534 026112 000010          CMP   10(R1),(R2)
5017 030540 001025                 BNE   10$             ;BRANCH IF INCORRECT.
5018 030542 026162 000012 000002   CMP   12(R1),2(R2)
5019 030550 001021                 BNE   10$
5020
5021 030552 012702 030722          MOV   #MODFD1,R2      ;CHECK THE INTEGER RESULT.
5022 030556 026112 000014          CMP   14(R1),(R2)
5023 030562 001016                 BNE   15$             ;BRANCH IF INCORRECT.
5024 030564 026162 000016 000002   CMP   16(R1),2(R2)
5025 030572 001012                 BNE   15$
5026
5027 030574 026104 000032          CMP   32(R1),R4       ;CHECK THE FPS.
5028 030600 001024                 BNE   20$             ;BRANCH IF INCORRECT.
5029
5030 030602 026105 000034          CMP   34(R1),R5       ;CHECK THE FEC.
5031 030606 001030                 BNE   25$             ;BRANCH IF INCORRECT.
5032
5033 030610 000161 000044     9$:  JMP   44(R1)          ;RETURN.
5034
5035                          ;FRACTIONAL ERROR.
5036 030614                   10$:                       ;THE ERROR WAS NOT ANTICIPATED SO
5037 030614 104165            12$: ERROR +165            ;REPORT THE INCORRECT FRACTION HERE.
5038 030616 000774                 BR    9$
5039
5040                          ;INTEGER ERROR.
5041 030620 026112 000024     15$: CMP   24(R1),(R2)     ;WAS THIS ERROR ANTICIPATED?
5042 030624 001010                 BNE   16$             ;BRANCH IF NOT.
5043 030626 026162 000026 000002   CMP   26(R1),2(R2)
5044 030634 001004                 BNE   16$
5045 030636 010102                 MOV   R1,R2           ;THE ERROR WAS ANTICIAPTED SO RETURN
5046 030640 062702 000042          ADD   #42,R2          ;TO THE ERROR REPORT IN THE CALLING
5047                                                     ;ROUTINE.
5048 030644 000112                 JMP   (R2)
5049
5050 030646                   16$:                       ;THE ERROR WAS NOT ANTICIPATED SO REPORT
5051 030646 104166            17$: ERROR +166            ;THE INTEGER FAILURE HERE.
5052 030650 000757                 BR    9$
5053
5054                          ;FPS INCORRECT.
5055 030652 010437 001254     20$: MOV   R4,@#$TMP11     ;REPORT INCORRECT FPS.
5056 030656 016137 000032 001256   MOV   32(R1),@#$TMP12
5057 030664 104167            21$: ERROR +167
```

```
                                        BR      9$
                                ;REPORT FEC ERROR.
                        25$:    MOV     R5,@#$TMP13
                                MOV     34(R1),@#$TMP14
                                MOV     R1,R2
                                ADD     #36,R2
                                JMP     (R2)
030712  000000  000000  000000  MODFD0: .WORD   0,0,0,0
030720  000000
030722  000000  000000  000000  MODFD1: .WORD   0,0,0,0
030730  000000
                        MMMDONE:
030732  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:UNDER/OVERFLOW, USING MODD WITH TRAPS DISABLED
                        ;;**************************************************************
                        ;*TEST 21       SEE COMMENT ABOVE FOR TEST TITLE
                        ;*
                        ;*THIS IS A TEST OF THE MODD INSTRUCTION'S OVER FLOW AND UNDER FLOW
                        ;*CONDITIONS. A SUBROUTINE IS USED TO SET UP THE OPERANDS, EXECUTE THE
                        ;*MODD INSTUCTION AND CHECK THE RESULTS.
                        ;*
                        ;;**************************************************************
030734  000004         TST21:  SCOPE

                        ;UNDERFLOW TEST WITH EXPONENT OF THE RESULT = -129, FIU = 1, FID = 1
                        NNN1:
                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                JSR     PC,@#MODDOV
                        1$:     .WORD   20252,125252    ;AC
                                .WORD   125252,125252
030754  020100  000000  000000  2$:     .WORD   20100,0,0,0     ;FSRC
030762  000000
030764  000177  177777  177777  3$:     .WORD   177,-1,-1,-1    ;FRACTIONAL RES.
030772  177777
030774  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
031002  000000
031004  020252  125252          5$:     .WORD   20252,125252    ;ERROR FRACTIONAL RES.
031010  125252  125252                  .WORD   125252,125252
031014  000000  000000  177777  6$:     .WORD   0,0,-1,-1       ;ERROR INTEGER RES.
031022  177777
031024  042200                  7$:     42200                   ;FPS BEFORE EXECUTION.
031026  142204                          142204                  ;FPS AFTER EXECUTION.
031030  000012                          12                      ;FEC
031032  104201                  8$:     ERROR   +201            ;FEC INCORRECT ON UNDERFLOW.
031034  000401                          BR      9$
```

```
                                        ERROR   +202            ;ST 155 (BUT FD)
                                9$:
                                ;UNDERFLOW TEST WITH EXPONENT OF THE RESULT = -193, FIU = 0, FID = 1
                                NNN2:
                                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                        JSR     PC,@#MODDOV
                                1$:     .WORD   10000,0         ;AC
                                        .WORD   123456,0
                                2$:     .WORD   10200,0,0,0     ;FSRC
                                3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
                                4$:     .WORD   0,0,0,0         ;INTEGER RES.
                                5$:     .WORD   0,0,0,0         ;ERROR FRACTIONAL RES.
                                6$:     .WORD   0,0             ;ERROR INTEGER RES.
                                        .WORD   123456,0
                                7$:     5213                    ;FPS BEFORE EXECUTION.
                                        5204                    ;FPS AFTER EXECUTION.
                                        12
                                8$:     NOP
                                        BR      9$
                                        ERROR   +203            ;ST 047 (BUT FD).
                                9$:
                                ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, FIV = 1, FID = 1
                                NNN3:
                                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                        JSR     PC,@#MODDOV
                                1$:     .WORD   60252,125252    ;AC
                                        .WORD   125252,125252
                                2$:     .WORD   60100,0,0,0     ;FSRC
                                3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
                                4$:     .WORD   177,-1,-1,-1    ;INTEGER RES.
                                5$:     .WORD   0,0,0,0         ;ERROR FRACTIONAL RES.
                                6$:     .WORD   177,-1          ;ERROR INTEGER RES.
                                        .WORD   125252,125252
                                7$:     41200                   ;FPS BEFORE EXECUTION.
                                        141206                  ;FPS AFTER EXECUTION.
                                        10                      ;FEC
                                8$:     ERROR   +204            ;FEC BAD ON OVERFLOW.
                                        BR      9$
                                        ERROR   +205            ;ST 520 TO 162 INTO 163 (BUT FD).
                                9$:
                                ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 130, FIV = 0, FID = 1
                                NNN4:
                                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                        JSR     PC,@#MODDOV
                                1$:     .WORD   60200,0         ;AC
                                        .WORD   125252,0
```

```
5148 031262  060200  000000  000000  2$:     .WORD   60200,0,0,0      ;FSRC
     031270  000000
5149 031272  000000  000000  000000  3$:     .WORD   0,0,0,0          ;FRACTIONAL RES.
     031300  000000
5150 031302  000000  000000  000000  4$:     .WORD   0,0,0,0          ;INTEGER RES.
     031310  000000
5151 031312  000000  000000  000000  5$:     .WORD   0,0,0,0          ;ERROR FRACTIONAL RES.
     031320  000000
5152 031322  000400  000000          6$:     .WORD   400,0            ;ERROR INTEGER RES.
5153 031326  125252  000000                  .WORD   125252,0
5154 031332  006211                  7$:     6211                     ;FPS BEFORE EXECUTION.
5155 031334  006206                          6206                     ;FPS AFTER EXECUTION.
5156 031336  000010                          10                       ;FEC
5157 031340  000240                  8$:     NOP
5158 031342  000401                          BR      9$
5159 031344  104206                          ERROR   +206             ;ST 520 TO 162 INTO STORE ZERO TWICE.
5160 031346  000137  031760          9$:     JMP     @#NNNDONE        ;GO TO NEXT TEST.
5161
5162                                ;THIS SUBROUTINE, MODDOV, IS CALLED TO SETUP THE
5163                                ;OPERANDS, EXECUTE THE MODD INSTRUCTION AND CHECK THE RESULTS.
5164                                ;IT IS CALLED THUS:
5165                                ;
5166                                ;               ACARG:   .WORD   X,X,X,X          ;AC OPERAND
5167                                ;               FSRCARG: .WORD   X,X,X,X          ;FSRC OPERAND
5168                                ;               FRES:    .WORD   X,X,X,X          ;FRACTIONAL RESULT
5169                                ;               INTRES:  .WORD   X,X,X,X          ;INTEGER RESULT
5170                                ;               ERFRES:  .WORD   X,X,X,X          ;ERROR FRACTION RESULT
5171                                ;               ERINTRES:.WORD   X,X,X,X          ;ERROR INTEGER RESULT
5172                                ;               FPSB:    .WORD   X                ;FPS BEFORE EXECUTION
5173                                ;               FPSA:    .WORD   X                ;FPS AFTER EXECUTION
5174                                ;               ERR1:    ERROR   +X               ;FRACTION ERROR
5175                                ;                        BR      CONT
5176                                ;               ERR2:    ERROR   +X               ;INTEGER ERROR
5177                                ;               CONT:                             ;RETURN ADDRESS
5178                                ;
5179                                ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODD
5180                                ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
5181                                ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
5182                                ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
5183                                ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
5184                                ;THEN MODDOV WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
5185                                ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
5186                                ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
5187                                ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
5188                                ;FAILURE MATCHES THE TRUE RESULT THEN  MODDOV PASSES CONTROL TO THE
5189                                ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
5190                                ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
5191                                ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
5192                                ;IF A MATCH IS MADE HOWEVER, MODDOV WILL RETURN CONTROL TO THE ERROR
5193                                ;CALL AT ERR2.
5194
5195 031352  012601                MODDOV: MOV     (SP)+,R1         ;GET A POINTER TO THE ARGUMENTS
5196 031354  012700  000200                MOV     #200,R0          ;SET FD MODE.
5197 031360  170100                        LDFPS   R0
5198 031362  010100                        MOV     R1,R0            ;SET UP AC0
5199 031364  172410                        LDD     (R0),AC0
5200 031366  012700  025666                MOV     #MODP1,R0        ;PUT A BACKROUND PATTERN INTO AC1.
```

```
5201 031372  172510                        LDD     (R0),AC1
5202 031374  016100  000060                MOV     60(R1),R0       ;SET UP THE FPS.
5203 031400  170100                        LDFPS   R0
5204 031402  012737  031416  001236        MOV     #1$,@#$TMP2
5205 031410  010100                        MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
5206 031412  062700  000010                ADD     #10,R0
5207
5208 031416  171410              1$:       MODD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
5209
5210 031420  170305                        STST    R5              ;GET THE FPS.
5211 031422  170204                        STFPS   R4              ;GET THE FPS.
5212 031424  012700  000200                MOV     #200,R0         ;SET FD MODE.
5213 031430  170100                        LDFPS   R0
5214 031432  012700  031740                MOV     #MODDD0,R0      ;GET THE FRACTIONAL RESULT.
5215 031436  174010                        STD     AC0,(R0)
5216 031440  012700  031750                MOV     #MODDD1,R0      ;GET THE INTEGER RESULT.
5217 031444  174110                        STD     AC1,(R0)
5218
5219 031446  010102                        MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
5220 031450  010237  001240                MOV     R2,@#$TMP3
5221 031454  062702  000010                ADD     #10,R2
5222 031460  010237  001242                MOV     R2,@#$TMP4
5223 031464  062702  000010                ADD     #10,R2
5224 031470  010237  001244                MOV     R2,@#$TMP5
5225 031474  062702  000010                ADD     #10,R2
5226 031500  010237  001246                MOV     R2,@#$TMP6
5227 031504  012737  031740  001250        MOV     #MODDD0,@#$TMP7
5228 031512  012737  031750  001252        MOV     #MODDD1,@#$TMP10
5229 031520  010437  001254                MOV     R4,@#$TMP11
5230 031524  016137  000062  001256        MOV     62(R1),@#$TMP12
5231 031532  010537  001260                MOV     R5,@#$TMP13
5232 031536  016137  000064  001262        MOV     64(R1),@#$TMP14
5233
5234 031544  012702  031740                MOV     #MODDD0,R2      ;CHECK THE FRACTIONAL RESULT.
5235 031550  010103                        MOV     R1,R3
5236 031552  062703  000020                ADD     #20,R3
5237 031556  012700  000004                MOV     #4,R0
5238 031562  022223              2$:       CMP     (R2)+,(R3)+
5239 031564  001023                        BNE     10$             ;BRANCH IF INCORRECT.
5240 031566  077003                        SOB     R0,2$
5241
5242 031570  012702  031750                MOV     #MODDD1,R2      ;CHECK THE INTEGER RESULT.
5243 031574  010103                        MOV     R1,R3
5244 031576  062703  000030                ADD     #30,R3
5245 031602  012700  000004                MOV     #4,R0
5246 031606  022223              3$:       CMP     (R2)+,(R3)+
5247 031610  001013                        BNE     15$             ;BRANCH IF INCORRECT.
5248 031612  077003                        SOB     R0,3$
```

```
                                                        ;CHECK THE FPS.
031614  026104  000062                CMP     62(R1),R4
031620  001027                        BNE     20$             ;BRANCH IF INCORRECT.

031622  026105  000064                CMP     64(R1),R5       ;CHECK THE FEC.
031626  001033                        BNE     25$

031630  000161  000074        9$:     JMP     74(R1)          ;RETURN.

                                ;FRACTIONAL ERROR.
031634                          10$:                            ;THE ERROR WAS NOT ANTICIPATED SO
031634  104176                  12$:    ERROR   +176            ;REPORT THE INCORRECT FRACTION HERE.
031636  000774                          BR      9$

                                ;INTEGER ERROR.
031640  012702  031750          15$:    MOV     #MODDD1,R2      ;WAS THE INTEGER ERROR ANTICIPATED?
031644  010103                          MOV     R1,R3
031646  062703  000050                  ADD     #50,R3
031652  012705  000004                  MOV     #4,R5
031656  022223                  60$:    CMP     (R2)+,(R3)+
031660  001005                          BNE     17$             ;BRANCH IF NOT ANTICIPATED.
031662  077503                          SOB     R5,60$
031664  010102                          MOV     R1,R2           ;THE ERROR WAS ANTICIAPTED SO RETURN
031666  062702  000072                  ADD     #72,R2          ;TO THE ERROR REPORT IN THE CALLING
                                                                ;ROUTINE.
031672  000112                          JMP     (R2)

031674                          16$:                            ;THE ERROR WAS NOT ANTICIPATED SO REPORT
031674  104177                  17$:    ERROR   +177            ;THE INTEGER FAILURE HERE.
031676  000754                          BR      9$

                                ;FPS INCORRECT.
031700  010437  001254          20$:    MOV     R4,@#$TMP11     ;REPORT INCORRECT FPS.
031704  016137  000062  001256          MOV     62(R1),@#$TMP12
031712  104200                  21$:    ERROR   +200
031714  000745                          BR      9$

                                ;REPORT FEC ERROR.
031716  010537  001260          25$:    MOV     R5,@#$TMP13
031722  016137  000064  001262          MOV     64(R1),@#$TMP14
031730  010102                          MOV     R1,R2
031732  062702  000066                  ADD     #66,R2
031736  000112                          JMP     (R2)

031740  000000  000000  000000  MODDD0: .WORD   0,0,0,0
031746  000000

031750  000000  000000  000000  MODDD1: .WORD   0,0,0,0
031756  000000

031760                          NNNDONE:
031760  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?).
```

```
5300
5342
5343
                        ;;*****************************************************************
                        ;*TEST 22       INTERRUPT CORRECT FLOWS TEST
                        ;*
                        ;*THIS IS A TEST OF THE 'CORRECT' FLOWS. THIS PART OF THE MICRO CODE
                        ;*HAS AS ITS PURPOSE INSURING THAT INTERRUPT REQUESTS MADE DURING
                        ;*CERTAIN LENGTHY FPP INSTRUCTIONS GET HONORED. THIS IS DONE
                        ;*IN A WAY SUCH THAT IF AN INTERRUPT REQUEST OCCURS DURING ONE
                        ;*OF THESE INSTRUCTIONS THE STATE OF THAT INSTRUCTION'S
                        ;*EXECUTION WILL BE THE SAME AS IF THAT INSTRUCTION HAD NEVER
                        ;*BEEN FETCHED AND ITS EXECUTION NEVER STARTED. THUS THE MICRO CODE
                        ;*WILL RESTORE ALL REGISTERS, BACK UP THE PC AND LEAVE THE
                        ;*FPS AND AC0 THROUGH AC5 UNMODIFIED.
                        ;*THE INSTRUCTIONS FOR WHICH THIS IS NECESSARY ARE:
                        ;*       ADD (OR SUB)
                        ;*       DIV
                        ;*       MUL
                        ;*       MOD
                        ;*(BOTH DOUBLE AND FLOATING)
                        ;*ALL ADDRESSING MODES WILL BE TRIED WITH THE ADDD INSTRUCTION. THEN
                        ;*EACH OF THE OTHER INSTRUCTIONS WILL BE TRIED USING MODE 1.
                        ;*NOTE THAT THIS TEST NEEDS A SPECIAL INTERRUPT MODULE,
                        ;*WHICH WILL PROBABLY ONLY BE PRESENT IN DEC'S MANUFACTURING ENVIRONMENT,
                        ;*TO RUN. THIS SPECIAL EQUIPMENT IS DESIGNED TO RAISE AN
                        ;*INTERRUPT REQUEST IN THE PROCESSOR IF A BIT IS SET IN ITS STATUS
                        ;*REGISTER AND ONLY WHEN AN FPP INSTRUCTION IS ENCOUNTERED.
                        ;*THEREFORE THIS TEST WILL BE RUN CONDITIONALLY (DEPENDENT UPON WHETHER
                        ;*OR NOT THE STATUS REGISTER OF THE TEST EQUIPMENT TIMES OUT WHEN REFERENCED).
                        ;*THIS TEST CAN ALSO BE DESELECTED BY TURNING SWITCH 7 OF THE
                        ;*SWITCH REGISTER (PHYSICAL OR VIRTUAL) ON.
                        ;*THE TEST ASSUMES THAT THE TEST EQUIPMENT'S STATUS REGISTER IS AT
                        ;*LOCATION 777774 (NOTE THAT ALL REFERENCES TO THIS LOCATION ARE
                        ;*MADE INDIRECT THROUGH THIS PROGRAMS LOCATION CORINT, SO THAT
                        ;*IF THE USER HAS MODIFIED THE TEST EQUIPMENT'S STATUS REGISTER TO
                        ;*RESPOND TO A DIFFERENT ADDRESS LOCATION CORINT MUST BE
                        ;*MADE TO CONTAIN THAT STATUS REGISTER'S NEW ADDRESS).
                        ;*THIS PROGRAM ASSUMES THAT THE TRAP VECTOR FOR THE TEST EQUIPMENT IS 110.
                        ;*AGAIN NOTE THAT ALL REFERENCES TO THIS TRAP VECTOR ARE INDIRECT, THROUGH
                        ;*THIS PROGRAM'S LOCATION CORTRP (IF THE TEST EQUIPMENT IS
                        ;*MADE TO TRAP TO A DIFFERENT VECTOR LOCATION CORTRP MUST CONTAIN THE
                        ;*ADDRESS OF THIS VECTOR).
                        ;*
                        ;;*****************************************************************
 031762  000004         TST22:  SCOPE
5344
5345 031764  132767  000200  147345        BITB    #200,$ENVM      ;SEE IF APT IS SELECTING TEST.
5346 031772  001406                        BEQ     COR1            ;BRANCH TO AUTOSIZE IF NOT.
5347 031774  032777  000200  147136        BIT     #200,@SWR       ;SEE IF THE USER HAS SELECTED THIS
5348                                                               ;TEST USING THE SWITCH REGISTER.
5349 032002  001022                        BNE     COR3            ;IF SO, PERFORM TEST.
5350 032004  000137  033142                JMP     @#CORDONE       ;ELSE DO NOT RUN TEST.
5351
5352 032010  012737  032034  000004  COR1:  MOV     #COR2,@#ERRVECT ;SEE IF THE TEST EQUIPMENT'S STATUS
5353 032016  012777  000000  001112        MOV     #0,@CORINT      ;REGISTER TIMES OUT.
5354 032024  012737  036712  000004        MOV     #CPSPUR,@#ERRVECT
5355 032032  000406                        BR      COR3            ;DIDN'T TIME OUT SO START TEST.
```

```
                                        COR2:   CMP     (SP)+,(SP)+     ;IF THE REFERENCE TIMES OUT DO
                                                MOV     #CPSPUR,@#ERRVECT
                                                JMP     @#CORDONE       ;NOT RUN TEST.

                                        ;TEST ADDD MODE 0
                                        COR3:
                                                INC     #-1
                                                BNE     COR33
                                                TYPE
                                                .WORD   CORMES
                                        COR33:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                                JSR     PC,@#CORSUB
                                        1$:     .WORD   40200,100,200,300       ;AC0
                                        2$:     .WORD   123456          ;R0
                                        3$:     200                     ;FPS
                                        4$:     ADDD    AC0,AC0         ;TEST INSTRUCTION.
                                                NOP
                                                CLR     @#CORFLG        ;RESET INTERRUPT FLAG
                                                ERROR   +252            ;NO INTERRUPT! TEST EQUIPMENT FAILED.
                                                BR      11$
                                        5$:     ERROR   +253            ;INCORRECT STATE AT INTERRUPT.
                                        11$:
                                        ;TEST ADD MODE 1
                                        COR4:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                                JSR     PC,@#CORSUB
                                        1$:     .WORD   40201,555,77007,111111  ;AC0
                                        2$:     .WORD   1$              ;R0
                                        3$:     217                     ;FPS
                                        4$:     ADDD    (R0),AC0        ;TEST INSTRUCTION
                                                NOP
                                                CLR     @#CORFLG        ;RESET INTERRUPT FLAG
                                                ERROR   +252            ;REPORT FAILURE. NO INTERRUPT.
                                                BR      11$
                                                ERROR   +254
                                        11$:

                                        ;TEST ADDD MODE 2
                                        COR5:
                                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
                                                JSR     PC,@#CORSUB
                                        1$:     .WORD   40202,111333,52525,70707        ;AC0
                                        2$:     .WORD   1$              ;R0
                                        3$:     205                     ;FPS
                                        4$:     ADDD    (R0)+,AC0       ;TEST INSTRUCTION
                                                NOP
                                                CLR     @#CORFLG        ;RESET THE INTERRUPT FLAG
                                                ERROR   +252            ;REPORT FAILURE. NO INTERRUPT.
                                                BR      11$
                                                ERROR   +255            ;CORRECT FLOWS FAILED.
                                        11$:
```

```
                                        ;TEST ADDD MODE 3
032222                                  COR6:
032222  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
032224  004737  032642                          JSR     PC,@#CORSUB
032230  040203  071735  072746        1$:     .WORD   40203,71735,72746,1     ;AC0
032236  000001
032240  032264                        2$:     .WORD   10$             ;R0
032242  000206                        3$:     206                     ;FPS
032244  172030                        4$:     ADDD    @(R0)+,AC0      ;TEST INSTRUCTION
032246  000240                                NOP
032250  005037  033124                        CLR     @#CORFLG                ;RESET THE INTERRUPT FLAG
032254  104252                                ERROR   +252            ;REPORT FAILURE, NO INTERRUPT.
032256  000403                                BR      11$
032260  104256                        5$:     ERROR   +256            ;CORRECT FLOWS FAILED.
032262  000401                                BR      11$
032264  032230                        10$:    .WORD   1$              ;USED FOR THE ADDRESSING OF THE OPERAND
                                                                        ;IN THIS MODE.
032266                                  11$:
                                        ;TEST ADDD MODE 4
032266                                  COR7:
032266  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
032270  004737  032642                          JSR     PC,@#CORSUB
032274  040204  123456  070123        1$:     .WORD   40204,123456,70123,45671        ;AC0
032302  045671
032304  032304                        2$:     .WORD   1$+10           ;R0
032306  000212                        3$:     212                     ;FPS
032310  172040                        4$:     ADDD    -(R0),AC0       ;TEST INSTRUCTION
032312  000240                                NOP
032314  005037  033124                        CLR     @#CORFLG        ;RESET THE INTERRUPT FLAG
032320  104252                                ERROR   +252            ;REPORT FAILURE. NO INTERRUPT.
032322  000401                                BR      11$
032324  104257                                ERROR   +257            ;CORRECT FLOWS FAILED
032326                                  11$:
                                        ;TEST ADDD MODE 5
032326                                  COR8:
032326  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
032330  004737  032642                          JSR     PC,@#CORSUB
032334  040205  076543  021076        1$:     .WORD   40205,76543,21076,54321 ;AC0
032342  054321
032344  032372                        2$:     .WORD   10$+2           ;R0
032346  000213                        3$:     213                     ;FPS
032350  172050                        4$:     ADDD    @-(R0),AC0      ;TEST INSTRUCTION
032352  000240                                NOP
032354  005037  033124                        CLR     @#CORFLG        ;RESET THE INTERRUPT FLAG
032360  104252                                ERROR   +252            ;REPORT ERROR. NO INTERRUPT.
032362  000403                                BR      11$
032364  104260                        5$:     ERROR   +260            ;CORRECT FLOWS FAILED.
032366  000401                                BR      11$
032370  032334                        10$:    .WORD   1$
032372                                  11$:
                                        ;TEST ADDD MODE 6
032372                                  COR9:
032372  104413                                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
032374  004737  032642                          JSR     PC,@#CORSUB
032400  040206  034353  063730        1$:     .WORD   40206,34353,63730,31323 ;AC0
```

```
        032406  031323
  5458  032410  032377                    2$:     .WORD   1$-1             ;R0
  5459  032412  000214                    3$:     214                      ;FPS
  5460  032414  172060  000001            4$:     ADDD    1(R0),AC0        ;TEST INSTRUCTION
  5461  032420  005037  033124                    CLR     @#CORFLG
  5462  032424  104252                            ERROR   +252             ;REPORT FAILURE NO TRAP.
  5463  032426  000401                            BR      11$
  5464  032430  104261                    5$:     ERROR   +261
  5465  032432                            11$:
  5466
  5467                                    ;TEST ADDD MODE 7
  5468  032432                            COR10:
        032432  104413                            LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
  5469  032434  004737  032642                    JSR     PC,@#CORSUB
  5470  032440  040210  070107  062426    1$:     .WORD   40210,70107,62426,55555 ;AC0
        032446  055555
  5471  032450  032473                    2$:     .WORD   10$-1            ;R0
  5472  032452  000204                    3$:     204                      ;FPS
  5473  032454  172070  000001            4$:     ADDD    @1(R0),AC0       ;TEST INSTRUCTION
  5474  032460  005037  033124                    CLR     @#CORFLG
  5475  032464  104252                            ERROR   +252             ;REPORT FAILURE NO TRAP
  5476  032466  000403                            BR      11$
  5477  032470  104262                    5$:     ERROR   +262             ;CORRECT FLOWS FAILED.
  5478  032472  000401                            BR      11$
  5479  032474  032440                    10$:    .WORD   1$
  5480  032476                            11$:
  5481
  5482                                    ;TEST DIVD MODE 1
  5483  032476                            COR11:
        032476  104413                            LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
  5484  032500  004737  032642                    JSR     PC,@#CORSUB
  5485  032504  040211  033445  056677    1$:     .WORD   40211,33445,56677,001122        ;AC0
        032512  001122
  5486  032514  032504                    2$:     .WORD   1$               ;R0
  5487  032516  000205                    3$:     205                      ;FPS
  5488  032520  174410                    4$:     DIVD    (R0),AC0         ;TEST INSTRUCTION
  5489  032522  000240                            NOP
  5490  032524  005037  033124                    CLR     @#CORFLG
  5491  032530  104252                            ERROR   +252             ;REPORT FAILURE, NO TRAP.
  5492  032532  000401                            BR      11$
  5493  032534  104263                    5$:     ERROR   +263             ;CORRECT FLOWS FAILED.
  5494  032536                            11$:
  5495
  5496                                    ;TEST MULD MODE 1
  5497  032536                            COR12:
        032536  104413                            LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
  5498  032540  004737  032642                    JSR     PC,@#CORSUB
  5499  032544  040212  165411  046252    1$:     .WORD   40212,165411,46252,63650        ;AC0
        032552  063650
  5500  032554  032544                    2$:     .WORD   1$               ;R0
  5501  032556  000210                    3$:     .WORD   210              ;FPS
  5502  032560  171010                    4$:     MULD    (R0),AC0         ;TEST INSTRUCTION
  5503  032562  000240                            NOP
  5504  032564  005037  033124                    CLR     @#CORFLG
  5505  032570  104252                            ERROR   +252             ;REPORT FAILURE, NO TRAP.
  5506  032572  000401                            BR      11$
  5507  032574  104264                    5$:     ERROR   +264             ;CORRECT FLOWS FAILED.
```

```
5508 032576                                  11$:
5509
5510                                         ;TEST MODD MODE 1
5511 032576                                  COR13:
     032576  104413                                 LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
5512 032600  004737  032642                         JSR     PC,@#CORSUB
5513 032604  040213  045654  054542  1$:      .WORD   40213,45654,54542,171623        ;AC0
     032612  171623
5514 032614  032604                        2$:      .WORD   1$               ;R0
5515 032616  000412                        3$:      .WORD   412              ;FPS
5516 032620  171410                        4$:      MODD    (R0),AC0         ;TEST INSTRUCTION.
5517 032622  000240                                 NOP
5518 032624  005037  033124                         CLR     @#CORFLG
5519 032630  104252                                 ERROR   +252             ;REPORT FAILURE NO TRAP.
5520 032632  000401                                 BR      11$
5521 032634  104265                        5$:      ERROR   +265             ;CORRECT FLOWS FAILED.
5522 032636  000137  033142                11$:     JMP     @#CORDONE        ;FINISHED TEST!
5523
5524                                         ;THIS SUBROUTINE, CORSUB, IS CALLED TO SET UP THE OPERANDS
5525                                         ;AND CHECK THE RESULTS IN THIS TEST. IT IS CALLED THUS:
5526                                         ;
5527                                         ;        JSR     PC,@#CORSUB
5528                                         ;1$:     .WORD   X,X,X,X          ;AC0 OPERAND
5529                                         ;2$:     .WORD   X                ;R0
5530                                         ;3$:     .WORD   X                ;FPS
5531                                         ;4$:     INST                     ;TEST INSTRUCTION TO BE
5532                                         ;                                 ;EXECUTED.
5533                                         ;        ADR                      ;AN ADDRESS OFFSET FOR
5534                                         ;                                 ;CERTAIN MODES OR NOP.
5535                                         ;        ERROR   +252             ;NO TRAP ERROR.
5536                                         ;        BR      11$
5537                                         ;5$:     ERROR   +N               ;CORRECT FLOWS FAILURE.
5538                                         ;        BR      11$              ;OPTIONAL FOR CERTAIN MODES.
5539                                         ;10$:    .WORD   ADDRESS          ;OPTIONAL FOR CERTAIN MODES.
5540                                         ;11$:
5541                                         ;
5542                                         ;CORSUB WILL PICK UP A POINTER TO THE ARGUMENTS, IN R1. AC0, R0 AND
5543                                         ;THE FPS WILL BE SET TO THE DESIGNATED VALUES. THEN THE TEST MODULE
5544                                         ;WILL BE SET UP TO INTERRUPT AND THE INSTRUCTION AT 4$ EXECUTED. IF
5545                                         ;NO TRAP OCCURS THEN THE TEST MODULE IS FAULTY. WHEN THE TRAP OCCURS
5546                                         ;THE PC ON THE STACK SHOULD BE 4$, AND AC0, R0 AND THE FPS SHOULD NOT
5547                                         ;HAVE BEEN MODIFIED. IF EVERYTHING IS CORRECT CORSUB WILL RETURN TO
5548                                         ;5$ PLUS TWO. IF AN ERROR IS DETECTED THEN CORSUB WILL RETURN TO THE
5549                                         ;ERROR REPORT AT 5$.
5550                                         ;NOTE THAT A FLAG, CORFLG, IS SET TO -1 WHEN AN INTERRUPT IS PENDING.
5551                                         ;CORFLG IS ZERO OTHERWISE.
5552
5553 032642  005037  033124          CORSUB: CLR     @#CORFLG         ;SET FLAG TO INDICATE NO INTERRUPT
5554                                                                  ;PENDING.
5555 032646  012601                         MOV     (SP)+,R1         ;GET A POINTER TO THE ARGUMENTS.
5556 032650  010102                         MOV     R1,R2            ;SET AC0.
5557 032652  012700  000200                MOV     #200,R0
5558 032656  170100                         LDFPS   R0
5559 032660  172412                         LDD     (R2),AC0
5560 032662  016100  000012                MOV     12(R1),R0        ;SET UP THE FPS.
5561 032666  170100                         LDFPS   R0
5562 032670  016100  000010                MOV     10(R1),R0        ;SET UP R0.
```

```
5563 032674  010102                        MOV     R1,R2
5564 032676  062702  000014                ADD     #14,R2
5565 032702  010237  001236                MOV     R2,@#$TMP2      ;SAVE ADDRESS OF INSTRUCTION IN CASE
5566                                                                ;OF ERROR.
5567 032706  005037  177776                CLR     @#PSW           ;CLEAR THE PRIORITY TO ALLOW INTERRUPTS.
5568 032712  012777  032740  000220        MOV     #CORTV,@CORTRP  ;SET UP THE INTERRUPT VECTOR.
5569 032720  012767  177777  000176        MOV     #-1,CORFLG      ;SET THE FLAG TO INDICATE
5570                                                                ;AN INTERRUPT IS PENDING.
5571 032726  012777  177777  000202        MOV     #-1,@CORINT     ;ENABLE THE TEST EQUIPMENT'S
5572                                                                ;TRAP FUNCTION AND GO
5573 032734  000161  000014                JMP     14(R1)          ;EXECUTE THE INSTRUCTION.
5574
5575                              ;TRAP TO HERE WHEN THE INTERRUPT OCCURS.
5576 032740  005137  033124        CORTV:  COM     @#CORFLG        ;FIRST SEE IF AN INTERRUPT WAS PENDING.
5577 032744  001060                BNE     CORTV1          ;IF NOT GO REPORT AN ERROR.
5578 032746  012777  000000  000162        MOV     #0,@CORINT      ;MAKE SURE THE TEST EQUIPMENT
5579                                                                ;IS NOT INTERRUPT ENABLED.
5580
5581 032754  170204                        STFPS   R4              ;GET THE FPS.
5582 032756  012702  000200                MOV     #200,R2         ;GET AC0
5583 032762  170102                        LDFPS   R2
5584 032764  012702  033126                MOV     #CORTMP,R2
5585 032770  174012                        STD     AC0,(R2)
5586 032772  012737  033126  001240        MOV     #CORTMP,@#$TMP3
5587 033000  010037  001244                MOV     R0,@#$TMP5
5588 033004  010437  001250                MOV     R4,@#$TMP7
5589 033010  011637  001254                MOV     (SP),@#$TMP11
5590 033014  010102                        MOV     R1,R2
5591 033016  010237  001242                MOV     R2,@#$TMP4
5592 033022  062702  000010                ADD     #10,R2
5593 033026  012237  001246                MOV     (R2)+,@#$TMP6
5594 033032  012237  001252                MOV     (R2)+,@#$TMP10
5595 033036  010237  001256                MOV     R2,@#$TMP12
5596 033042  021602                        CMP     (SP),R2         ;SEE IF THE TRAP OCCURRED
5597                                                                ;AT THE CORRECT ADDRESS.
5598 033044  001016                        BNE     CORTV0
5599 033046  022626                        CMP     (SP)+,(SP)+     ;RESET THE STACK.
5600 033050  020061  000010                CMP     R0,10(R1)       ;SEE IF R0 IS CORRECT.
5601 033054  001012                        BNE     CORTV0          ;BR IF NOT CORRECT.
5602 033056  010102                        MOV     R1,R2           ;SEE IF AC0 WAS CORRECT
5603 033060  012703  033126                MOV     #CORTMP,R3
5604 033064  012705  000004                MOV     #4,R5
5605 033070  022223                1$:     CMP     (R2)+,(R3)+
5606 033072  001003                        BNE     CORTV0          ;BRANCH IF INCORRECT.
5607 033074  077503                        SOB     R5,1$
5608 033076  000161  000032                JMP     32(R1)          ;IF EVERYTHING IS CORRECT THEN RETURN.
5609
5610 033102  000161  000030        CORTV0: JMP     30(R1)          ;CORRECT FLOWS FAILED SO GO REPORT ERROR.
5611
5612 033106  011637  001236        CORTV1: MOV     (SP),@#$TMP2    ;AN INTERRUPT OCCURRED WHEN THE FLAG
5613                                                                ;CORFLG, DID NOT INDICATE THAT ONE WAS
5614                                                                ;PENDING SO REPORT SPURIOUS TRAP.
5615 033112  0C5037  033124                CLR     @#CORFLG
5616 033116  022626                        CMP     (SP)+,(SP)+
5617 033120  104266                        ERROR   +266
5618 033122  000407                        BR      CORDONE
5619
```

```
5620 033124  000000                    CORFLG: .WORD   0
5621 033126  000000  000000  000000    CORTMP: .WORD   0,0,0,0
     033134  000000
5622
5623 033136  177774                    CORINT: .WORD   177774          ;THIS IS THE ADDRESS, 177774, OF THE
5624                                                                    ;TEST EQUIPMENT'S STATUS REGISTER.
5625                                                                    ;THE CONTENTS OF CORINT CAN BE MODIFIED
5626                                                                    ;IF THIS STATUS REGISTER'S ADDRESS IS
5627                                                                    ;CHANGED.
5628 033140  000110                    CORTRP: .WORD   110             ;THIS IS THE ADDRESS OF THE TEST EQUIPMENTS
5629                                                                    ;TRAP VECTOR. LIKE THE STATUS REGISTER'S ADDRESS
5630                                                                    ;DESCRIBED IMMEDIATELY ABOVE
5631                                                                    ;THIS VECTOR CAN BE CHANGED, BUT THE
5632                                                                    ;CONTENTS OF CORTRP MUST INDICATE THE
5633                                                                    ;CHANGE.
5634
5635 033142                            CORDONE:
     033142  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?).
5636
5637
5638
5639 033144                            TST23:
5640
5641
5642
5643                                    .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
     033144                             $EOP:
     033144  000004                             SCOPE
     033146  005067  145730                     CLR     $TSTNM          ;;ZERO THE TEST NUMBER
     033152  005067  146124                     CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
     033156  005267  146142                     INC     $PASS           ;;INCREMENT THE PASS NUMBER
     033162  042767  100000  146134           BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
     033170  005327                             DEC     (PC)+           ;;LOOP?
     033172  000001                    $EOPCT: .WORD   1
     033174  003074                             BGT     $DOAGN          ;;YES
     033176  012737                             MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
     033200  000001                    $ENDCT: .WORD   1
     033202  033172                             $EOPCT
     033204  104401  033212                     TYPE    ,65$            ;;TYPE ASCIZ STRING
     033210  000407                             BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <12><15>/END PASS #/
     033230                             64$:
     033230  016746  146070                     MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                        ;;TYPE PASS NUMBER IN OCTAL
     033234  104403                             TYPOS                   ;;GO TYPE--OCTAL ASCII
     033236     006                             .BYTE   6               ;;TYPE 6 DIGITS
     033237     000                             .BYTE   0               ;;SUPPRESS LEADING ZEROS
```

```
033240  104401  033246                  TYPE    ,67$            ;;TYPE ASCIZ STRING
033244  000421                          BR      66$             ;;GET OVER THE ASCIZ
                              ;;67$:    .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
033310                        66$:
033310  016746  145576                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS IN OCTAL
033314  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
033316     006                          .BYTE   6               ;;TYPE 6 DIGITS
033317     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
033320  104401  001313                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
033324  005067  145562                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
033330  013700  000042        $GET42:   MOV     @#42,R0         ;;GET MONITOR ADDRESS
033334  001414                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
033336  005046                          CLR     -(SP)           ;;INSURE THE "T" BIT IS CLEAR
033340  012746  033346                  MOV     #$CLR.T,-(SP)   ;;SETUP FOR AN RTI OR RTT
033344  000426                          BR      $RTRN           ;;GO DO AN RTI OR RTT TO LOAD THE PSW
                                                                ;;WITH A CLEARED "T" BIT
033346                        $CLR.T:
033346  013700  000042                  MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
033352  001405                          BEQ     $DOAGN          ;;RETURN ADDRESS
033354  000005                          RESET                   ;;CLEAR THE WORLD
033356  004710        $ENDAD:           JSR     PC,(R0)         ;;GO TO MONITOR
033360  000240                          NOP                     ;;SAVE ROOM
033362  000240                          NOP                     ;;FOR
033364  000240                          NOP                     ;;ACT11
033366                        $DOAGN:
033366  104400                          TRAP                    ;;PUSH OLD PSW AND PC ON STACK
033370  042716  000020                  BIC     #20,(SP)        ;;CLEAR THE "T" BIT
033374  032777  010000  145536          BIT     #BIT12,@SWR     ;;RUN WITH TRACE TRAP?
033402  001005                          BNE     1$              ;;BR IF NO
033404  005167  000020                  COM     $TBIT           ;;IS IT TIME FOR TRACE TRAP
033410  100402                          BMI     1$              ;;BR IF NO
033412  052716  000020                  BIS     #20,(SP)        ;;SET TRACE TRAP
033416  012746  033424          1$:     MOV     #$LOOP,-(SP)    ;;JUMP TO START OF TEST
033422  000002                $RTRN:    RTI                     ;;RETURN--THIS IS CHANGED TO
                                                                ;;AN "RTT" IF "RTT" IS A LEGAL
                                                                ;;INSTRUCTION
033424                        $LOOP:
033424  000137                          JMP     @(PC)+          ;;RETURN
033426  005020                $RTNAD:   .WORD   LOOP
033430  000000                $TBIT:    .WORD   0               ;;"T" BIT STATE INDICATOR
033432     377     377     000  $ENULL:  .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                        .EVEN
5644
5645                                    .SBTTL  SCOPE HANDLER ROUTINE
                                        ;;**************************************************************
                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        ;*AND LOAD THE TEST NUMBER($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
033436                          $SCOPE:
```

```
033436  104406                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
033440  032777  040000  145472  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
033446  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
033450  000416                $XTSTR: BR      6$              ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A ''NOP'' (NOP=240)
033452  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
033456  012737  033476  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
033464  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
033470  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
033474  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
033476  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
033500  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
033504  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
033506                          6$:;#####END OF CODE FOR THE XOR TESTER#####
033506  032777  000400  145424          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
033514  001404                          BEQ     2$              ;;BR IF NO
033516  127767  145416  145356          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
033524  001465                          BEQ     $OVER           ;;BR IF YES
033526  105767  145351          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
033532  001421                          BEQ     3$              ;;BR IF NO
033534  126767  145355  145341          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
033542  101015                          BHI     3$              ;;BR IF NO
033544  032777  001000  145366          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
033552  001404                          BEQ     4$              ;;BR IF NO
033554  016767  145330  145324  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
033562  000446                          BR      $OVER
033564  105067  145313          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
033570  005067  145506                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
033574  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
033576  032777  004000  145334  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
033604  001011                          BNE     1$              ;;BR IF YES
033606  005767  145512                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
033612  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
033614  005267  145264                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
033620  026767  145456  145256          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
033626  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
033630  012767  000001  145246  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
033636  016767  000052  145436          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
033644  105267  145232          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
033650  116767  145226  145444          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
033656  011667  145224                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
033662  011667  145222                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
033666  005067  145412                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
033672  112767  000001  145215          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
033700  016777  145176  145234  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
033706  016716  145174                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
033712  000002                          RTI                     ;;FIXES PS
033714  000001                $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
5646
5647                                    .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
033716                          $ERROR:
033716  104406                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
033720  105267  145157        7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
033724  001775                        BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
033726  016777  145150 145206        MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
033734  032777  002000 145176        BIT     #BIT10,@SWR     ;;BELL ON ERROR?
033742  001402                        BEQ     1$              ;;NO - SKIP
033744  104401  001306                TYPE    ,$BELL          ;;RING BELL
033750  005267  145136        1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
033754  011667  145136                MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
033760  162767  000002 145130        SUB     #2,$ERRPC
033766  117767  145124 145120        MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
033774  032777  020000 145136        BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
034002  001004                        BNE     20$             ;;SKIP TYPEOUTS
034004  004767  002174                JSR     PC,ERTYPE       ;;GO TO USER ERROR ROUTINE
034010  104401  001313                TYPE    ,$CRLF
034014                          20$:
034014  122767  000001 145314        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
034022  001007                        BNE     2$              ;;NO,SKIP APT ERROR REPORT
034024  116767  145064 000004        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
034032  004767  001010                JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
034036     000          21$:    .BYTE   0
034037     000                  .BYTE   0
034040  000777          22$:    BR      22$             ;;APT ERROR LOOP
034042  005777  145072        2$:     TST     @SWR            ;;HALT ON ERROR
034046  100002                        BPL     3$              ;;SKIP IF CONTINUE
034050  000000                        HALT                    ;;HALT ON ERROR!
034052  104406                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
034054  032777  001000 145056  3$:    BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
034062  001402                        BEQ     4$              ;;BR IF NO
034064  016716  145020                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
034070  005767  145210        4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
034074  001402                        BEQ     5$              ;;BR IF NONE
034076  016716  145202                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
034102                          5$:
034102  022737  033356 000042        CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
034110  001001                        BNE     6$              ;;BRANCH IF NO
034112  000000                        HALT                    ;;YES
034114                          6$:
034114  032777  001000 145016        BIT     #BIT09,@SWR
034122  001013                        BNE     ERM10
034124  011637  001162                MOV     (SP),@#$REG0            ;SEE IF ERROR #377
034130  062737  177776 001162        ADD     #-2,@#$REG0
034136  122777  000377 145016        CMPB    #377,@$REG0
034144  001002                        BNE     ERM10
034146  062716  000002                ADD     #2,(SP)
034152  000002          ERM10:  RTI
5648
5649                            .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
034154
034154  010046                  $SAVREG:
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
034156  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
034160  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
034162  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
034164  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
034166  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
034170  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
034174  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
034200  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
034204  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
034210  000002                          RTI
                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
034212                          $RESREG:
034212  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
034216  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
034222  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
034226  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
034232  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
034234  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
034236  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
034240  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
034242  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
034244  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
034246  000002                          RTI
5650
5651                            .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
                                ;*
034250  105767  144703          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
034254  100002                          BPL     1$              ;;BR IF YES
034256  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
```

```
034260  000430                    BR      3$              ;;LEAVE
034262  010046              1$:   MOV     R0,-(SP)        ;;SAVE R0
034264  017600  000002            MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
034270  122767  000001  145040    CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
034276  001011                    BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
034300  132767  000100  145031    BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
034306  001405                    BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
034310  010067  000004            MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
034314  004767  000516            JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
034320  000000              61$:  .WORD   0               ;;MESSAGE ADDRESS
034322  132767  000040  145007  62$:  BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
034330  001003                    BNE     60$             ;;YES,SKIP TYPE OUT
034332  112046              2$:   MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
034334  001005                    BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
034336  005726                    TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
034340  012600              60$:  MOV     (SP)+,R0        ;;RESTORE R0
034342  062716  000002      3$:   ADD     #2,(SP)         ;;ADJUST RETURN PC
034346  000002                    RTI                     ;;RETURN
034350  122716  000011      4$:   CMPB    #HT,(SP)        ;;BRANCH IF <HT>
034354  001430                    BEQ     8$
034356  122716  000200            CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
034362  001006                    BNE     5$
034364  005726                    TST     (SP)+           ;;POP  <CR><LF> EQUIV
034366  104401                    TYPE                    ;;TYPE A CR AND LF
034370  001313                    $CRLF
034372  105067  000200            CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
034376  000755                    BR      2$              ;;GET NEXT CHARACTER
034400  004767  000056      5$:   JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
034404  126726  144546      6$:   CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
034410  001350                    BNE     2$              ;;IF NO GO GET NEXT CHAR.
034412  016746  144536            MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                          ;;AND THE NULL CHAR.
034416  105366  000001      7$:   DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
034422  002770                    BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
034424  004767  000032            JSR     PC,$TYPEC       ;;GO TYPE A NULL
034430  105367  000142            DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
034434  000770                    BR      7$              ;;LOOP
                        ;HORIZONTAL TAB PROCESSOR
034436  112716  000040      8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
034442  004767  000014      9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
034446  132767  000007  000122    BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
034454  001372                    BNE     9$              ;;TAB STOP
034456  005726                    TST     (SP)+           ;;POP SPACE OFF STACK
034460  000724                    BR      2$              ;;GET NEXT CHARACTER
034462  105777  144462      $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
034466  100375                    BPL     $TYPEC
034470  116677  000002  144454    MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
034476  105777  144442            TSTB    @$TKS           ;;SEE IF KEYBOARD IS TALKING.
034502  100021                    BPL     2$              ;;BRANCH IF IT ISN'T.
034504  017746  144436            MOV     @$TKB,-(SP)     ;;PUSH CHARACTER ONTO STACK.
034510  042716  177600            BIC     #177600,(SP)    ;;BIT CLEAR TOP BYTE AND PARITY BIT.
034514  022726  000023            CMP     #23,(SP)+       ;;SEE IF THIS IS A ^S.
034520  001012                    BNE     2$              ;;BRANCH TO CONTINUE IF IT ISN'T.
034522  105777  144416      3$:   TSTB    @$TKS           ;;WAIT FOR ANOTHER INPUT.
034526  100375                    BPL     3$              ;;BRANCH BACK IF NOT READY.
034530  017746  144412            MOV     @$TKB,-(SP)     ;;PUSH NEXT CHARACTER ON STACK.
034534  042716  177600            BIC     #177600,(SP)    ;;BIT CLEAR TOP BYTE AND PARITY BIT.
```

```
034540  022726  000021                      CMP     #21,(SP)+       ;;SEE IF THIS IS A ^Q.
034544  001366                              BNE     3$              ;;BRANCH BACK FOR MORE WAIT IF NOT.
034546  122766  000015  000002  2$:         CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
034554  001003                              BNE     1$              ;;BRANCH IF NO
034556  105067  000014                      CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
034562  000406                              BR      $TYPEX          ;;EXIT
034564  122766  000012  000002  1$:         CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
034572  001402                              BEQ     $TYPEX          ;;BRANCH IF YES
034574  105227                              INCB    (PC)+           ;;COUNT THE CHARACTER
034576  000000                  $CHARCNT:.WORD  0                   ;;CHARACTER COUNT STORAGE
034600  000207                  $TYPEX: RTS     PC
5652
5653                            .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
                                ;;**************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$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
034602  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
034606  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
034614  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
034620  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
034624  000406                          BR      $TYPON
034626  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
034634  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
034642  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
034650  010346                          MOV     R3,-(SP)        ;;SAVE R3
034652  010446                          MOV     R4,-(SP)        ;;SAVE R4
034654  010546                          MOV     R5,-(SP)        ;;SAVE R5
034656  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
034662  005404                          NEG     R4
034664  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
034670  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
034674  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
034700  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
034704  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
034706  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
034710  000404                          BR      3$              ;;GO DO MSB
034712  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
034714  006105                          ROL     R5
```

```
        034716  006105                          ROL     R5
        034720  010503                          MOV     R5,R3
        034722  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
        034724  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
        034730  100016                          BPL     7$              ;;BR IF NO
        034732  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
        034736  001002                          BNE     4$              ;;TEST FOR 0
        034740  005704                          TST     R4              ;;SUPPRESS THIS 0?
        034742  001403                          BEQ     5$              ;;BR IF YES
        034744  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
        034746  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
        034752  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
        034756  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
        034762  104401  035022                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
        034766  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
        034772  003347                          BGT     2$              ;;BR IF MORE TO DO
        034774  002402                          BLT     6$              ;;BR IF DONE
        034776  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
        035000  000744                          BR      2$              ;;GO DO THE LAST DIGIT
        035002  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
        035004  012604                          MOV     (SP)+,R4        ;;RESTORE R4
        035006  012603                          MOV     (SP)+,R3        ;;RESTORE R3
        035010  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
        035016  012616                          MOV     (SP)+,(SP)
        035020  000002                          RTI                     ;;RETURN
        035022     000          8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
        035023     000                  .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
        035024     000          $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
        035025     000          $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
        035026  000000          $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
5654
5655                                    .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;**************************************************************
        035030  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
        035036  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
        035044  000403                          BR      $ATYC
        035046  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
        035054                          $ATYC:
        035054  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
        035056  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
        035060  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
        035064  001450                          BEQ     5$              ;;IF NOT:  BR
        035066  122767  000001  144242          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
        035074  001031                          BNE     3$              ;;IF NOT:  BR
        035076  132767  000100  144233          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
        035104  001425                          BEQ     3$              ;;IF NOT:  BR
        035106  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
        035112  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
        035120  005767  144172          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
        035124  001375                          BNE     1$              ;;IF NOT:  WAIT
        035126  010067  144200                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
        035132  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
        035134  001376                          BNE     2$
        035136  166700  144170                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
        035142  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
        035144  010067  144164                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
        035150  012767  000004  144140          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
```

```
035156  000413                          BR      5$
035160  017667  000004  000016  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
035166  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
035174  016746  142576                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
035200  004767  177044                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
035204  000000                  4$:     .WORD   0
035206                          5$:
035206  105767  000062          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
035212  001416                          BEQ     12$             ;;IF NOT:  BR
035214  005767  144116                  TST     $ENV            ;;RUNNING UNDER APT?
035220  001413                          BEQ     12$             ;;IF NOT:  BR
035222  005767  144070          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
035226  001375                          BNE     11$             ;;IF NOT:  WAIT
035230  017667  000004  144062          MOV     @4(SP),$FATAL   ;;GET ERROR #
035236  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
035244  005267  144046                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
035250  105067  000020          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
035254  105067  000013                  CLRB    $LFLG           ;;CLEAR LOG FLAG
035260  105067  000006                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
035264  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
035266  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
035270  000207                          RTS     PC              ;;RETURN
035272     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
035273     000          $LFLG:  .BYTE   0               ;;LOG FLAG
035274     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
                                .EVEN
            000200      APTSIZE=200
            000001      APTENV=001
            000100      APTSPOOL=100
            000040      APTCSUP=040
5656
5657                            .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.
035276  022767  000176  143634  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
035304  001074                          BNE     15$             ;;BRANCH IF NO
035306  105777  143632                  TSTB    @$TKS           ;;CHAR THERE?
035312  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
035314  117746  143626                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
035320  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
035324  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
035330  001062                          BNE     15$             ;;NO, RETURN TO USER
035332  126727  143576  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
035340  001456                          BEQ     15$             ;;BRANCH IF YES
035342  104401  035705                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
035346  104401  035712          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
035352  016746  142620                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
035356  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
035360  104401  035723                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
035364  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
035366  005046                          CLR     -(SP)           ;;THE NEW SWR
035370  105777  143550          7$:     TSTB    @$TKS           ;;CHAR THERE?
```

```
035374  100375                          BPL     7$              ;;IF NOT TRY AGAIN
035376  117746  143544                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
035402  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
035406  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
035412  001005                          BNE     10$             ;;BRANCH IF NOT
035414  104401  035700                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
035420  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
035424  000757                          BR      19$             ;;LET'S TRY IT AGAIN
035426  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
035432  001022                          BNE     16$             ;;BRANCH IF NO
035434  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
035440  001403                          BEQ     11$             ;;BRANCH IF YES
035442  016677  000002  143470          MOV     2(SP),@SWR      ;;SAVE NEW SWR
035450  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
035454  104401  001313          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
035460  126727  143451  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
035466  001003                          BNE     15$             ;;BRANCH IF NOT
035470  012777  000100  143446          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
035476  000002          15$:    RTI                             ;;RETURN
035500  004767  176756          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
035504  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
035510  002420                          BLT     18$             ;;BRANCH IF YES
035512  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
035516  003015                          BGT     18$             ;;BRANCH IF YES
035520  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
035524  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
035530  001403                          BEQ     17$             ;;BRANCH IF YES
035532  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
035534  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
035536  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
035540  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
035544  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
035550  000707                          BR      7$              ;;GET THE NEXT ONE
035552  104401  001312          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
035556  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
                                ;
035560  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
035562  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
035570  105777  143350          1$:     TSTB    @$TKS           ;;WAIT FOR
035574  100375                          BPL     1$              ;;A CHARACTER
035576  117766  143344  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
035604  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
035612  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
035620  001013                          BNE     3$              ;;BRANCH IF NO
035622  105777  143316          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
035626  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
035630  117746  143312                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
035634  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
035640  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
035644  001366                          BNE     2$              ;;IF NOT DISCARD IT
```

```
035646  000750                          BR      1$                      ;;YES, RESUME
035650  026627  000004  000140  3$:     CMP     4(SP),#140              ;;IS IT UPPER CASE?
035656  002407                          BLT     4$                      ;;BRANCH IF YES
035660  026627  000004  000175          CMP     4(SP),#175              ;;IS IT A SPECIAL CHAR?
035666  003003                          BGT     4$                      ;;BRANCH IF YES
035670  042766  000040  000004          BIC     #40,4(SP)               ;;MAKE IT UPPER CASE
035676  000002                  4$:     RTI                             ;;GO BACK TO USER
035700     136     125     015  $CNTLU: .ASCIZ   /^U/<15><12>           ;;CONTROL 'U'
035703     012     000
035705     136     107     015  $CNTLG: .ASCIZ   /^G/<15><12>           ;;CONTROL 'G'
035710     012     000
035712     015     012     123  $MSWR:  .ASCIZ   <15><12>/SWR = /
035715     127     122     040
035720     075     040     000
035723     040     040     116  $MNEW:  .ASCIZ   /  NEW = /
035726     105     127     040
035731     075     040     000
 5658
 5659                                   .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.
035734  010046                  $TRAP:  MOV     R0,-(SP)                ;;SAVE R0
035736  016600  000002                  MOV     2(SP),R0                ;;GET TRAP ADDRESS
035742  005740                          TST     -(R0)                   ;;BACKUP BY 2
035744  111000                          MOVB    (R0),R0                 ;;GET RIGHT BYTE OF TRAP
035746  006300                          ASL     R0                      ;;POSITION FOR INDEXING
035750  016000  035770                  MOV     $TRPAD(R0),R0           ;;INDEX TO TABLE
035754  000200                          RTS     R0                      ;;GO TO ROUTINE
                                    ;;THIS IS USE TO HANDLE THE ''GETPRI'' MACRO
035756  011646                  $TRAP2: MOV     (SP),-(SP)              ;;MOVE THE PC DOWN
035760  016666  000004  000002          MOV     4(SP),2(SP)             ;;MOVE THE PSW DOWN
035766  000002                          RTI                             ;;RESTORE THE PSW
                                        .SBTTL  TRAP TABLE
                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE ''TRAP'' INSTRUCTION.
                                        ;       ROUTINE
                                        ;       -------
035770  035756                  $TRPAD: .WORD   $TRAP2
035772  034250                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
035774  034626                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
035776  034602                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
036000  034642                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
036002  035346                          $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING
036004  035276                          $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
036006  035560                          $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
036010  034154                          $SAVREG ;;CALL=SAVREG   TRAP+10(104410) SAVE R0-R5 ROUTINE
036012  034212                          $RESREG ;;CALL=RESREG   TRAP+11(104411) RESTORE R0-R5 ROUTINE
 5660 036014  036754                    .RSET   ;;CALL=RSETUP   TRAP+12(104412) ROUTINE TO INITIALIZE AFTER EVERY TEST
 5661 036016  036746                    .LPER   ;;CALL=LPERR    TRAP+13(104413) ROUTINE TO SET LOOP ON ERROR ADDRESS
 5662         000030            $TERM=.-$TRPAD
 5663
 5664                                   .SBTTL  POWER DOWN AND UP ROUTINES
                                        ;;**************************************************************************
                                        ;POWER DOWN ROUTINE
```

```
036020  012737  036176  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
036026  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
036034  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
036036  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
036040  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
036042  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
036044  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
036046  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
036050  017746  143064                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
036054  010667  000122                  MOV     SP,$SAVR6       ;;SAVE SP
036060  012737  036072  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
036066  000000                          HALT
036070  000776                          BR      .-2             ;;HANG UP
                                ;;**************************************************************
                                ;POWER UP ROUTINE
036072  012737  036176  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
036100  016706  000076                  MOV     $SAVR6,SP       ;;GET SP
036104  005067  000072                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
036110  005267  000066          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
036114  001375                          BNE     1$              ;;OF  WORD
036116  012677  143016                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
036122  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
036124  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
036126  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
036130  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
036132  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
036134  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
036136  012737  036020  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
036144  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
036152  104401                          TYPE                    ;;REPORT THE POWER FAILURE
036154  037150          $PWRMG: .WORD   POWERM                  ;;POWER FAIL MESSAGE POINTER
036156  012716                  MOV     (PC)+,(SP)              ;;RESTART AT START
036160  004336          $PWRAD: .WORD   START                   ;;RESTART ADDRESS
036162  042766  000020  000002          BIC     #20,2(SP)       ;;CLEAR 'T' BIT
036170  005067  175234                  CLR     $TBIT           ;;CLEAR THE 'T' BIT FLAG
036174  000002                          RTI
036176  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
036200  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
036202  000000          $SAVR6: 0                               ;;PUT THE SP HERE
5665
5666
5667                            .SBTTL  ERROR TYPE OUT ROUTINE
5668                    ;;**************************************************************
                        ;;**************************************************************
5669                    ;*THIS ROUTINE IS CALLED TO TYPE AN ERROR MESSAGE WHICH IS INCLUDED
5670                    ;*IN THE ERROR MESSAGE DATA TABLE. IT IS CALLED BY THE $ERROR ROUTINE
5671                    ;*OR BY FIRST SETTING $ITEMB EQUAL TO THE ERROR TABLE ITEM TO BE PRINTED
5672                    ;*OUT AND THEN ECECUTING A:
5673                    ;*              JSR     PC,ERTYPE
5674                    ;*
5675 036204  104401    ERTYPE: TYPE                                    ;TYPE A CRLF
5676 036206  001313            .WORD   $CRLF
5677 036210  113737  001102  001232    MOVB    @#$TSTNM,@#$TMP0
5678 036216  042737  177400  001232    BIC     #177400,@#$TMP0
5679 036224  013737  001116  001234    MOV     @#$ERRPC,@#$TMP1        ;GET PC OF CALL
5680 036232  010046                    MOV     R0,-(SP)                ;SAVE R0
5681
```

```
5682 036234  113700  001114          MOVB    @#$ITEMB,R0          ;GET THE ITEM NUMBER.
5683 036240  042700  177400          BIC     #177400,R0
5684 036244  001005                  BNE     1$
5685
5686 036246  013746  001116          MOV     @#$ERRPC,-(SP)       ;IF ZERO THEN JUST
5687 036252  104402                  TYPOC                        ;PRINT THE PC
5688 036254  000137  036654          JMP     @#ERT5
5689
5690 036260  022700  000377    1$:   CMP     #377,R0
5691 036264  001005                  BNE     20$
5692 036266  016600  000004          MOV     4(SP),R0
5693 036272  011000                  MOV     (R0),R0
5694 036274  062700  000400          ADD     #400,R0
5695 036300  005300          20$:    DEC     R0                   ;OTHERWISE MAKE R0 AN
5696 036302  006300                  ASL     R0                   ;INDEX FOR THE TABLE.
5697 036304  006300                  ASL     R0
5698 036306  006300                  ASL     R0
5699 036310  062700  001442          ADD     #$ERRTB,R0
5700
5701 036314  012037  036324          MOV     (R0)+,@#2$           ;PICK UP THE ADDRESS
5702 036320  001404                  BEQ     3$                   ;OF THE EM, ERROR MESSAGE
5703 036322  104401                  TYPE
5704 036324  000000          2$:     .WORD   0
5705 036326  104401                  TYPE
5706 036330  001313                  .WORD   $CRLF
5707
5708 036332  012037  036342    3$:   MOV     (R0)+,@#4$           ;GET THE DH,DATA HEADER
5709 036336  001404                  BEQ     5$
5710 036340  104401                  TYPE
5711 036342  000000          4$:     .WORD   0
5712 036344  104401                  TYPE
5713 036346  001313                  .WORD   $CRLF
5714
5715 036350  010146          5$:     MOV     R1,-(SP)             ;SAVE R1,R2 AND R3
5716 036352  010246                  MOV     R2,-(SP)
5717 036354  010346                  MOV     R3,-(SP)
5718
5719 036356  012001                  MOV     (R0)+,R1             ;GET THE ADDRESS OF THE
5720                                                               ;DATA TABLE.
5721 036360  001002                  BNE     6$
5722 036362  000137  036642          JMP     @#ERT4               ;RETURN IF NO DATA.
5723
5724 036366  011000          6$:     MOV     (R0),R0              ;GET A POINTER TO THE DATA
5725                                                               ;FORMAT TABLE.
5726 036370  105710          ERT1:   TSTB    (R0)                 ;FORMAT ZERO?
5727 036372  001004                  BNE     7$
5728
5729 036374  013146                  MOV     @(R1)+,-(SP)         ;FORMAT ZERO SO TYPE
5730 036376  104402                  TYPOC                        ;AN OCTAL NUMBER.
5731 036400  000137  036624          JMP     @#ERT2
5732
5733 036404          7$:
5734 036404  122710  000002    8$:   CMPB    #2,(R0)              ;FORMAT TWO?
5735 036410  001011                  BNE     9$
5736
5737 036412  013102                  MOV     @(R1)+,R2            ;FORMAT TWO SO TYPE TWO
5738 036414  012246                  MOV     (R2)+,-(SP)          ;OCTAL NUMBERS.
```

```
 5739 036416  104402                        TYPOC
 5740 036420  104401                        TYPE
 5741 036422  037217                        .WORD   SPACE
 5742 036424  011246                        MOV     (R2),-(SP)
 5743 036426  104402                        TYPOC
 5744 036430  000137  036624                JMP     @#ERT2
 5745
 5746 036434  122710  000003        9$:     CMPB    #3,(R0)                 ;FORMAT THREE?
 5747 036440  001021                        BNE     10$
 5748
 5749 036442  013102                        MOV     @(R1)+,R2               ;FORMAT THREE SO TYPE
 5750 036444  012246                        MOV     (R2)+,-(SP)             ;FOUR OCTAL NUMBERS.
 5751 036446  104402                        TYPOC
 5752 036450  104401                        TYPE
 5753 036452  037217                        .WORD   SPACE
 5754 036454  012246                        MOV     (R2)+,-(SP)
 5755 036456  104402                        TYPOC
 5756 036460  104401                        TYPE
 5757 036462  037217                        .WORD   SPACE
 5758 036464  012246                        MOV     (R2)+,-(SP)
 5759 036466  104402                        TYPOC
 5760 036470  104401                        TYPE
 5761 036472  037217                        .WORD   SPACE
 5762 036474  011246                        MOV     (R2),-(SP)
 5763 036476  104402                        TYPOC
 5764 036500  000137  036624                JMP     @#ERT2
 5765
 5766 036504  122710  000004       10$:     CMPB    #4,(R0)                 ;FORMAT FOUR?
 5767 036510  001005                        BNE     11$
 5768
 5769 036512  013146                        MOV     @(R1)+,-(SP)            ;FORMAR FOUR SO TYPE
 5770 036514  104403                        TYPOS                           ;AN OCTAL NUMBER
 5771 036516     016                        .BYTE   16                      ;SUPPRESSING LEADING ZEROES.
 5772 036517     000                        .BYTE   0
 5773 036520  000137  036624                JMP     @#ERT2
 5774
 5775 036524  122710  000005       11$:     CMPB    #5,(R0)                 ;FORMAT FIVE?
 5776 036530  001006                        BNE     13$
 5777
 5778 036532  012137  036540                MOV     (R1)+,@#12$             ;FORMAT FIVE SO TYPE AN
 5779 036536  104401                        TYPE                            ;ASCIZ STRING.
 5780 036540  000000        12$:    .WORD   0
 5781 036542  000137  036630                JMP     @#ERT3
 5782
 5783 036546  122710  000011       13$:     CMPB    #11,(R0)                ;FORMAT ELEVEN?
 5784 036552  001006                        BNE     15$
 5785
 5786 036554  013137  036562                MOV     @(R1)+,@#14$            ;FORMAT ELEVEN SO PICK
 5787 036560  104401                        TYPE                            ;A POINTER TO AN ASCIZ
 5788 036562  000000        14$:    .WORD   0                               ;STRING.
 5789 036564  000137  036630                JMP     @#ERT3
 5790
 5791 036570  122710  000012       15$:     CMPB    #12,(R0)                ;FORMAT TWELVE?
 5792 036574  001012                        BNE     17$
 5793
 5794 036576  013102                        MOV     @(R1)+,R2               ;FORMAT TWELVE SO TYPE
 5795 036600  012703  000006                MOV     #6,R3                   ;TYPE SIX OCTAL NUMBERS
```

```
 5796 036604  012246                16$:   MOV     (R2)+,-(SP)
 5797 036606  104402                       TYPOC
 5798 036610  104401                       TYPE
 5799 036612  037217                       .WORD   SPACE
 5800 036614  077305                       SOB     R3,16$
 5801 036616  000137  036624               JMP     @#ERT2
 5802
 5803 036622  000000                17$:   HALT                    ;UNDEFINED FORMAT FOR DATA?????
 5804
 5805 036624  104401                ERT2:  TYPE                    ;PRINT A TAB AFTER TYPING
 5806 036626  037215                       .WORD   $TAB            ;AN DATA TABLE ENTRY
 5807                                                              ;OF ALL FORMATS EXCEPT
 5808                                                              ;ASCIZ, FORMATS 5 OR 11
 5809
 5810 036630  005200                ERT3:  INC     R0              ;POINT TO THE NEXT FORMAT
 5811 036632  005711                       TST     (R1)            ;END OF DATA TABLE.
 5812 036634  001402                       BEQ     ERT4
 5813 036636  000137  036370               JMP     @#ERT1
 5814
 5815 036642  104401                ERT4:  TYPE                    ;DONE.
 5816 036644  001313                       .WORD   $CRLF
 5817 036646  012603                       MOV     (SP)+,R3        ;RESTORE R1,R2 AND R3
 5818 036650  012602                       MOV     (SP)+,R2
 5819 036652  012601                       MOV     (SP)+,R1
 5820 036654  012600                ERT5:  MOV     (SP)+,R0        ;RESTORE R0.
 5821 036656  000207                       RTS     PC              ;AND RETURN.
 5822
 5823
 5824
 5825
 5826                                      .SBTTL  FPP SPURIOUS TRAP TO 244 HANDLER
 5827                                ;;**********************************************************************
                                    ;;**********************************************************************
 5828                                ;*THIS ROUTINE HANDLES UNEXPECTED TRAPS TO THE FPP TRAP VECTOR AT 244.
 5829                                ;*THE LAST FPP INSTRUCTION EXECUTED AND ITS ADDRESS HAS BEEN RECORDED
 5830                                ;*THESE ALONG WITH THE FEC, FPS AND PC OF TRAP ARE REPORTED.
 5831                                ;*
 5832 036660  011637  001236  FPSPUR: MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
 5833 036664  022626                       CMP     (SP)+,(SP)+     ;RESTORE SP.
 5834 036666  170200                       STFPS   R0              ;GET FPS
 5835 036670  010037  001240               MOV     R0,@#$TMP3
 5836 036674  170300                       STST    R0              ;GET FEC
 5837 036676  010037  001242               MOV     R0,@#$TMP4
 5838 036702  104247                1$:    ERROR   +247
 5839 036704  104412                       RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                   ;SEE IF THE USER HAS EXPRESSED
                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                   ;THE USER TYPED CONTROL G?).
 5840 036706  000137  033144               JMP     @#$EOP
 5841
 5842
 5843                                      .SBTTL  CPU SPURIOUS TRAP TO 4 HANDLER
 5844                                ;;**********************************************************************
                                    ;;**********************************************************************
 5845                                ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 4.
 5846                                ;*
```

```
5847 036712  011637  001236          CPSPUR: MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
5848 036716  022626                          CMP     (SP)+,(SP)+
5849 036720  104250                  1$:     ERROR   +250
5850 036722  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?).
5851 036724  000137  033144                  JMP     @#$EOP
5852
5853
5854                                         .SBTTL  CPU SPURIOUS TRAP TO 10 HANDLER
5855                                 ;;**********************************************************************
                                     ;;**********************************************************************
5856                                 ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 10.
5857                                 ;*
5858 036730  011637  001236          CPTWO:  MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
5859 036734  022626                          CMP     (SP)+,(SP)+
5860 036736  104251                  1$:     ERROR   +251
5861 036740  104412                          RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                             ;SEE IF THE USER HAS EXPRESSED
                                                                             ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                             ;THE USER TYPED CONTROL G?).
5862 036742  000137  033144                  JMP     @#$EOP
5863
5864
5865
5866
5867
5868                                         .SBTTL  SET LOOP ON ERROR ADDRESS ROUTINE
5869                                 ;;**********************************************************************
                                     ;;**********************************************************************
5870                                 ;*
5871 036746  011637  001110          .LPER:  MOV     (SP),@#$LPERR
5872 036752  000002                          RTI
5873
5874                                         .SBTTL  FLAG RESET AND CONSOLE TEST ROUTINE
5875                                 ;;**********************************************************************
                                     ;;**********************************************************************
5876                                 ;*THIS ROUTINE WILL BE CALLED AT THE END OF EACH TEST TO
5877                                 ;*RESET THE STACK, CLEAR THE FPS AND SEE IF THE USER HAS TYPED
5878                                 ;* CONTROL G ON THE TERMINAL. IF THE USER HAS TYPED CONTROL G AND
5879                                 ;*THERE IS NO PHYSICAL CONSOLE SWITCH REGISTER THEN THE CONTENTS
5880                                 ;*OF THE SOFTWARE SWITCH REGISTER WILL BE TYPED IN OCTAL ON THE
5881                                 ;*TELETYPE AND THE USER CAN MODIFY IT.
5882                                 ;*
5883 036754  023727  001140  177570  .RSET:  CMP     @#SWR,#177570           ;SEE IF THERE IS A PHYSICAL
5884                                                                          ;CONSOLE SWITCH REGISTER.
5885 036762  001001                          BNE     1$                      ;BRANCH IF NO.
5886 036764  104406                          CKSWR                           ;OTHERWISE TYPE THE CONTENTS
5887                                                                          ;OF THE PROGRAM VIRTUAL SWITCH REGISTER
5888                                                                          ;AND GIVE THE USER A CHANCE TO
5889                                                                          ;MODIFY IT.
5890 036766  012737  036660  000244  1$:     MOV     #FPSPUR,@#FPVECT
5891 036774  012737  036712  000004          MOV     #CPSPUR,@#ERRVECT
5892 037002  012737  036730  000010          MOV     #CPTWO,@#10
```

```
 5893 037010  011600                    MOV     (SP),R0                 ;SAVE RETURN ADDRESS.
 5894 037012  012706  001100             MOV     #STACK,SP               ;RESET THE STACK POINTER.
 5895 037016  005004                     CLR     R4                      ;CLEAR THE FPS.
 5896 037020  170104                     LDFPS   R4
 5897 037022  000110                     JMP     (R0)                    ;RETURN.
 5898
 5899
 5900                                    .NLIST  BEX
 5901
 5902                            ;THESE ARE SPECIAL MESSAGES:
 5903
 5904 037024      124     122     101    MSA1:   .ASCIZ  'TRAPPED AT:'<TAB><TAB>
 5905 037042      105     130     120    MSA2:   .ASCIZ  'EXPECTED TRAP AT:'<TAB>
 5906 037065      107     117     124    MSA3:   .ASCIZ  'GOT R0:'<TAB><TAB>
```

```
 5908 037077  105  130  120  MSA4:   .ASCIZ  'EXPECTED R0:'<TAB>
 5909 037115  107  117  124  MSA5:   .ASCIZ  'GOT AC0:'<TAB><TAB>
 5910 037130  105  130  120  MSA6:   .ASCIZ  'EXPECTED AC0:'<TAB><TAB>
 5911
 5912
 5913 037150  200  120  117  POWERM: .ASCIZ  <CRLF>'POWER FAILURE. PROGRAM RESTARTING.'<CRLF>
 5914 037215  011  000       $TAB:   .ASCIZ  <TAB>
 5915 037217  040  040  000  SPACE:  .ASCIZ  '  '
 5916 037222  101  103  040  MS1:    .ASCIZ  'AC OPERAND:'<TAB><TAB>
 5917 037240  106  123  122  MS2:    .ASCIZ  'FSRC OPERAND:'<TAB><TAB>
 5918 037260  101  103  060  MS3:    .ASCIZ  'AC0 BEFORE EXECUTION:'<TAB>
 5919 037307  101  103  060  MS4:    .ASCIZ  'AC0 AFTER EXECUTION:'<TAB>
 5920 037335  105  130  120  MS5:    .ASCIZ  'EXPECTED RESULT:'<TAB>
 5921 037357  107  117  124  MS6:    .ASCIZ  'GOT RESULT:'<TAB><TAB>
 5922 037375  106  122  101  MS7:    .ASCIZ  'FRACTIONAL RESULT:'<TAB>
 5923 037421  111  116  124  MS10:   .ASCIZ  'INTEGER RESULT: '<TAB>
 5924 037443  105  130  120  MS11:   .ASCIZ  'EXPECTED FRACTION:'<TAB>
 5925 037467  105  130  120  MS12:   .ASCIZ  'EXPECTED INTEGER:'<TAB>
 5926 037512  114  117  101  MS37:   .ASCIZ  'LOADED DATA: '
 5927 037530  122  105  101  MS40:   .ASCIZ  'READ DATA: '
 5928 037544  105  130  120  MS415:  .ASCIZ  'EXPECTED DATA: '
 5929 037564  104  101  124  MS41:   .ASCIZ  'DATA IN (R0) FSRC: '
 5930 037610  104  101  124  MS42:   .ASCIZ  'DATA IN AC0: '
 5931 037626  107  117  124  MS43:   .ASCIZ  'GOT RESULT: '
 5932 037643  105  130  120  MS44:   .ASCIZ  'EXPECTED RESULT: '
 5933 037665  200  124  105  CORMES: .ASCIZ  <CRLF>'TEST 22, TESTING INTERRUPTS.'<CRLF>
 5934
 5935
 5936                         ;THESE ARE ERROR MESSAGES:
 5937 037724  106  120  123  EM1:    .ASCIZ  'FPS BAD AFTER CMPD (R),A.'
 5938 037756  101  103  060  EM2:    .ASCIZ  'AC0 MODIFIED BY CMPD (R),A.'
 5942 040012                  EM3:
      040012  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5943 040036  050  102  125           .ASCIZ  '(BUT ENBT) STATE 225 WENT TO 475 INSTEAD OF 075.'
 5944 040117                  EM4:
      040117  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5945 040143  050  102  125           .ASCIZ  '(BUT ENBT) STATE 225 WENT TO 075 INSTEAD OF 475.'
 5946 040224                  EM5:
      040224  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5947 040250  050  102  125           .ASCIZ  '(BUT ENBT) STATE 035 WENT TO 075 INSTEAD OF 475.'
 5948 040331                  EM6:
      040331  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5949 040355  050  102  125           .ASCIZ  '(BUT ENBT) STATE 035 WENT TO 475 INSTEAD OF 075.'
 5950 040436                  EM7:
      040436  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5951 040462  050  102  125           .ASCIZ  '(BUT ENBT Y8) STATE 777 SHOULD HAVE GONE TO 007.'
 5952 040543                  EM10:
      040543  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5953 040567  050  102  125           .ASCIZ  '(BUT ENBT Y8) STATE 777 SHOULD HAVE GONE TO 405.'
 5954 040650                  EM11:
      040650  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5955 040674  050  102  125           .ASCIZ  '(BUT NBIT ZBIT) STATE 456 SHOULD HAVE GONE TO 010.'
 5956 040757                  EM12:
      040757  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5957 041003  050  102  125           .ASCIZ  '(BUT NBIT ZBIT) STATE 456 SHOULD HAVE GONE TO 110.'
 5958 041066                  EM13:
      041066  106  120  123           .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
```

```
 5959 041112      104      111      104                .ASCIZ  /DIDN'T TAKE THE PATH: STATE 456, TO 012, TO 363 TO 120./
 5960 041202                                  EM14:
      041202      106      120      123                .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5961 041226      050      102      125                .ASCIZ  '(BUT XNBT XZBT) STATE 363 WENT TO 140 INSTEAD OF 100.'
 5962 041314                                  EM15:
      041314      106      120      123                .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5963 041340      050      102      125                .ASCIZ  '(BUT XNBT XZBT) STATE 363 WENT TO 100 INSTEAD OF 140.'
 5964 041426                                  EM16:
      041426      106      120      123                .ASCII  'FPS BAD AFTER CMPD.'<CRLF>
 5965 041452      104      111      104                .ASCIZ  /DIDN'T TAKE THE PATH: STATE 777, TO 407./
 5966 041523      104      111      126     EM17:      .ASCIZ  'DIVD (R),A TRAPPED TO 244. FSRC=0 AND FID=1.'
 5967 041600      106      120      123     EM20:      .ASCIZ  'FPS BAD AFTER DIVD (R),A.'
 5968 041632      106      105      103     EM21:      .ASCIZ  'FEC BAD AFTER DIVD (R),A.'
 5969 041664      104      111      126     EM22:      .ASCIZ  /DIVD (R),A DIDN'T TRAP TO 244. FSRC=0 AND FID=0./
 5970 041745      104      111      126     EM23:      .ASCIZ  'DIVF (R),A FAILED.'
 5971 041770      106      120      123     EM32:      .ASCIZ  'FPS BAD AFTER DIVF (R),A.'
 5975 042022                                  EM24:
      042022      104      111      126                .ASCII  'DIVF (R),A FAILED.'
 5976 042044      050      102      125                .ASCIZ  '(BUT Y61) WENT TO STATE 006 INSTEAD OF 206.'
 5977 042120                                  EM25:
      042120      104      111      126                .ASCII  'DIVF (R),A FAILED.'
 5978 042142      130      117      122                .ASCIZ  'XOR OF SIGN BITS FAILED STATE 470.'
 5979 042205                                  EM26:
      042205      104      111      126                .ASCII  'DIVF (R),A FAILED.'
 5980 042227      050      102      125                .ASCIZ  '(BUT Y61) WENT TO STATE 206 INSTEAD OF 006.'
 5981          042120                         EM27=EM25
 5982 042303                                  EM30:
      042303      104      111      126                .ASCII  'DIVF (R),A FAILED.'
 5983 042325      124      122      125                .ASCIZ  'TRUNCATION ERROR. FT=1.'
 5984 042355                                  EM31:
      042355      104      111      126                .ASCII  'DIVF (R),A FAILED.'
 5985 042377      122      117      125                .ASCIZ  'ROUND ERROR. FT=0.'
 5986 042422      104      111      126     EM33:      .ASCIZ  'DIVD (R),A FAILED.'
 5987 042445      106      120      123     EM34:      .ASCIZ  'FPS BAD AFTER DIVD (R),A.'
 5991 042477                                  EM35:
      042477      104      111      126                .ASCII  'DIVD (R),A FAILED.'<CRLF>
 5992 042522      124      122      125                .ASCIZ  'TRUNCATION ERROR. FT=1.'
 5993 042552                                  EM36:
      042552      104      111      126                .ASCII  'DIVD (R),A FAILED.'<CRLF>
 5994 042575      122      117      125                .ASCIZ  'ROUND ERROR. FT=0.'
 5995 042620      115      125      114     EM37:      .ASCIZ  'MULF (R),A FAILED.'
 5999 042643      106      120      123     EM40:      .ASCIZ  'FPS BAD AFTER MULF (R),A.'
 6000 042675                                  EM41:
      042675      115      125      114                .ASCII  'MULF (R),A FAILED.'<CRLF>
 6001 042720      123      111      107                .ASCIZ  'SIGN BIT BAD STATE 511.'
 6002 042750                                  EM42:
      042750      115      125      114                .ASCII  'MULF (R),A FAILED.'<CRLF>
 6003 042773      116      117      122                .ASCII  'NORMALIZATION FAILED.'<CRLF>
 6004 043021      050      102      125                .ASCIZ  '(BUT Y62) STATE 252 WENT TO 044 INSTEAD OF 444.'
 6005 043101                                  EM43:
      043101      115      125      114                .ASCII  'MULF (R),A FAILED.'<CRLF>
 6006 043124      116      117      122                .ASCII  'NORMALIZATION FAILED.'<CRLF>
 6007 043152      050      102      125                .ASCIZ  '(BUT Y62) STATE 252 WENT TO 444 INSTEAD OF 044.'
 6008 043232                                  EM44:
      043232      115      125      114                .ASCII  'MULF (R),A FAILED.'<CRLF>
 6009 043255      122      117      125                .ASCIZ  'ROUND ERROR. FT=0.'
 6010 043300                                  EM45:
```

```
        043300  115  125  114          .ASCII  'MULF (R),A FAILED.'<CRLF>
6011    043323  124  122  125          .ASCIZ  'TRUNCATION ERROR. FT=1.'
6012    043353  106  120  123  EM46:    .ASCIZ  'FPS BAD AFTER MULD (R),A.'
6016    043405  115  125  114  EM246:   .ASCIZ  'MULD (R),A FAILED.'
6017    043430                EM47:
        043430  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6018    043453  102  101  104          .ASCII  'BAD CONSTANT USED IN THE MUL ALGORITHM.'
6019    043522  200  125  123          .ASCIZ  <CRLF>'USED 24 INSTEAD OF 56 STATE 020.'
6020    043564                EM50:
        043564  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6021    043607  124  122  125          .ASCIZ  'TRUNCATION ERROR. FT=1.'
6022    043637                EM51:
        043637  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6023    043662  122  117  125          .ASCIZ  'ROUND ERROR. FT=0.'
6024    043705                EM52:
        043705  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6025    043730  102  101  104          .ASCIZ  'BAD CONSTANT USED IN ROUNDING, FT=0.'
6026    043775  106  120  123  EM111:   .ASCIZ  'FPS BAD AFTER MULF (R),A. EXPECTED OVERFLOW.'
6027    044052  106  120  123  EM112:   .ASCIZ  'FPS BAD AFTER MULF (R),A. EXPECTED UNDERFLOW.'
6028    044130                EM113:
        044130  115  125  114          .ASCII  'MULF (R),A FAILED.'<CRLF>
6029    044153  105  130  120          .ASCIZ  'EXPECTING OVERFLOW, FIV=0.'
6030    044206                EM114:
        044206  115  125  114          .ASCII  'MULF (R),A FAILED.'<CRLF>
6031    044231  105  130  120          .ASCIZ  'EXPECTING UNDERFLOW, FIU=0.'
6032    044265  115  125  114  EM115:   .ASCIZ  'MULF (R),A TRAPPED TO 244 ON OVERFLOW. FIV=0.'
6033    044343  115  125  114  EM116:   .ASCIZ  'MULF (R),A TRAPPED TO 244 ON UNDERFLOW. FIU=0.'
6040    044422                EM117:
        044422  115  125  114          .ASCII  'MULF (R),A TRAPPED TO 244 ON UNDERFLOW. FIU=0.'
6041    044500  050  102  125          .ASCIZ  '(BUT FIU) STATE 331 WENT TO 155 INSTEAD OF 115.'
6042    044560                EM120:
        044560  115  125  114          .ASCII  'MULF (R),A TRAPPED TO 244 ON UNDERFLOW. FIU=0.'
6043    044636  050  102  125          .ASCIZ  '(BUT FIU) STATE 137 WENT TO 155 INSTEAD OF 115.'
6044    044716                EM121:
        044716  115  125  114          .ASCII  'MULF (R),A TRAPPED TO 244 ON OVERFLOW. FIV=0.'
6045    044773  050  102  125          .ASCIZ  '(BUT FIV) STATE 333 WENT TO 136 INSTEAD OF 116.'
6046    045053                EM122:
        045053  115  125  114          .ASCII  'MULF (R),A TRAPPED TO 244 ON OVERFLOW. FIV=0.'
6047    045130  050  102  125          .ASCIZ  '(BUT FIV) STATE 133 WENT TO 136 INSTEAD OF 116.'
6048    045210  106  120  123  EM123:   .ASCIZ  'FPS BAD AFTER MULD (R),A. EXPECTING OVERFLOW.'
6049    045266  106  120  123  EM124:   .ASCIZ  'FPS BAD AFTER MULD (R),A. EXPECTING UNDERFLOW.'
6050    045345                EM125:
        045345  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6051    045370  105  130  120          .ASCIZ  'EXPECTING OVERFLOW, FIV=0.'
6052    045423                EM126:
        045423  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6053    045446  105  130  120          .ASCIZ  'EXPECTING UNDERFLOW, FIU=0.'
6054    045502  115  125  114  EM127:   .ASCIZ  'MULD (R),A TRAPPED TO 244 ON OVERFLOW. FIV=0.'
6055    045560  115  125  114  EM130:   .ASCIZ  'MULD (R),A TRAPPED TO 244 ON UNDERFLOW. FIU=0.'
6062    045637                EM131:
        045637  115  125  114          .ASCII  'MULF (R),A TRAPPED TO 244 ON UNDERFLOW. FIU=0.'
6063    045715  050  102  125          .ASCIZ  '(BUT FIU) STATE 331 WENT TO 155 INSTEAD OF 115.'
6064    045775                EM132:
        045775  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6065    046020  105  130  120          .ASCII  'EXPECTING UNDERFLOW, FIU=0.'
6066    046053  200  050  102          .ASCIZ  <CRLF>'(BUT FD) STATE 115 WENT TO 424 INSTEAD OF 425.'
6067    046133                EM133:
```

```
                046133  115  125  114          .ASCII  'MULF (R),A TRAPPED TO 244 ON UNDERFLOW. FIU=0.'
6068            046211  050  102  125          .ASCIZ  '(BUT FIU) STATE 137 WENT TO 155 INSTEAD OF 115.'
6069            046271                  EM134:
                046271  115  125  114          .ASCII  'MULD (R),A TRAPPED TO 244 ON OVERFLOW. FIV=0.'
6070            046346  050  102  125          .ASCIZ  '(BUT FIV) STATE 333 WENT TO 136 INSTEAD OF 116.'
6071            046426                  EM135:
                046426  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6072            046451  105  130  120          .ASCII  'EXPECTING OVERFLOW, FIV=0.'
6073            046503  200  050  102          .ASCIZ  <CRLF>'(BUT FD) STATE 116 WENT TO 424 INSTEAD OF 425.'
6074            046563                  EM136:
                046563  115  125  114          .ASCII  'MULD (R),A TRAPPED TO 244 ON OVERFLOW. FIV=0.'
6075            046640  050  102  125          .ASCIZ  '(BUT FIV) STATE 133 WENT TO 136 INSTEAD OF 116.'
6076            046720  106  105  103  EM137:  .ASCIZ  'FEC BAD AFTER MULF (R),A. EXPECTING OVERFLOW, FEC=10.'
6077            047006  106  105  103  EM140:  .ASCIZ  'FEC BAD AFTER MULF (R),A. EXPECTING UNDERFLOW, FEC=12.'
6078                    043775          EM141=EM111
6079                    044052          EM142=EM112
6080            047075                  EM143:
                047075  115  125  114          .ASCII  'MULF (R),A FAILED.'<CRLF>
6081            047120  105  130  120          .ASCIZ  'EXPECTING OVERFLOW, FIV=1.'
6082            047153                  EM144:
                047153  115  125  114          .ASCII  'MULF (R),A FAILED.'<CRLF>
6083            047176  105  130  120          .ASCIZ  'EXPECTING UNDERFLOW, FIU=1.'
6090            047232                  EM145:
                047232  115  125  114          .ASCII  'MULF (R),A FAILED TO TRAP TO 244 ON UNDERFLOW. FIU=1.'
6091            047317  050  102  125          .ASCIZ  '(BUT FIU) STATE 331 WENT TO 115 INSTEAD OF 155.'
6092            047377                  EM146:
                047377  115  125  114          .ASCII  'MULF (R),A FAILED TO TRAP TO 244 ON UNDERFLOW. FIU=1.'
6093            047464  050  102  125          .ASCIZ  '(BUT FIU) STATE 137 WENT TO 115 INSTEAD OF 155.'
6094            047544                  EM147:
                047544  115  125  114          .ASCII  'MULF (R),A FAILED TO TRAP TO 244 ON OVERFLOW. FIV=1.'
6095            047630  050  102  125          .ASCIZ  '(BUT FIV) STATE 333 WENT TO 116 INSTEAD OF 136.'
6096            047710                  EM150:
                047710  115  125  114          .ASCII  'MULF (R),A FAILED TO TRAP TO 244 ON OVERFLOW. FIV=1.'
6097            047774  050  102  125          .ASCIZ  '(BUT FIV) STATE 133 WENT TO 116 INSTEAD OF A36.'
6098            050054  106  105  103  EM151:  .ASCIZ  'FEC BAD AFTER MULD (R),A. EXPECTING OVERFLOW, FEC=10.'
6099            050142  106  105  103  EM152:  .ASCIZ  'FEC BAD AFTER MULD (R),A. EXPECTING UNDERFLOW, FEC=12.'
6100                    045210          EM153=EM123
6101                    045266          EM154=EM124
6102            050231                  EM155:
                050231  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6103            050254  105  130  120          .ASCIZ  'EXPECTING OVERFLOW, FIV=1.'
6104            050307                  EM156:
                050307  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6105            050332  105  130  120          .ASCIZ  'EXPECTING UNDERFLOW, FIU=1.'
6112            050366                  EM157:
                050366  115  125  114          .ASCII  'MULD (R),A FAILED TO TRAP TO 244 ON UNDERFLOW. FIU=1.'
6113            050453  050  102  125          .ASCIZ  '(BUT FIU) STATE 331 WENT TO 115 INSTEAD OF 155.'
6114            050533                  EM160:
                050533  115  125  114          .ASCII  'MULD (R),A FAILED.'<CRLF>
6115            050556  105  130  120          .ASCII  'EXPECTING UNDERFLOW, FIU=1.'
6116            050611  200  050  102          .ASCIZ  <CRLF>'(BUT FD) STATE 155 WENT TO 426 INSTEAD OF 427.'
6117            050671                  EM161:
                050671  115  125  114          .ASCII  'MULD (R),A FAILED TO TRAP TO 244 ON UNDERFLOW. FIU=1.'
6118            050756  050  102  125          .ASCIZ  '(BUT FIU) STATE 137 WENT TO 115 INSTEAD OF 155.'
6119            051036                  EM162:
                051036  115  125  114          .ASCII  'MULD (R),A FAILED TO TRAP TO 244 ON OVERFLOW. FIV=1.'
6120            051122  050  102  125          .ASCIZ  '(BUT FIV) STATE 333 WENT TO 116 INSTEAD OF 136.'
```

```
6121 051202                                EM163:
     051202  115  125  114                        .ASCII  'MULD (R),A FAILED.'<CRLF>
6122 051225  105  130  120                        .ASCII  'EXPECTING OVERFLOW, FIV=1.'
6123 051257  200  050  102                        .ASCIZ  <CRLF>'(BUT FD) STATE 700 WENT TO 426 INSTEAD OF 427.'
6124 051337                                EM164:
     051337  115  125  114                        .ASCII  'MULD (R),A FAILED TO TRAP TO 244 ON OVERFLOW. FIV=1.'
6125 051423  050  102  125                        .ASCIZ  '(BUT FIV) STATE 133 WENT TO 116 INSTEAD OF 136.'
6126 051503  106  120  123  EM55:                 .ASCIZ  'FPS BAD AFTER MODF (R),A.'
6127 051535  115  117  104  EM53:                 .ASCIZ  'MODF (R),A FRACTION BAD.'
6128 051566  115  117  104  EM54:                 .ASCIZ  'MODF (R),A INTEGER BAD.'
6135 051616                                EM56:
     051616  115  117  104                        .ASCII  'MODF (R),A FRACTION BAD.'<CRLF>
6136 051647  101  103  060                        .ASCIZ  'AC0 DID NOT GET 0 IN STATE 424.'
6137 051707                                EM57:
     051707  115  117  104                        .ASCII  'MODF (R),A INTEGER BAD.'<CRLF>
6138 051737  101  103  061                        .ASCIZ  'AC1 DID NOT GET 0 IN STATE 142.'
6139 051777                                EM60:
     051777  115  117  104                        .ASCII  'MODF (R),A INTEGER BAD.'<CRLF>
6140 052027  101  103  061                        .ASCIZ  'AC1 DID NOT GET THE INTEGER IN STATE 134.'
6141 052101                                EM61:
     052101  115  117  104                        .ASCII  'MODF (R),A FRACTION BAD.'<CRLF>
6142 052132  101  040  102                        .ASCII  'A BAD CONSTANT WAS USED (NOT 24) IN STATE 046.'
6143 052210  200  117  122                        .ASCIZ  <CRLF>'OR (BUT NBIT) STATE 525 WENT TO 050 INSTEAD OF 150.'
6144 052275                                EM62:
     052275  115  117  104                        .ASCII  'MODF (R),A FRACTION BAD.'<CRLF>
6145 052326  101  040  102                        .ASCII  'A BAD CONSTANT WAS USED (NOT 24) IN STATE 046.'
6146 052404  200  117  122                        .ASCIZ  <CRLF>'OR (BUT NBIT) STATE 525 WENT TO 150 INSTEAD OF 050.'
6147 052471                                EM63:
     052471  115  117  104                        .ASCII  'MODF (R),A FRACTION BAD.'<CRLF>
6148 052522  050  102  125                        .ASCIZ  '(BUT ZBT) STATE 532 WENT TO 10  INSTEAD OF 122.'
6149 052602                                EM64:
     052602  115  117  104                        .ASCII  'MODF (R),A FRACTION BAD.'<CRLF>
6150 052633  050  102  125                        .ASCIZ  '(BUT ENBT EZBT) STATE 041 WENT TO 046 INSTEAD OF 246.'
6151 052721                                EM65:
     052721  115  117  104                        .ASCII  'MODF (R),A FRACTION BAD.'<CRLF>
6152 052752  050  102  125                        .ASCIZ  '(BUT FT) STATE 126 SHOULD HAVE GONE TO 133. FT=0.'
6153 053034                                EM66:
     053034  115  117  104                        .ASCII  'MODF (R),A FRACTION BAD.'<CRLF>
6154 053065  123  111  107                        .ASCIZ  'SIGN BIT BAD.'
6155 053103                                EM67:
     053103  115  117  104                        .ASCII  'MODF (R),A INTEGER BAD.'<CRLF>
6156 053133  123  111  107                        .ASCIZ  'SIGN BIT BAD IN STATE 733.'
6157 053166  115  117  104  EM70:                 .ASCIZ  'MODD (R),A FRACTION BAD.'
6158 053217  115  117  104  EM71:                 .ASCIZ  'MODD (R),A INTEGER BAD.'
6159 053247  106  120  123  EM72:                 .ASCIZ  'FPS BAD AFTER MODD (R),A.'
6160         053247          EM73=EM72
6167 053301                                EM74:
     053301  115  117  104                        .ASCII  'MODD (R),A INTEGER BAD.'<CRLF>
6168 053331  050  102  125                        .ASCIZ  '(BUT FD) STATE 231 WENT TO 142 INSTEAD OF 143.'
6169 053410                                EM75:
     053410  115  117  104                        .ASCII  'MODD (R),A FRACTION BAD.'<CRLF>
6170 053441  101  103  060                        .ASCIZ  'AC0 GETS 0 IN STATE 425 FAILED.'
6171 053501                                EM76:
     053501  115  117  104                        .ASCII  'MODD (R),A INTEGER BAD.'<CRLF>
6172 053531  101  103  061                        .ASCIZ  'AC1 GETS 0 IN STATE 143 FAILED.'
6173 053571                                EM77:
     053571  115  117  104                        .ASCII  'MODD (R),A FRACTION BAD.'<CRLF>
```

```
6174 053622  050  102  125          .ASCIZ  '(BUT FD) STATE 526 WENT TO 134 INSTEAD OF 135.'
6175 053701                  EM100:
     053701  115  117  104          .ASCII  'MODD (R),A INTEGER BAD.'<CRLF>
6176 053731  123  111  107          .ASCIZ  'SIGN BIT BAD IN STATE 526.'
6177 053764                  EM101:
     053764  115  117  104          .ASCII  'MODD (R),A FRACTION BAD.'<CRLF>
6178 054015  101  040  102          .ASCII  'A BAD CONSTANT WAS USED (NOT 56) IN STATE 046,'
6179 054073  200  117  122          .ASCIZ  <CRLF>'OR (BUT NBIT) STATE 525 WENT TO 050 INSTEAD OF 150.'
6180 054160                  EM102:
     054160  115  117  104          .ASCII  'MODD (R),A FRACTION BAD.'<CRLF>
6181 054211  101  040  102          .ASCII  'A BAD CONSTANT WAS USED (NOT 56) IN STATE 046,'
6182 054267  200  117  122          .ASCIZ  <CRLF>'OR (BUT NBIT) STATE 525 WENT TO 150 INSTEAD OF 050.'
6183 054354                  EM103:
     054354  115  117  104          .ASCII  'MODD (R),A FRACTION BAD.'<CRLF>
6184 054405  050  102  125          .ASCIZ  '(BUT ZBIT) STATE 532 WENT TO 122 INSTEAD OF 102.'
6185 054466                  EM104:
     054466  115  117  104          .ASCII  'MODD (R),A INTEGER BAD.'<CRLF>
6186 054516  123  105  124          .ASCII  'SET INTEGER IN AC1 FAILED,'
6187 054550  200  117  122          .ASCIZ  <CRLF>'OR (BUT FD) STATE 733 WENT TO 156 INSTEAD OF 157.'
6188 054633                  EM105:
     054633  115  117  104          .ASCII  'MODD (R),A INTEGER BAD.'<CRLF>
6189 054663  050  102  125          .ASCIZ  '(BUT FD) STATE 122 WENT TO 424 INSTEAD OF 425.'
6190 054742                  EM106:
     054742  115  117  104          .ASCII  'MODD (R),A INTEGER BAD.'<CRLF>
6191 054772  050  102  125          .ASCIZ  '(BUT FD) STATE 246 WENT TO 126 INSTEAD OF 127.'
6192 055051                  EM107:
     055051  115  117  104          .ASCII  'MODD (R),A INTEGER BAD.'<CRLF>
6193 055101  050  102  125          .ASCIZ  '(BUT FD) STATE 446 WENT TO 126 INSTEAD OF 127.'
6194 055160                  EM110:
     055160  115  117  104          .ASCII  'MODD (R),A FRACTION BAD.'<CRLF>
6195 055211  050  102  125          .ASCIZ  '(BUT FT) STATE 127 WENT TO 313 INSTEAD OF 113.'
6208 055270                  EM165:
     055270  115  117  104          .ASCII  /MODF (R),A FRACTION BAD. RESULT OVER OR UNDER FLOW./
6209 055353  000                    .BYTE   0
6210 055354                  EM166:
     055354  115  117  104          .ASCII  /MODF (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6211 055436  000                    .BYTE   0
6212 055437                  EM167:
     055437  106  120  123          .ASCII  /FPS BAD AFTER MODF (R),A./
6213 055470  000                    .BYTE   0
6214 055471                  EM170:
     055471  106  105  103          .ASCII  /FEC BAD AFTER MODF (R),A./
6215 055522  105  130  120          .ASCIZ  'EXPECTING UNDERFLOW, FEC=12.'
6216 055557                  EM171:
     055557  115  117  104          .ASCII  /MODF (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6217 055641  200  101  103          .ASCIZ  <CRLF>'AC1 GETS 0 IN STATE 126 FAILED.'
6218 055702                  EM172:
     055702  106  105  103          .ASCII  /FEC BAD AFTER MODF (R),A./
6219 055733  105  130  120          .ASCIZ  'EXPECTING OVERFLOW, FEC=10.'
6220 055767                  EM173:
     055767  115  117  104          .ASCII  /MODF (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6221 056051  200  050  102          .ASCIZ  <CRLF>'(BUT FIV FD) STATE 520 WENT TO 142 INSTEAD OF 162.'
6222 056135                  EM174:
     056135  115  117  104          .ASCII  /MODF (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6223 056217  200  050  102          .ASCIZ  <CRLF>'(BUT FIV FD) STATE 520 WENT TO 162 INSTEAD OF 142.'
6224 056303                  EM175:
     056303  115  117  104          .ASCII  /MODF (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
```

```
6225 056365  200  123  111          .ASCIZ  <CRLF>'SIGN BAD IN STATE 517.'
6226 056415                  EM176:
     056415  115  117  104          .ASCII  /MODD (R),A FRACTION BAD. RESULT OVER OR UNDER FLOW./
6227 056500  000                    .BYTE   0
6228 056501                  EM177:
     056501  115  117  104          .ASCII  /MODD (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6229 056563  000                    .BYTE   0
6230 056564                  EM200:
     056564  106  120  123          .ASCII  /FPS BAD AFTER MODD (R),A./
6231 056615  000                    .BYTE   0
6232 056616                  EM201:
     056616  106  105  103          .ASCII  /FEC BAD AFTER MODD (R),A./
6233 056647  200  105  130          .ASCIZ  <CRLF>'EXPECTING UNDERFLOW, FEC=12.'
6234 056705                  EM202:
     056705  115  117  104          .ASCII  /MODD (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6235 056767  200  050  102          .ASCIZ  <CRLF>'(BUT FD) STATE 241 WENT TO 126 INSTEAD OF 127.'
6236 057047                  EM203:
     057047  115  117  104          .ASCII  /MODD (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6237 057131  200  050  102          .ASCIZ  <CRLF>'(BUT FD) STATE 047 WENT TO 126 INSTEAD OF 127.'
6238 057211                  EM204:
     057211  106  105  103          .ASCII  /FEC BAD AFTER MODD (R),A./
6239 057242  200  105  130          .ASCIZ  <CRLF>'EXPECTING OVERFLOW, FEC=10.'
6240 057277                  EM205:
     057277  115  117  104          .ASCII  /MODD (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6241 057361  200  050  102          .ASCIZ  <CRLF>'(BUT FIV FD) STATE 520 WENT TO 162 INSTEAD OF 163.'
6242 057445                  EM206:
     057445  115  117  104          .ASCII  /MODD (R),A INTEGER BAD. RESULT OVER OR UNDER FLOW./
6243 057527  200  050  102          .ASCIZ  <CRLF>'(BUT FIV FD) STATE 520 WENT TO 162 INSTEAD OF 143.'
6244
6245                         ;*
6338 057613                  EM207:
     057613  101  104  104          .ASCIZ  /ADDD (R),A PRODUCED A BAD RESULT./
6339 057655                  EM210:
     057655  124  110  105          .ASCIZ  /THE FPS WAS BAD AFTER ADDD (R),A./
6340 057717  101  104  104  EM211:  .ASCII  'ADDD (R),A FAILED IN THE ROUND\TRUNK FLOWS.'
6341 057772  200  127  105          .ASCII  <CRLF>'WENT FROM STATE 663 TO 313,'<CRLF>
6342 060027  111  116  123          .ASCIZ  'INSTEAD OF FROM 663 TO 353.'
6367 060063                  EM212:
     060063  101  104  104          .ASCII  'ADDD (R),A FAILED IN THE ROUND\TRUNK FLOWS.'
     060136  200  124  110          .ASCII  <CRLF>'THE FPS WAS BAD.'<CRLF>
     060160  104  111  104          .ASCII  'DID NOT TAKE THE PATH:'<CRLF>
     060207  106  122  117          .ASCIZ  /FROM STATE 664, TO 505, TO 251./
6368 060247                  EM213:
     060247  101  104  104          .ASCII  'ADDD (R),A FAILED IN THE ROUND\TRUNK FLOWS.'
     060322  200  124  110          .ASCII  <CRLF>'THE FPS WAS BAD.'<CRLF>
     060344  104  111  104          .ASCII  'DID NOT TAKE THE PATH:'<CRLF>
     060373  106  122  117          .ASCIZ  /FROM STATE 664, TO 505, TO 253./
6369 060433                  EM214:
     060433  101  104  104          .ASCII  'ADDD (R),A FAILED IN THE ROUND\TRUNK FLOWS.'
     060506  200  124  110          .ASCII  <CRLF>'THE FPS WAS BAD.'<CRLF>
     060530  104  111  104          .ASCII  'DID NOT TAKE THE PATH:'<CRLF>
     060557  106  122  117          .ASCIZ  /FROM STATE 664, TO 705, TO 735./
6370 060617                  EM215:
     060617  101  104  104          .ASCII  'ADDD (R),A FAILED IN THE ROUND\TRUNK FLOWS.'
     060672  200  124  110          .ASCII  <CRLF>'THE FPS WAS BAD.'<CRLF>
     060714  104  111  104          .ASCII  'DID NOT TAKE THE PATH:'<CRLF>
     060743  106  122  117          .ASCIZ  /FROM STATE 664, TO 705, TO 737./
```

```
 6371 061003  124  110  105  EM216:  .ASCII  'THE (BUT FIU FORK IN THE OVER\UNDER FLOWS FAILED. FIU =1.'
 6372 061074  200  127  105           .ASCII  <CRLF>'WENT FROM STATE 331 TO 115.'<CRLF>
 6373 061131  111  116  123           .ASCIZ  'INSTEAD OF FROM 331 TO 155.'
 6374 061165  101  104  104  EM217:  .ASCIZ  'ADDD (R)A TRAPPED TO 244,. FID=1.'
 6375 061227                EM220:
      061227  101  104  104           .ASCII  /ADDD (R),A TRAPPED TO 244./<CRLF>
      061262  124  110  105           .ASCII  'THE RESULT WAS AN OVERFLOW CONDITION BUT FIV= 0.'
      061342  200  050  102           .ASCIZ  <CRLF>/(BUT FIV) STATE 133 WENT TO 136 INSTEAD OF 116./
      061423  111  116  123           .ASCIZ  /INSTEAD OF FROM 133 TO 116./
 6376 061457                EM221:
      061457  101  104  104           .ASCII  /ADDD (R),A FAILED TO TRAP TO 244./<CRLF>
      061521  124  110  105           .ASCII  'THE RESULT WAS A OVERFLOW CONDITION AND FIV=1.'<CRLF>
      061600  124  110  105           .ASCII  /THE (BUT FIV) FORK FAILED./<CRLF>
      061633  127  105  116           .ASCII  /WENT FROM STATE 133 TO 116./<CRLF>
      061667  111  116  123           .ASCIZ  /INSTEAD OF FROM 133 TO 136./
 6377 061723                EM222:
      061723  101  104  104           .ASCII  /ADDD (R),A TRAPPED TO 244./<CRLF>
      061756  124  110  105           .ASCII  'THE RESULT WAS AN UNDERFLOW CONDITION BUT FIU= 0.'
      062037  200  050  102           .ASCIZ  <CRLF>/(BUT FIU) STATE 331 WENT TO 155 INSTEAD OF 115./
      062120  111  116  123           .ASCIZ  /INSTEAD OF FROM 331 TO 115./
 6378 062154                EM223:
      062154  101  104  104           .ASCII  /ADDD (R),A FAILED TO TRAP TO 244./<CRLF>
      062216  124  110  105           .ASCII  'THE RESULT WAS A UNDERFLOW CONDITION AND FIU=1.'<CRLF>
      062276  124  110  105           .ASCII  /THE (BUT FIU) FORK FAILED./<CRLF>
      062331  127  105  116           .ASCII  /WENT FROM STATE 331 TO 115./<CRLF>
      062365  111  116  123           .ASCIZ  /INSTEAD OF FROM 331 TO 155./
 6379 062421                EM224:
      062421  101  104  104           .ASCII  /ADDD (R),A TRAPPED TO 244./<CRLF>
      062454  124  110  105           .ASCII  'THE RESULT WAS AN UNDERFLOW CONDITION BUT FIU= 0.'
      062535  200  050  102           .ASCIZ  <CRLF>/(BUT FIU) STATE 137 WENT TO 155 INSTEAD OF 115./
      062616  111  116  123           .ASCIZ  /INSTEAD OF FROM 137 TO 115./
 6380 062652                EM225:
      062652  101  104  104           .ASCII  /ADDD (R),A FAILED TO TRAP TO 244./<CRLF>
      062714  124  110  105           .ASCII  'THE RESULT WAS A UNDERFLOW CONDITION AND FIU=1.'<CRLF>
      062774  124  110  105           .ASCII  /THE (BUT FIU) FORK FAILED./<CRLF>
      063027  127  105  116           .ASCII  /WENT FROM STATE 137 TO 115./<CRLF>
      063063  111  116  123           .ASCIZ  /INSTEAD OF FROM 137 TO 155./
 6381 063117                EM226:
      063117  101  104  104           .ASCII  /ADDD (R),A TRAPPED TO 244./
      063151  200  102  105           .ASCII  <CRLF>'BECAUSE OF AN EXPECTED OVERFLOW CONDITION,'<CRLF>
      063225  102  125  124           .ASCIZ  'BUT THE FEC WAS BAD.'
 6382 063252                EM227:
      063252  101  104  104           .ASCII  /ADDD (R),A TRAPPED TO 244./
      063304  200  102  105           .ASCII  <CRLF>'BECAUSE OF AN EXPECTED OVERFLOW CONDITION,'<CRLF>
      063360  102  125  124           .ASCIZ  'BUT THE FPS WAS BAD.'
 6383 063405                EM230:
      063405  101  104  104           .ASCII  /ADDD (R),A TRAPPED TO 244./
      063437  200  102  105           .ASCII  <CRLF>'BECAUSE OF AN EXPECTED UNDERFLOW CONDITION,'<CRLF>
      063514  102  125  124           .ASCIZ  'BUT THE FEC WAS BAD.'
 6384 063541                EM231:
      063541  101  104  104           .ASCII  /ADDD (R),A TRAPPED TO 244./
      063573  200  102  105           .ASCII  <CRLF>'BECAUSE OF AN EXPECTED UNDERFLOW CONDITION,'<CRLF>
      063650  102  125  124           .ASCIZ  'BUT THE FPS WAS BAD.'
 6385         057655                EM232=EM210
 6386
 6387
 6388
```

```
 6393
 6397
 6404
 6414
 6415 063675                                    EM233:
      063675  114     104     103                       .ASCIZ  \LDCFD (R)+,A RESULT INCORRECT.\
 6416 063734                                    EM234:
      063734  122     060     040                       .ASCII  \R0 BAD AFTER LDCFD (R)+,A.\
      063766  200     101     040                       .ASCIZ  <CRLF>'A BAD CONSTANT WAS USED.'
 6417 064020                                    EM235:
      064020  120     103     040                       .ASCIZ  \PC BAD AFTER LDCFD #NUM,A. TRAP TO 4.\
 6418 064066                                    EM236:
      064066  120     103     040                       .ASCIZ  \PC BAD AFTER LDCFD #NUM,A.\
 6419 064121                                    EM237:
      064121  122     060     040                       .ASCII  \R0 BAD AFTER LDCDF (R)+,A.\
      064153  200     101     040                       .ASCIZ  <CRLF>'A BAD CONSTANT WAS USED.'
 6420 064205                                    EM240:
      064205  106     120     123                       .ASCIZ  \FPS BAD AFTER LDCFD (R)+,A.\
 6421 064241                                    EM241:
      064241  114     104     103                       .ASCII  \LDCFD (R)+,A FAILED.\
      064265  200     124     110                       .ASCII  <CRLF>'THE FD '
      064275  102     111     124                       .ASCII  'BIT WAS NOT COMPLIMENTED '
      064326  111     116     040                       .ASCIZ  'IN STATE 017.'
 6422 064344                                    EM242:
      064344  106     120     123                       .ASCIZ  \FPS BAD AFTER LDCDF (R)+,A.\
 6423 064400                                    EM243:
      064400  114     104     103                       .ASCII  \LDCDF (R)+,A FAILED.\
      064424  200     124     110                       .ASCII  <CRLF>'THE FD '
      064434  102     111     124                       .ASCII  'BIT WAS NOT COMPLIMENTED '
      064465  111     116     040                       .ASCIZ  'IN STATE 017.'
 6424 064503                                    EM244:
      064503  114     104     103                       .ASCIZ  \LDCDF (R)+,A RESULT INCORRECT.\
 6425
 6426 064542  114     104     103  EM245:               .ASCII  'LDCFD (R),A FAILED.'
 6427 064565  200     123     105                       .ASCII  <CRLF>'SET SIGN FAILED '
 6428 064606  111     116     040                       .ASCIZ  'IN STATE 512.'
 6429 064624  125     116     105  EM247:               .ASCIZ  'UNEXPECTED FPP TRAP TO 244.'
 6430 064660  125     116     105  EM250:               .ASCIZ  'UNEXPECTED CPU TRAP TO 4.'
 6431 064712  125     116     105  EM251:               .ASCIZ  'UNEXPECTED CPU TRAP TO 10.'
 6432
 6433 064745  103     117     122  EM252:               .ASCIZ  'CORRECT FLOWS INTERRUPT TEST MODULE FAILED TO INTERRUPT.'
 6434
 6438
 6439 065036                                    EM253:
      065036  101     104     104                       .ASCIZ  /ADDD AC0,AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
 6440 065122                                    EM254:
      065122  101     104     104                       .ASCIZ  /ADDD (R0),AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
 6441 065207                                    EM255:
      065207  101     104     104                       .ASCIZ  /ADDD (R0)+,AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
 6442 065275                                    EM256:
      065275  101     104     104                       .ASCIZ  /ADDD @(R0)+,AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
 6443 065364                                    EM257:
      065364  101     104     104                       .ASCIZ  /ADDD -(R0),AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
 6444 065452                                    EM260:
      065452  101     104     104                       .ASCIZ  /ADDD @-(R0),AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
 6445 065541                                    EM261:
      065541  101     104     104                       .ASCIZ  /ADDD NUM(R0),AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
```

```
6446 065631                          EM262:
     065631    101   104   104               .ASCIZ  /ADDD @NUM(R0),AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
6447 065722                          EM263:
     065722    104   111   126               .ASCIZ  /DIVD (R0),AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
6448 066007                          EM264:
     066007    115   125   114               .ASCIZ  /MULD (R0),AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
6449 066074                          EM265:
     066074    115   117   104               .ASCIZ  /MODD (R0),AC0 FAILED IN THE INTERRUPT CORRECT FLOWS./
6450
6451 066161    103   117   122   EM266:      .ASCIZ  'CORRECT FLOWS INTERRUPT TEST MODULE CAUSED UNEXPECTED INTERRUPT.'
6452
6453                                ;THESE ARE ERROR DATA TABLE HEADERS:
6454
6455 066262    040   040   124   DH1:        .ASCII  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR AT PC.'<TAB>
6456 066323    107   117   124               .ASCIZ  'GOT FPS.'<TAB>'EXPECTED FPS.'
6457 066352    040   040   124   DH2:        .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'<TAB>'ERROR AT PC.'
6458          066262             DH3=DH1
6459          066262             DH4=DH1
6460          066262             DH5=DH1
6461          066262             DH6=DH1
6462          066262             DH7=DH1
6463          066262             DH10=DH1
6464          066262             DH11=DH1
6465          066262             DH12=DH1
6466          066262             DH13=DH1
6467          066262             DH14=DH1
6468          066262             DH15=DH1
6469          066262             DH16=DH1
6470 066413    040   040   124   DH17:       .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>'FEC.'<TAB>'FPS.'
6471          066262             DH20=DH1
6472 066466    040   040   124   DH21:       .ASCII  '  TEST.'<TAB>'PC OF CALL'<TAB>'PC OF ERROR.'<TAB>
6473 066526    107   117   124               .ASCIZ  'GOT FEC.'<TAB>'EXPECTED FEC.'
6474 066555    040   040   124   DH22:       .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>'FPS.'
6475          066262             DH23=DH1
6476          066262             DH32=DH1
6477          066262             DH24=DH1
6478          066262             DH25=DH1
6479          066262             DH26=DH1
6480          066262             DH27=DH1
6481          066262             DH30=DH1
6482          066262             DH31=DH1
6483          066262             DH33=DH1
6484          066262             DH34=DH1
6485          066262             DH35=DH1
6486          066262             DH36=DH1
6487          066262             DH37=DH1
6488          066262             DH40=DH1
6489          066262             DH41=DH1
6490          066262             DH42=DH1
6491          066262             DH43=DH1
6492          066262             DH44=DH1
6493          066262             DH45=DH1
6494          066262             DH246=DH1
6495          066262             DH46=DH1
6496          066262             DH47=DH1
6497          066262             DH50=DH1
6498          066262             DH51=DH1
```

```
 6499            066262                  DH52=DH1
 6500            066262                  DH111=DH1
 6501            066262                  DH112=DH1
 6502            066262                  DH113=DH1
 6503            066262                  DH114=DH1
 6504 066623      040      040     124   DH115:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>
 6505 066664      107      117     124           .ASCIZ  'GOT FEC.''GOT FPS.''EXPECTED FPS.'
 6506            066623                  DH116=DH115
 6507            066623                  DH117=DH115
 6508            066623                  DH120=DH115
 6509            066623                  DH121=DH115
 6510            066623                  DH122=DH115
 6511            066262                  DH123=DH1
 6512            066262                  DH124=DH1
 6513            066262                  DH125=DH1
 6514            066262                  DH126=DH1
 6515            066623                  DH127=DH115
 6516            066623                  DH130=DH115
 6517            066623                  DH131=DH115
 6518            066262                  DH132=DH1
 6519            066623                  DH133=DH115
 6520            066623                  DH134=DH115
 6521            066262                  DH135=DH1
 6522            066623                  DH136=DH115
 6523            066623                  DH137=DH115
 6524            066623                  DH140=DH115
 6525            066623                  DH141=DH115
 6526            066623                  DH142=DH115
 6527            066623                  DH143=DH115
 6528            066623                  DH144=DH115
 6529            066262                  DH145=DH1
 6530            066262                  DH146=DH1
 6531            066262                  DH147=DH1
 6532            066262                  DH150=DH1
 6533            066623                  DH151=DH115
 6534            066623                  DH152=DH115
 6535            066623                  DH153=DH115
 6536            066623                  DH154=DH115
 6537            066623                  DH155=DH115
 6538            066623                  DH156=DH115
 6539            066262                  DH157=DH1
 6540            066623                  DH160=DH115
 6541            066262                  DH161=DH1
 6542            066262                  DH162=DH1
 6543            066623                  DH163=DH115
 6544            066262                  DH164=DH1
 6545            066262                  DH53=DH1
 6546            066262                  DH54=DH1
 6547            066262                  DH55=DH1
 6548            066262                  DH56=DH1
 6549            066262                  DH57=DH1
 6550            066262                  DH60=DH1
 6551            066262                  DH61=DH1
 6552            066262                  DH62=DH1
 6553            066262                  DH63=DH1
 6554            066262                  DH64=DH1
 6555            066262                  DH65=DH1
```

```
 6556          066262                  DH66=DH1
 6557          066262                  DH67=DH1
 6558          066262                  DH70=DH1
 6559          066262                  DH71=DH1
 6560          066262                  DH72=DH1
 6561          066262                  DH73=DH1
 6562          066262                  DH74=DH1
```

```
 6564            066262                  DH75=DH1
 6565            066262                  DH76=DH1
 6566            066262                  DH77=DH1
 6567            066262                  DH100=DH1
 6568            066262                  DH101=DH1
 6569            066262                  DH102=DH1
 6570            066262                  DH103=DH1
 6571            066262                  DH104=DH1
 6572            066262                  DH105=DH1
 6573            066262                  DH106=DH1
 6574            066262                  DH107=DH1
 6575            066262                  DH110=DH1
 6576            066623                  DH165=DH115
 6577            066623                  DH166=DH115
 6578            066623                  DH167=DH115
 6579            066623                  DH170=DH115
 6580            066623                  DH171=DH115
 6581            066623                  DH172=DH115
 6582            066623                  DH173=DH115
 6583            066623                  DH174=DH115
 6584            066623                  DH175=DH115
 6585            066623                  DH176=DH115
 6586            066623                  DH177=DH115
 6587            066623                  DH200=DH115
 6588            066623                  DH201=DH115
 6589            066623                  DH202=DH115
 6590            066623                  DH203=DH115
 6591            066623                  DH204=DH115
 6592            066623                  DH205=DH115
 6593            066623                  DH206=DH115
 6594 066722      040     040     124    DH220:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
 6595 066762      040     040     124    DH207:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 6596 067022      011     107     117            .ASCIZ  <TAB>'GOT FPS.'<TAB>'EXPECTED FPS.'
 6597 067052      040     040     124    DH210:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 6598            067052                  DH211=DH210
 6599            066762                  DH212=DH207
 6600            066762                  DH213=DH207
 6601            066762                  DH214=DH207
 6602            066762                  DH215=DH207
 6603            067052                  DH216=DH210
 6604 067113      040     040     124    DH217:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 6605 067153      011     106     120            .ASCIZ  <TAB>'FPS.'<TAB>'FEC.'
 6606            067052                  DH221=DH210
 6607            066722                  DH222=DH220
 6608            067052                  DH223=DH210
 6609            066722                  DH224=DH220
 6610            067052                  DH225=DH210
 6611 067166      040     040     124    DH226:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
 6612 067225      011     107     117            .ASCIZ  <TAB>'GOT FEC.'<TAB>'EXPECTED FEC.'
 6613 067255      040     040     124    DH227:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
 6614 067314      011     107     117            .ASCIZ  <TAB>'GOT FPS.'<TAB>'EXPECTED FPS.'
 6615            067166                  DH230=DH226
 6616            067255                  DH231=DH227
 6617            067255                  DH232=DH227
 6618
 6619
 6620 067344      040     040     124    DH233:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
```

```
                                                                
 6621                                                           
 6622 067405     040     040     124  DH234:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 6623 067445     011     107     117          .ASCIZ  <TAB>'GOT R0.'<TAB>'EXPECTED R0.'
 6624                                                           
 6625 067473     040     040     124  DH235:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>
 6626 067517     120     103     040          .ASCII  'PC OF TRAP.'<TAB>
 6627                                                           
 6628 067533     040     040     124  DH236:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>
 6629 067557     120     103     040          .ASCII  'PC OF ERROR.'<TAB>'GOT PC.'
 6630 067603     011     105     130          .ASCIZ  <TAB>'EXPECTED PC.'
 6631                                                           
 6632         067405                  DH237=DH234               
 6633                                                           
 6634 067621     040     040     124  DH240:  .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>
 6635 067645     120     103     040          .ASCII  'PC OF ERROR.'<TAB>
 6636 067662     107     117     124          .ASCIZ  'GOT FPS.'<TAB>'EXPECTED FPS.'
 6637                                                           
 6638         067344                  DH241=DH233               
 6639         067621                  DH242=DH240               
 6640         067344                  DH243=DH233               
 6641         067344                  DH244=DH233               
 6642         067344                  DH245=DH233               
 6643 067711     040     040     124  DH247:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'<TAB>'FEC.'
 6644 067756     040     040     124  DH250:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
 6645         067756                  DH251=DH250               
 6646                                                           
 6647         066352                  DH252=DH2                 
 6648         066262                  DH253=DH1                 
 6649         066262                  DH254=DH1                 
 6650         066262                  DH255=DH1                 
 6651         066262                  DH256=DH1                 
 6652         066262                  DH257=DH1                 
 6653         066262                  DH260=DH1                 
 6654         066262                  DH261=DH1                 
 6655         066262                  DH262=DH1                 
 6656         066262                  DH263=DH1                 
 6657         066262                  DH264=DH1                 
 6658         066262                  DH265=DH1                 
 6659 070016     040     040     124  DH266:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
 6660                                                           
 6661                                                           
 6662                                  ;THESE ARE THE DATA FORMAT SPECIFIERS FOR THE DATA TABLE:
 6663 070056     004     000     005  DF1:    .BYTE   4,0,5,0,5,0,5,0,5,5,3,5,5,3
 6664 070074     004     000     005  DF2:    .BYTE   4,0,5,0,5,5,3,5,5,3
 6665         070056                  DF3=DF1                   
 6666         070056                  DF4=DF1                   
 6667         070056                  DF5=DF1                   
 6668         070056                  DF6=DF1                   
 6669         070056                  DF7=DF1                   
 6670         070056                  DF10=DF1                  
 6671         070056                  DF11=DF1                  
 6672         070056                  DF12=DF1                  
 6673         070056                  DF13=DF1                  
 6674         070056                  DF14=DF1                  
 6675         070056                  DF15=DF1                  
 6676         070056                  DF16=DF1                  
 6677 070106     004     000     005  DF17:   .BYTE   4,0,5,0,5,0,0
```

```
 6678 070115      004      000      005  DF20:    .BYTE   4,0,5,0,5,0,5,0
 6679           070115                    DF21=DF20
 6680           070115                    DF22=DF20
 6681 070125      004      000      005  DF23:    .BYTE   4,0,5,0,5,0,5,0,5,5,2,5,5,2,5,5,2,5,5,2
 6682           070125                    DF24=DF23
 6683           070125                    DF32=DF23
 6684           070125                    DF25=DF23
 6685           070125                    DF26=DF23
 6686           070125                    DF27=DF23
 6687           070125                    DF30=DF23
 6688           070125                    DF31=DF23
 6689 070151      004      000      005  DF33:    .BYTE   4,0,5,0,5,0,5,0,5,5,3,5,5,3,5,5,3,5,5,3
 6690           070151                    DF34=DF33
 6691           070151                    DF35=DF33
 6692           070151                    DF36=DF33
 6693           070125                    DF37=DF23
 6694           070125                    DF40=DF23
 6695           070125                    DF41=DF23
 6696           070125                    DF42=DF23
 6697           070125                    DF43=DF23
 6698           070125                    DF44=DF23
 6699           070125                    DF45=DF23
 6700           070151                    DF246=DF33
 6701           070151                    DF46=DF33
 6702           070151                    DF47=DF33
 6703           070151                    DF50=DF33
 6704           070151                    DF51=DF33
 6705           070151                    DF52=DF33
 6706           070125                    DF111=DF23
 6707           070125                    DF112=DF23
 6708           070125                    DF113=DF23
 6709           070125                    DF114=DF23
 6710 070175      004      000      005  DF115:   .BYTE   4,0,5,0,5,0,0,0,5,5,2,5,5,2,5,5,2,5,5,2
 6711           070175                    DF116=DF115
 6712           070175                    DF117=DF115
 6713           070175                    DF120=DF115
 6714           070175                    DF121=DF115
 6715           070175                    DF122=DF115
 6716           070151                    DF123=DF33
 6717           070151                    DF124=DF33
 6718           070151                    DF125=DF33
 6719           070151                    DF126=DF33
 6720 070221      004      000      005  DF127:   .BYTE   4,0,5,0,5,0,0,0,5,5,3,5,5,3,5,5,3,5,5,3
 6721           070221                    DF130=DF127
 6722           070221                    DF131=DF127
 6723           070151                    DF132=DF33
 6724           070221                    DF133=DF127
 6725           070221                    DF134=DF127
 6726           070151                    DF135=DF33
 6727           070221                    DF136=DF127
 6728           070175                    DF137=DF115
 6729           070175                    DF140=DF115
 6730           070175                    DF141=DF115
 6731           070175                    DF142=DF115
 6732           070175                    DF143=DF115
 6733           070175                    DF144=DF115
 6734           070125                    DF145=DF23
```

```
 6735            070125                  DF146=DF23
 6736            070125                  DF147=DF23
 6737            070125                  DF150=DF23
 6738            070221                  DF151=DF127
 6739            070221                  DF152=DF127
 6740            070221                  DF153=DF127
 6741            070221                  DF154=DF127
 6742            070221                  DF155=DF127
 6743            070221                  DF156=DF127
 6744            070151                  DF157=DF33
 6745            070221                  DF160=DF127
 6746            070151                  DF161=DF33
 6747            070151                  DF162=DF33
 6748            070221                  DF163=DF127
 6749            070151                  DF164=DF33
 6750 070245       004       000   005   DF53:   .BYTE   4,0,5,0,5,0,5,0,5,5,2,5,5,2,5,5,2,5,5,2,5,5,2,5,5,2
 6751            070245                  DF54=DF53
 6752            070245                  DF55=DF53
 6753            070245                  DF56=DF53
 6754            070245                  DF57=DF53
 6755            070245                  DF60=DF53
 6756            070245                  DF61=DF53
 6757            070245                  DF62=DF53
 6758            070245                  DF63=DF53
 6759            070245                  DF64=DF53
 6760            070245                  DF65=DF53
 6761            070245                  DF66=DF53
 6762            070245                  DF67=DF53
 6763 070277       004       000   005   DF70:   .BYTE   4,0,5,0,5,0,5,0,5,5,3,5,5,3,5,5,3,5,5,3,5,5,3,5,5,3
 6764            070277                  DF71=DF70
 6765            070277                  DF72=DF70
 6766            070277                  DF73=DF70
 6767            070277                  DF74=DF70
 6768            070277                  DF75=DF70
 6769            070277                  DF76=DF70
 6770            070277                  DF77=DF70
 6771            070277                  DF100=DF70
 6772            070277                  DF101=DF70
 6773            070277                  DF102=DF70
 6774            070277                  DF103=DF70
 6775            070277                  DF104=DF70
 6776            070277                  DF105=DF70
 6777            070277                  DF106=DF70
 6778            070277                  DF107=DF70
 6779            070277                  DF110=DF70
 6780 070331       004       000   005   DF165:  .BYTE   4,0,5,0,5,0,0,0,5,5,2,5,5,2,5,5,2,5,5,2,5,5,2,5,5,2
 6781            070331                  DF166=DF165
 6782            070331                  DF167=DF165
 6783            070331                  DF170=DF165
 6784            070331                  DF171=DF165
 6785            070331                  DF172=DF165
 6786            070331                  DF173=DF165
 6787            070331                  DF174=DF165
 6788            070331                  DF175=DF165
 6789 070363       004       000   005   DF176:  .BYTE   4,0,5,0,5,0,0,0,5,5,3,5,5,3,5,5,3,5,5,3,5,5,3,5,5,3
 6790            070363                  DF177=DF176
 6791            070363                  DF200=DF176
```

```
 6792            070363                  DF201=DF176
 6793            070363                  DF202=DF176
 6794            070363                  DF203=DF176
 6795            070363                  DF204=DF176
 6796            070363                  DF205=DF176
 6797            070363                  DF206=DF176
 6798 070415        004      000    005  DF210:  .BYTE   4,0,5,0,5,0,5,0
 6799 070425        004      000    005  DF207:  .BYTE   4,0,5,0,5,5,5,3,5,5,5,3,5,5,5,3,5,5,5,3
 6800            070425                  DF211=DF207
 6801 070451        004      000    005  DF212:  .BYTE   4,0,5,0,5,0,5,0,5,5,5,3,5,5,5,3,5,5,5,3,5,5,5,3
 6802            070451                  DF213=DF212
 6803            070451                  DF214=DF212
 6804            070451                  DF215=DF212
 6805            070425                  DF216=DF207
 6806 070501        004      000    005  DF217:  .BYTE   4,0,5,0,5,0,0
 6807            070425                  DF220=DF207
 6808            070425                  DF221=DF207
 6809            070425                  DF222=DF207
 6810            070425                  DF223=DF207
 6811            070425                  DF224=DF207
 6812            070425                  DF225=DF207
 6813            070451                  DF226=DF212
 6814            070451                  DF227=DF212
 6815            070451                  DF230=DF212
 6816            070451                  DF231=DF212
 6817            070451                  DF232=DF212
 6818 070510        004      000    005  DF233:  .BYTE   4,0,5,0,5,5,3,5,5,3,5,5,3
 6819 070525        004      000    005  DF234:  .BYTE   4,0,5,0,5,0,0
 6820
 6821 070534        004      000    005  DF235:  .BYTE   4,0,5,0
 6822            070525                  DF236=DF234
 6823            070525                  DF237=DF234
 6824            070525                  DF240=DF234
 6825            070510                  DF241=DF233
 6826            070525                  DF242=DF234
 6827            070510                  DF243=DF233
 6828            070510                  DF244=DF233
 6829            070510                  DF245=DF233
 6830 070540        004      000    005  DF247:  .BYTE   4,0,5,0,5,0
 6831            070540                  DF250=DF247
 6832            070540                  DF251=DF247
 6833
 6834 070546        004      000    005  DF252:  .BYTE   4,0,5,0
 6835 070552        004      000    005  DF253:  .BYTE   4,0,5,0,5,0,5,0,5,5,0,5,5,0,5,5,0,5,5,0,5,5,0,5,5,3,5,5,3
 6836            070552                  DF254=DF253
 6837            070552                  DF255=DF253
 6838            070552                  DF256=DF253
 6839            070552                  DF257=DF253
 6840            070552                  DF260=DF253
 6841            070552                  DF261=DF253
 6842            070552                  DF262=DF253
 6843            070552                  DF263=DF253
 6844            070552                  DF264=DF253
 6845            070552                  DF265=DF253
 6846            070546                  DF266=DF252
 6847
 6848
```

```
                                        .EVEN

                                ;THESE ARE ERROR DATA TABLES (FORMATTED ABOVE):
070604  001232  001234  037215  DT1:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP5,$TAB,$TMP6
070624  001313  037222  001240          .WORD   $CRLF,MS1,$TMP3,$CRLF,MS2,$TMP4,0
070642  001232  001234  037215  DT2:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS3,$TMP3,$CRLF,MS4,$TMP4,0
        070604                  DT3=DT1
        070604                  DT4=DT1
        070604                  DT5=DT1
        070604                  DT6=DT1
        070604                  DT7=DT1
        070604                  DT10=DT1
        070604                  DT11=DT1
        070604                  DT12=DT1
        070604                  DT13=DT1
        070604                  DT14=DT1
        070604                  DT15=DT1
        070604                  DT16=DT1
070670  001232  001234  037215  DT17:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TMP5,0
070710  001232  001234  037215  DT20:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TAB,$TMP5,0
070732  001232  001234  037215  DT21:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TAB,$TMP3,0
070754  001232  001234  037215  DT22:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP5,0
070772  001232  001234  037215  DT23:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TAB,$TMP10,$CRLF,MS1,$TMP3,$CRLF,MS2,$TMP
071026  001313  037335  001244          .WORD   $CRLF,MS5,$TMP5,$CRLF,MS6,$TMP6,0
        070772                  DT32=DT23
        070772                  DT24=DT23
        070772                  DT25=DT23
        070772                  DT26=DT23
        070772                  DT27=DT23
        070772                  DT30=DT23
        070772                  DT31=DT23
        070772                  DT33=DT23
        070772                  DT34=DT23
        070772                  DT35=DT23
        070772                  DT36=DT23
        070772                  DT37=DT23
        070772                  DT40=DT23
        070772                  DT41=DT23
        070772                  DT42=DT23
        070772                  DT43=DT23
        070772                  DT44=DT23
        070772                  DT45=DT23
        070772                  DT246=DT23
        070772                  DT46=DT23
        070772                  DT47=DT23
        070772                  DT50=DT23
        070772                  DT51=DT23
        070772                  DT52=DT23
        070772                  DT111=DT23
        070772                  DT112=DT23
        070772                  DT113=DT23
        070772                  DT114=DT23
071044  001232  001234  037215  DT115:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP11,$TMP7,$TAB,$TMP10
071066  001313  037222  001240          .WORD   $CRLF,MS1,$TMP3,$CRLF,MS2,$TMP4
071102  001313  037335  001244          .WORD   $CRLF,MS5,$TMP5,$CRLF,MS6,$TMP6,0
        071044                  DT116=DT115
```

```
 6906            071044                  DT117=DT115
 6907            071044                  DT120=DT115
 6908            071044                  DT121=DT115
 6909            071044                  DT122=DT115
 6910            070772                  DT123=DT23
 6911            070772                  DT124=DT23
 6912            070772                  DT125=DT23
 6913            070772                  DT126=DT23
 6914            071044                  DT127=DT115
 6915            071044                  DT130=DT115
 6916            071044                  DT131=DT115
 6917            070772                  DT132=DT23
 6918            071044                  DT133=DT115
 6919            071044                  DT134=DT115
 6920            070772                  DT135=DT23
 6921            071044                  DT136=DT115
 6922            071044                  DT137=DT115
 6923            071044                  DT140=DT115
 6924            071044                  DT141=DT115
 6925            071044                  DT142=DT115
 6926            071044                  DT143=DT115
 6927            071044                  DT144=DT115
 6928            070772                  DT145=DT23
 6929            070772                  DT146=DT23
 6930            070772                  DT147=DT23
 6931            070772                  DT150=DT23
 6932            071044                  DT151=DT115
 6933            071044                  DT152=DT115
 6934            071044                  DT153=DT115
 6935            071044                  DT154=DT115
 6936            071044                  DT155=DT115
 6937            071044                  DT156=DT115
 6938            070772                  DT157=DT23
 6939            071044                  DT160=DT115
 6940            070772                  DT161=DT23
 6941            070772                  DT162=DT23
 6942            071044                  DT163=DT115
 6943            070772                  DT164=DT23
 6944 071120    001232  001234  037215  DT53:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP11,$TAB,$TMP12
 6945 071140    001313  037222  001240          .WORD   $CRLF,MS1,$TMP3,$CRLF,MS2,$TMP4
 6946 071154    001313  037443  001244          .WORD   $CRLF,MS11,$TMP5,$CRLF,MS12,$TMP6
 6947 071170    001313  037375  001250          .WORD   $CRLF,MS7,$TMP7,$CRLF,MS10,$TMP10,0
 6948            071120                  DT54=DT53
 6949            071120                  DT55=DT53
 6950            071120                  DT56=DT53
 6951            071120                  DT57=DT53
 6952            071120                  DT60=DT53
 6953            071120                  DT61=DT53
 6954            071120                  DT62=DT53
 6955            071120                  DT63=DT53
 6956            071120                  DT64=DT53
 6957            071120                  DT65=DT53
 6958            071120                  DT66=DT53
 6959            071120                  DT67=DT53
 6960            071120                  DT70=DT53
 6961            071120                  DT71=DT53
 6962            071120                  DT72=DT53
```

```
 6963            071120                  DT73=DT53
 6964            071120                  DT74=DT53
 6965            071120                  DT75=DT53
 6966            071120                  DT76=DT53
 6967            071120                  DT77=DT53
 6968            071120                  DT100=DT53
 6969            071120                  DT101=DT53
 6970            071120                  DT102=DT53
 6971            071120                  DT103=DT53
 6972            071120                  DT104=DT53
 6973            071120                  DT105=DT53
 6974            071120                  DT106=DT53
 6975            071120                  DT107=DT53
 6976            071120                  DT110=DT53
 6977 071206    001232  001234  037215  DT165:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP13,$TMP11,$TMP12
 6978 071226    001313  037222  001240          .WORD   $CRLF,MS1,$TMP3,$CRLF,MS2,$TMP4
 6979 071242    001313  037443  001244          .WORD   $CRLF,MS11,$TMP5,$CRLF,MS12,$TMP6
 6980 071256    001313  037375  001250          .WORD   $CRLF,MS7,$TMP7,$CRLF,MS10,$TMP10,0
 6981            071206                  DT166=DT165
 6982            071206                  DT167=DT165
 6983            071206                  DT170=DT165
 6984            071206                  DT171=DT165
 6985            071206                  DT172=DT165
 6986            071206                  DT173=DT165
 6987            071206                  DT174=DT165
 6988            071206                  DT175=DT165
 6989            071206                  DT176=DT165
 6990            071206                  DT177=DT165
 6991            071206                  DT200=DT165
 6992            071206                  DT201=DT165
 6993            071206                  DT202=DT165
 6994            071206                  DT203=DT165
 6995            071206                  DT204=DT165
 6996            071206                  DT205=DT165
 6997            071206                  DT206=DT165
 6998
 6999 071274    001232  001234  037215  DT210:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
 7000 071310    037215  001242  000000          .WORD   $TAB,$TMP4,0
 7001 071316    001232  001234  037215  DT207:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS41,$CRLF,$TMP3
 7002 071336    001313  037610  001313          .WORD   $CRLF,MS42,$CRLF,$TMP4,$CRLF,MS43,$CRLF,$TMP5
 7003 071356    001313  037643  001313          .WORD   $CRLF,MS44,$CRLF,$TMP6,0
 7004            071316                  DT211=DT207
 7005 071370    001232  001234  037215  DT212:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP10,$TAB,$TMP11
 7006 071410    001313  037564  001313          .WORD   $CRLF,MS41,$CRLF,$TMP3,$CRLF,MS42,$CRLF,$TMP4
 7007 071430    001313  037626  001313          .WORD   $CRLF,MS43,$CRLF,$TMP5,$CRLF,MS44,$CRLF,$TMP6,0
 7008            071370                  DT213=DT212
 7009            071370                  DT214=DT212
 7010            071370                  DT215=DT212
 7011            071316                  DT216=DT207
 7012 071452    001232  001234  037215  DT217:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TMP4,0
 7013            071316                  DT220=DT207
 7014            071316                  DT221=DT207
 7015            071316                  DT222=DT207
 7016            071316                  DT223=DT207
 7017            071316                  DT224=DT207
 7018            071316                  DT225=DT207
 7019            071370                  DT226=DT212
```

```
                071370                  DT227=DT212
                071370                  DT230=DT212
                071274                  DT231=DT210
                071316                  DT232=DT207

071472  001232  001234  037215  DT233:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF
071504  037512  001244  001313          .WORD   MS37,$TMP5,$CRLF,MS40,$TMP6
071516  001313  037564  001250          .WORD   $CRLF,MS41,$TMP7,0

071526  001232  001234  037215  DT234:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TMP4,0

071546  001232  001234  037215  DT235:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,0

                071526                  DT236=DT234
                071526                  DT237=DT234
                071526                  DT240=DT234
                071472                  DT241=DT233
                071526                  DT242=DT234
                071472                  DT243=DT233
                071472                  DT244=DT233
                071472                  DT245=DT233
071560  001232  001234  037215  DT247:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,0
071576  001232  001234  037215  DT250:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,0
                071576                  DT251=DT250

071610  001232  001234  037215  DT252:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,0

071622  001232  001234  037215  DT253:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TAB,$TMP10
071642  001313  037024  001254          .WORD   $CRLF,MSA1,$TMP11,$CRLF,MSA2,$TMP12
071656  001313  037065  001244          .WORD   $CRLF,MSA3,$TMP5,$CRLF,MSA4,$TMP6
071672  001313  037115  001240          .WORD   $CRLF,MSA5,$TMP3,$CRLF,MSA6,$TMP4,0
                071622                  DT254=DT253
                071622                  DT255=DT253
                071622                  DT256=DT253
                071622                  DT257=DT253
                071622                  DT260=DT253
                071622                  DT261=DT253
                071622                  DT262=DT253
                071622                  DT263=DT253
                071622                  DT264=DT253
                071622                  DT265=DT253

                071610                  DT266=DT252

                                        ;12345
                000001                          .END
```

SYMBOL TABLE

```
AAADON  014406
AAA1    013404
AAA10   014022
AAA11   014060
AAA12   014116
AAA13   014154
AAA2    013442
AAA3    013500
AAA4    013536
AAA5    013574
AAA6    013632
AAA7    013670
AAA8    013726
AAA9    013764
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
BBBDON  015110
BBBER1  014664
BBBER2  014750
BBBER3  014776
BBBER4  015024
BBBP1   015070
BBBP2   015100
BBBO    014414
BBB1    014450
BBB2    014476
BBB3    014526
BBB4    014554
BBB5    014612
BBB6    014620
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
CCCDON  016120
CCC1    015114
CCC10   015510
CCC11   015544
CCC12   015600
CCC13   015634
CCC2    015150
CCC3    015204
CCC4    015240
CCC5    015274
CCC6    015330
CCC7    015364
CCC8    015420
CCC9    015454
CKSWR = 104406
CMPSUB  014216
CMPTMP  014376
CNT   = 000267
CORDON  033142
CORFLG  033124
CORINT  033136
CORMES  037665
CORSUB  032642
CORTMP  033126
CORTRP  033140
CORTV   032740
CORTV0  033102
CORTV1  033106
COR1    032010
COR10   032432
COR11   032476
COR12   032536
COR13   032576
COR2    032034
COR3    032050
COR33   032062
COR4    032122
COR5    032162
COR6    032222
COR7    032266
COR8    032326
COR9    032372
CPSPUR  036712
CPTWO   036730
CR    = 000015
CRLF  = 000200
DDDDON  017054
DDD1    016124
DDD2    016200
DDD3    016254
DDD4    016330
DDD5    016404
DDD6    016460
DDD7    016534
DDISP = 177570
DF1     070056
DF10  = 070056
DF100 = 070277
DF101 = 070277
DF102 = 070277
DF103 = 070277
DF104 = 070277
DF105 = 070277
DF106 = 070277
DF107 = 070277
DF11  = 070056
DF110 = 070277
DF111 = 070125
DF112 = 070125
DF113 = 070125
DF114 = 070125
DF115   070175
DF116 = 070175
DF117 = 070175
DF12  = 070056
DF120 = 070175
DF121 = 070175
DF122 = 070175
DF123 = 070151
DF124 = 070151
DF125 = 070151
DF126 = 070151
DF127   070221
DF13  = 070056
DF130 = 070221
DF131 = 070221
DF132 = 070151
DF133 = 070221
DF134 = 070221
DF135 = 070151
DF136 = 070221
DF137 = 070175
DF14  = 070056
DF140 = 070175
DF141 = 070175
DF142 = 070175
DF143 = 070175
DF144 = 070175
DF145 = 070125
DF146 = 070125
DF147 = 070125
DF15  = 070056
DF150 = 070125
DF151 = 070221
DF152 = 070221
DF153 = 070221
DF154 = 070221
DF155 = 070221
DF156 = 070221
DF157 = 070151
DF16  = 070056
DF160 = 070221
DF161 = 070151
DF162 = 070151
DF163 = 070221
DF164 = 070151
DF165   070331
DF166 = 070331
DF167 = 070331
DF17    070106
DF170 = 070331
DF171 = 070331
DF172 = 070331
DF173 = 070331
DF174 = 070331
DF175 = 070331
DF176   070363
DF177 = 070363
DF2     070074
DF20    070115
DF200 = 070363
DF201 = 070363
DF202 = 070363
DF203 = 070363
DF204 = 070363
DF205 = 070363
DF206 = 070363
DF207   070425
DF21  = 070115
DF210   07[illegible]15
DF211 = 070425
DF212   070451
DF213 = 070451
DF214 = 070451
DF215 = 070451
DF216 = 070425
DF217   070501
DF22  = 070115
DF220 = 070425
DF221 = 070425
DF222 = 070425
DF223 = 070425
DF224 = 070425
DF225 = 070425
DF226 = 070451
DF227 = 070451
DF23    070125
DF230 = 070451
DF231 = 070451
DF232 = 070451
DF233   070510
DF234   070525
DF235   070534
DF236 = 070525
DF237 = 070525
DF24  = 070125
DF240 = 070525
DF241 = 070510
DF242 = 070525
DF243 = 070510
```

SYMBOL TABLE

```
DF244 = 070510
DF245 = 070510
DF246 = 070151
DF247   070540
DF25  = 070125
DF250 = 070540
DF251 = 070540
DF252   070546
DF253   070552
DF254 = 070552
DF255 = 070552
DF256 = 070552
DF257 = 070552
DF26  = 070125
DF260 = 070552
DF261 = 070552
DF262 = 070552
DF263 = 070552
DF264 = 070552
DF265 = 070552
DF266 = 070546
DF27  = 070125
DF3   = 070056
DF30  = 070125
DF31  = 070125
DF32  = 070125
DF33    070151
DF34  = 070151
DF35  = 070151
DF36  = 070151
DF37  = 070125
DF4   = 070056
DF40  = 070125
DF41  = 070125
DF42  = 070125
DF43  = 070125
DF44  = 070125
DF45  = 070125
DF46  = 070151
DF47  = 070151
DF5   = 070056
DF50  = 070151
DF51  = 070151
DF52  = 070151
DF53    070245
DF54  = 070245
DF55  = 070245
DF56  = 070245
DF57  = 070245
DF6   = 070056
DF60  = 070245
DF61  = 070245
DF62  = 070245
DF63  = 070245
DF64  = 070245
DF65  = 070245
DF66  = 070245
DF67  = 070245
DF7   = 070056
DF70    070277
DF71  = 070277
DF72  = 070277
DF73  = 070277
DF74  = 070277
DF75  = 070277
DF76  = 070277
DF77  = 070277
DH1     066262
DH10  = 066262
DH100 = 066262
DH101 = 066262
DH102 = 066262
DH103 = 066262
DH104 = 066262
DH105 = 066262
DH106 = 066262
DH107 = 066262
DH11  = 066262
DH110 = 066262
DH111 = 066262
DH112 = 066262
DH113 = 066262
DH114 = 066262
DH115   066623
DH116 = 066623
DH117 = 066623
DH12  = 066262
DH120 = 066623
DH121 = 066623
DH122 = 066623
DH123 = 066262
DH124 = 066262
DH125 = 066262
DH126 = 066262
DH127 = 066623
DH13  = 066262
DH130 = 066623
DH131 = 066623
DH132 = 066262
DH133 = 066623
DH134 = 066623
DH135 = 066262
DH136 = 066623
DH137 = 066623
DH14  = 066262
DH140 = 066623
DH141 = 066623
DH142 = 066623
DH143 = 066623
DH144 = 066623
DH145 = 066262
DH146 = 066262
DH147 = 066262
DH15  = 066262
DH150 = 066262
DH151 = 066623
DH152 = 066623
DH153 = 066623
DH154 = 066623
DH155 = 066623
DH156 = 066623
DH157 = 066262
DH16  = 066262
DH160 = 066623
DH161 = 066262
DH162 = 066262
DH163 = 066623
DH164 = 066262
DH165 = 066623
DH166 = 066623
DH167 = 066623
DH17    066413
DH170 = 066623
DH171 = 066623
DH172 = 066623
DH173 = 066623
DH174 = 066623
DH175 = 066623
DH176 = 066623
DH177 = 066623
DH2     066352
DH20  = 066262
DH200 = 066623
DH201 = 066623
DH202 = 066623
DH203 = 066623
DH204 = 066623
DH205 = 066623
DH206 = 066623
DH207   066762
DH21    066466
DH210   067052
DH211 = 067052
DH212 = 066762
DH213 = 066762
DH214 = 066762
DH215 = 066762
DH216 = 067052
DH217   067113
DH22    066555
DH220   066722
DH221 = 067052
DH222 = 066722
DH223 = 067052
DH224 = 066722
DH225 = 067052
DH226   067166
DH227   067255
DH23  = 066262
DH230 = 067166
DH231 = 067255
DH232 = 067255
DH233   067344
DH234   067405
DH235   067473
DH236   067533
DH237 = 067405
DH24  = 066262
DH240   067621
DH241 = 067344
DH242 = 067621
DH243 = 067344
DH244 = 067344
DH245 = 067344
DH246 = 066262
DH247   067711
DH25  = 066262
DH250   067756
DH251 = 067756
DH252 = 066352
DH253 = 066262
DH254 = 066262
DH255 = 066262
DH256 = 066262
DH257 = 066262
DH26  = 066262
DH260 = 066262
DH261 = 066262
DH262 = 066262
DH263 = 066262
DH264 = 066262
DH265 = 066262
DH266   070016
DH27  = 066262
DH3   = 066262
DH30  = 066262
DH31  = 066262
DH32  = 066262
DH33  = 066262
DH34  = 066262
DH35  = 066262
DH36  = 066262
DH37  = 066262
DH4   = 066262
DH40  = 066262
DH41  = 066262
DH42  = 066262
DH43  = 066262
DH44  = 066262
DH45  = 066262
DH46  = 066262
DH47  = 066262
DH5   = 066262
DH50  = 066262
DH51  = 066262
DH52  = 066262
DH53  = 066262
DH54  = 066262
DH55  = 066262
DH56  = 066262
DH57  = 066262
DH6   = 066262
DH60  = 066262
DH61  = 066262
DH62  = 066262
DH63  = 066262
DH64  = 066262
DH65  = 066262
DH66  = 066262
DH67  = 066262
DH7   = 066262
DH70  = 066262
DH71  = 066262
DH72  = 066262
DH73  = 066262
DH74  = 066262
DH75  = 066262
DH76  = 066262
DH77  = 066262
DISPLA  001142
DISPRE  000174
DIVDSU  016614
DIVDT   017044
DIVFSU  015674
DIVFT   016110
DSWR  = 177570
DT1     070604
DT10  = 070604
DT100 = 071120
DT101 = 071120
DT102 = 071120
DT103 = 071120
DT104 = 071120
DT105 = 071120
DT106 = 071120
DT107 = 071120
DT11  = 070604
DT110 = 071120
DT111 = 070772
DT112 = 070772
DT113 = 070772
DT114 = 070772
DT115   071044
DT116 = 071044
DT117 = 071044
DT12  = 070604
DT120 = 071044
DT121 = 071044
DT122 = 071044
DT123 = 070772
DT124 = 070772
DT125 = 070772
DT126 = 070772
DT127 = 071044
DT13  = 070604
```

```
DT130 = 071044
DT131 = 071044
DT132 = 070772
DT133 = 071044
DT134 = 071044
DT135 = 070772
DT136 = 071044
DT137 = 071044
DT14  = 070604
DT140 = 071044
DT141 = 071044
DT142 = 071044
DT143 = 071044
DT144 = 071044
DT145 = 070772
DT146 = 070772
DT147 = 070772
DT15  = 070604
DT150 = 070772
DT151 = 071044
DT152 = 071044
DT153 = 071044
DT154 = 071044
DT155 = 071044
DT156 = 071044
DT157 = 070772
DT16  = 070604
DT160 = 071044
DT161 = 070772
DT162 = 070772
DT163 = 071044
DT164 = 070772
DT165   071206
DT166 = 071206
DT167 = 071206
DT17    070670
DT170 = 071206
DT171 = 071206
DT172 = 071206
DT173 = 071206
DT174 = 071206
DT175 = 071206
DT176 = 071206
DT177 = 071206
DT2     070642
DT20    070710
DT200 = 071206
DT201 = 071206
DT202 = 071206
DT203 = 071206
DT204 = 071206
DT205 = 071206
DT206 = 071206
DT207   071316
DT21    070732
DT210   071274
DT211 = 071316
DT212   071370
DT213 = 071370
DT214 = 071370
DT215 = 071370
DT216 = 071316
DT217   071452
DT22    070754
DT220 = 071316
DT221 = 071316
DT222 = 071316
DT223 = 071316
DT224 = 071316
DT225 = 071316
DT226 = 071370
DT227 = 071370
DT23    070772
DT230 = 071370
DT231 = 071274
DT232 = 071316
DT233   071472
DT234   071526
DT235   071546
DT236 = 071526
DT237 = 071526
DT24  = 070772
DT240 = 071526
DT241 = 071472
DT242 = 071526
DT243 = 071472
DT244 = 071472
DT245 = 071472
DT246 = 070772
DT247   071560
DT25  = 070772
DT250   071576
DT251 = 071576
DT252   071610
DT253   071622
DT254 = 071622
DT255 = 071622
DT256 = 071622
DT257 = 071622
DT26  = 070772
DT260 = 071622
DT261 = 071622
DT262 = 071622
DT263 = 071622
DT264 = 071622
DT265 = 071622
DT266 = 071610
DT27  = 070772
DT3   = 070604
DT30  = 070772
DT31  = 070772
DT32  = 070772
DT33  = 070772
DT34  = 070772
DT35  = 070772
DT36  = 070772
DT37  = 070772
DT4   = 070604
DT40  = 070772
DT41  = 070772
DT42  = 070772
DT43  = 070772
DT44  = 070772
DT45  = 070772
DT46  = 070772
DT47  = 070772
DT5   = 070604
DT50  = 070772
DT51  = 070772
DT52  = 070772
DT53    071120
DT54  = 071120
DT55  = 071120
DT56  = 071120
DT57  = 071120
DT6   = 070604
DT60  = 071120
DT61  = 071120
DT62  = 071120
DT63  = 071120
DT64  = 071120
DT65  = 071120
DT66  = 071120
DT67  = 071120
DT7   = 070604
DT70  = 071120
DT71  = 071120
DT72  = 071120
DT73  = 071120
DT74  = 071120
DT75  = 071120
DT76  = 071120
DT77  = 071120
EEEDON  020064
EEE1    017060
EEE10   017454
EEE11   017510
EEE12   017544
EEE13   017600
EEE2    017114
EEE3    017150
EEE4    017204
EEE5    017240
EEE6    017274
EEE7    017330
EEE8    017364
EEE9    017420
EMTVEC= 000030
EM1     037724
EM10    040543
EM100   053701
EM101   053764
EM102   054160
EM103   054354
EM104   054466
EM105   054633
EM106   054742
EM107   055051
EM11    040650
EM110   055160
EM111   043775
EM112   044052
EM113   044130
EM114   044206
EM115   044265
EM116   044343
EM117   044422
EM12    040757
EM120   044560
EM121   044716
EM122   045053
EM123   045210
EM124   045266
EM125   045345
EM126   045423
EM127   045502
EM13    041066
EM130   045560
EM131   045637
EM132   045775
EM133   046133
EM134   046271
EM135   046426
EM136   046563
EM137   046720
EM14    041202
EM140   047006
EM141 = 043775
EM142 = 044052
EM143   047075
EM144   047153
EM145   047232
EM146   047377
EM147   047544
EM15    041314
EM150   047710
EM151   050054
EM152   050142
EM153 = 045210
EM154 = 045266
EM155   050231
EM156   050307
EM157   050366
EM16    041426
EM160   050533
EM161   050671
EM162   051036
EM163   051202
EM164   051337
EM165   055270
EM166   055354
EM167   055437
EM17    041523
EM170   055471
EM171   055557
EM172   055702
EM173   055767
EM174   056135
EM175   056303
EM176   056415
EM177   056501
EM2     037756
EM20    041600
EM200   056564
EM201   056616
EM202   056705
EM203   057047
EM204   057211
EM205   057277
EM206   057445
EM207   057613
EM21    041632
EM210   057655
EM211   057717
EM212   060063
EM213   060247
EM214   060433
EM215   060617
EM216   061003
EM217   061165
EM22    041664
EM220   061227
EM221   061457
EM222   061723
EM223   062154
EM224   062421
EM225   062652
EM226   063117
EM227   063252
EM23    041745
EM230   063405
EM231   063541
EM232 = 057655
EM233   063675
EM234   063734
EM235   064020
EM236   064066
EM237   064121
EM24    042022
EM240   064205
EM241   064241
EM242   064344
EM243   064400
EM244   064503
EM245   064542
```

```
EM246   043405          EM70    053166          GGG5    024464          HHHDON  030004          HXER11  013200
EM247   064624          EM71    053217          GGG6    024534          HHH1    025702          HXER12  013220
EM25    042120          EM72    053247          GGG7    024604          HHH10   027002          HXER13  013250
EM250   064660          EM73  = 053247          GGG8    024654          HHH11   027102          HXER2   012716
EM251   064712          EM74    053301          GGG9    024724          HHH12   027202          HXER22  012732
EM252   064745          EM75    053410          GGP1    011550          HHH13   027302          HXER3   012746
EM253   065036          EM76    053501          GGP2    011560          HHH2    026002          HXER33  012762
EM254   065122          EM77    053571          GGP3    011570          HHH3    026102          HXER4   012776
EM255   065207          ERM10   034152          GGP4    011600          HHH4    026202          HXER5   012676
EM256   065275          ERROR = 104000          GGP5    011610          HHH5    026302          HXER6   013032
EM257   065364          ERRVEC= 000004          GGP6    011620          HHH6    026402          HXER66  013046
EM26    042205          ERTYPE  036204          GGP7    011630          HHH7    026502          HXER7   013062
EM260   065452          ERT1    036370          GGP8    011640          HHH8    026602          HXER8   013014
EM261   065541          ERT2    036624          GGP9    011650          HHH9    026702          HXER9   013112
EM262   065631          ERT3    036630          GG1     006746          HHP0    006602          HXP1    013300
EM263   065722          ERT4    036642          GG10    007242          HHP1    006612          HXP2    013310
EM264   066007          ERT5    036654          GG11    007300          HHP10   006722          HXP3    013320
EM265   066074          FFFDON  020614          GG12    007322          HHP11   006732          HXP4    013330
EM266   066161          FFF1    020070          GG13    007332          HHP2    006622          HXP5    013340
EM27  = 042120          FFF2    020144          GG14    007350          HHP3    006632          HXP6    013350
EM3     040012          FFF3    020220          GG15    007406          HHP4    006642          HXP7    013360
EM30    042303          FFF4    020274          GG16    007424          HHP5    006652          HXP8    013370
EM31    042355          FPSPUR  036660          GG17    007472          HHP6    006662          HX1     011664
EM32    041770          FPVECT= 000244          GG18    007502          HHP7    006672          HX10    012072
EM33    042422          GGDATO  011540          GG19    007536          HHP8    006702          HX11    012114
EM34    042445          GGDONE  011660          GG2     007004          HHP9    006712          HX12    012134
EM35    042477          GGER0   010036          GG20    007574          HHTRAP  006522          HX13    012144
EM36    042552          GGER1   010134          GG21    007616          HH1     005022          HX14    012152
EM37    042620          GGER10  010646          GG22    007626          HH10    005234          HX15    012170
EM4     040117          GGER11  010714          GG23    007644          HH11    005252          HX16    012216
EM40    042643          GGER12  010762          GG24    007702          HH12    005302          HX165   012222
EM41    042675          GGER13  011030          GG25    007720          HH13    005324          HX17    012250
EM42    042750          GGER14  011076          GG26    007766          HH14    005344          HX18    012306
EM43    043101          GGER15  011174          GG27    007776          HH15    005354          HX19    012326
EM44    043232          GGER16  011242          GG28    010032          HH16    005362          HX2     011714
EM45    043300          GGER17  011310          GG3     007026          HH17    005400          HX20    012336
EM46    043353          GGER18  011356          GG4     007036          HH18    005436          HX21    012340
EM47    043430          GGER19  011424          GG5     007054          HH19    005460          HX22    012370
EM5     040224          GGER2   010202          GG6     007112          HH2     005052          HX23    012410
EM50    043564          GGER20  011472          GG7     007130          HH20    005470          HX24    012430
EM51    043637          GGER3   010250          GG8     007176          HH21    005506          HX25    012440
EM52    043705          GGER4   010316          GG9     007206          HH22    005536          HX26    012446
EM53    051535          GGER5   010320          GTSWR = 104405          HH23    005560          HX27    012450
EM54    051566          GGER6   010366          HHDATO  006572          HH24    005570          HX28    012502
EM55    051503          GGER7   010434          HHDONE  006742          HH25    005606          HX29    012524
EM56    051616          GGER8   010532          HHER0   005612          HH3     005074          HX3     011730
EM57    051707          GGER9   010600          HHER00  005726          HH4     005114          HX30    012534
EM6     040331          GGGDON  025676          HHER1   005660          HH5     005124          HX31    012552
EM60    051777          GGG1    024224          HHER10  006454          HH6     005132          HX32    012602
EM61    052101          GGG10   024774          HHER2   005774          HH7     005144          HX33    012624
EM62    052275          GGG11   025044          HHER3   006042          HH8     005202          HX34    012634
EM63    052471          GGG12   025114          HHER4   006110          HH9     005224          HX35    012652
EM64    052602          GGG13   025164          HHER5   006156          HT    = 000011          HX4     011750
EM65    052721          GGG14   025234          HHER6   006224          HXDATO  013270          HX5     011770
EM66    053034          GGG2    024274          HHER7   006272          HXDONE  013400          HX6     012000
EM67    053103          GGG3    024344          HHER8   006340          HXER1   012656          HX7     012006
EM7     040436          GGG4    024414          HHER9   006406          HXER10  013146          HX8     012024
```

```
HX9     012056  MS37    037512  SW0   = 000001  $ATY1   035030  $FATAL  001320
IIIDON  021454  MS4     037307  SW00  = 000001  $ATY3   035036  $FFLG   035274
IIII    020620  MS40    037530  SW01  = 000002  $ATY4   035046  $FILLC  001156
IIII2   020664  MS41    037564  SW02  = 000004  $AUTOB  001134  $FILLS  001155
IIII3   020730  MS415   037544  SW03  = 000010  $BASE   001372  $GDADR  001120
IIII4   020774  MS42    037610  SW04  = 000020  $BDADR  001122  $GDDAT  001124
IOTVEC= 000020  MS43    037626  SW05  = 000040  $BDDAT  001126  $GET42  033330
JJJDON  022414  MS44    037643  SW06  = 000100  $BELL   001306  $GTSWR  035346
JJJ1    021460  MS5     037335  SW07  = 000200  $CDW1   001376  $HD   = 000003
JJJ2    021544  MS6     037357  SW08  = 000400  $CDW2   001400  $HIBTS  004322
JJJ3    021630  MS7     037375  SW09  = 001000  $CHARC  034576  $ICNT   001104
JJJ4    021714  MULDSU  020354  SW1   = 000002  $CKSWR  035276  $ILLUP  036176
KKKDON  023256  MULDT   020604  SW10  = 002000  $CLR.T  033346  $INTAG  001135
KKK1    022420  MULFSU  017640  SW11  = 004000  $CMTAG  001100  $ITEMB  001114
KKK3    022464  MULFT   020054  SW12  = 010000  $CM1   = 000024  $LF     001314
KKK4    022530  NNNDON  031760  SW13  = 020000  $CM2   = 000050  $LFLG   035273
KKK5    022574  NNN1    030736  SW14  = 040000  $CM3   = 000024  $LOOP   033424
LF    = 000012  NNN2    031040  SW15  = 100000  $CM4   = 000024  $LPADR  001106
LLLDON  024220  NNN3    031142  SW2   = 000004  $CNTLG  035705  $LPERR  001110
LLL1    023262  NNN4    031244  SW3   = 000010  $CNTLU  035700  $MADR1  001350
LLL2    023346  OVDNTT  022404  SW4   = 000020  $CPUOP  001344  $MADR2  001354
LLL3    023432  OVDTT   024210  SW5   = 000040  $CRLF   001313  $MADR3  001360
LLL4    023516  OVFNTT  021444  SW6   = 000100  $DDW0   001402  $MADR4  001364
LOOP    005020  OVFTT   023246  SW7   = 000200  $DDW1   001404  $MAIL   001316
LPERR = 104413  OVUNDN  022004  SW8   = 000400  $DDW10  001426  $MAMS1  001346
MMMDON  030732  OVUNDT  023606  SW9   = 001000  $DDW11  001430  $MAMS2  001352
MMM1    030010  OVUNFN  021044  TAB   = 000011  $DDW12  001432  $MAMS3  001356
MMM2    030062  OVUNFT  022644  TBITVE= 000014  $DDW13  001434  $MAMS4  001362
MMM3    030134  PIRQ  = 177772  TKVEC = 000060  $DDW14  001436  $MBADR  004324
MMM4    030206  PIRQVE= 000240  TPVEC = 000064  $DDW15  001440  $MFLG   035272
MMM5    030260  POWERM  037150  TRAPVE= 000034  $DDW2   001406  $MNEW   035723
MNUMBE= 000266  PROGNU= 000002  TRTVEC= 000014  $DDW3   001410  $MSGAD  001332
MODDD0  031740  PR0   = 000000  TST1    005020  $DDW4   001412  $MSGLG  001334
MODDD1  031750  PR1   = 000040  TST10   017056  $DDW5   001414  $MSGTY  001316
MODDOV  031352  PR2   = 000100  TST11   020066  $DDW6   001416  $MSWR   035712
MODDSU  027406  PR3   = 000140  TST12   020616  $DDW7   001420  $MTYP1  001347
MODDT0  027764  PR4   = 000200  TST13   021456  $DDW8   001422  $MTYP2  001353
MODDT1  027774  PR5   = 000240  TST14   022416  $DDW9   001424  $MTYP3  001357
MODFD0  030712  PR6   = 000300  TST15   023260  $DEVCT  001326  $MTYP4  001363
MODFD1  030722  PR7   = 000340  TST16   024222  $DEVM   001374  $MXCNT  033714
MODFOV  030336  PS    = 177776  TST17   025700  $DOAGN  033366  $NULL   001154
MODFSU  025310  PSW   = 177776  TST2    006744  $ENDAD  033356  $NWTST= 000001
MODFT0  025646  PWRVEC= 000024  TST20   030006  $ENDCT  033200  $OCNT   035024
MODFT1  025656  RDCHR = 104407  TST21   030734  $ENULL  033432  $OMODE  035026
MODP1   025666  RESREG= 104411  TST22   031762  $ENV    001336  $OVER   033700
MSA1    037024  RESVEC= 000010  TST23   033144  $ENVM   001337  $PASS   001324
MSA2    037042  RSETUP= 104412  TST3    011662  $EOP    033144  $PASTM  004330
MSA3    037065  R6    =%000006  TST4    013402  $EOPCT  033172  $PWRAD  036160
MSA4    037077  R7    =%000007  TST5    014410  $ERFLG  001103  $PWRDN  036020
MSA5    037115  SAVREG= 104410  TST6    015112  $ERMAX  001115  $PWRMG  036154
MSA6    037130  SCOPE = 000004  TST7    016122  $ERROR  033716  $PWRUP  036072
MS1     037222  SPACE   037217  TYPE  = 104401  $ERRPC  001116  $QUES   001312
MS10    037421  STACK = 001100  TYPOC = 104402  $ERRTB  001442  $RDCHR  035560
MS11    037443  START   004336  TYPON = 104404  $ERTTL  001112  $RDSZ = 000001
MS12    037467  STKLMT= 177774  TYPOS = 104403  $ESCAP  001304  $REGAD  001160
MS2     037240  SWR     001140  $APTHD  004322  $ETABL  001336  $REG0   001162
MS3     037260  SWREG   000176  $ATYC   035054  $ETEND  001442  $REG1   001164
```

```
CKFPBA0 FP11F FLTG PNT PRT B    MACRO M1111  18-SEP-79 13:43  PAGE 19-14
SYMBOL TABLE                                                          SEQ 0171

$REG10  001202          $RESRE  034212          $TESTN  001322          $TMP22  001276          $TYPE   034250
$REG11  001204          $RTNAD  033426          $TIMES  001302          $TMP23  001300          $TYPEC  034462
$REG12  001206          $RTRN   033422          $TKB    001146          $TMP3   001240          $TYPEX  034600
$REG13  001210          $SAVRE  034154          $TKS    001144          $TMP4   001242          $TYPOC  034626
$REG14  001212          $SAVR6  036202          $TMP0   001232          $TMP5   001244          $TYPON  034642
$REG15  001214          $SCOPE  033436          $TMP1   001234          $TMP6   001246          $TYPOS  034602
$REG16  001216          $SETUP= 000137          $TMP10  001252          $TMP7   001250          $UNIT   001330
$REG17  001220          $STUP = 177777          $TMP11  001254          $TN    = 000023         $UNITM  004332
$REG2   001166          $SVLAD  033644          $TMP12  001256          $TPB    001152          $USWR   001342
$REG20  001222          $SVPC = 004322          $TMP13  001260          $TPFLG  001157          $VECT1  001366
$REG21  001224          $SWR  = 177400          $TMP14  001262          $TPS    001150          $VECT2  001370
$REG22  001226          $SWREG  001340          $TMP15  001264          $TRAP   035734          $XTSTR  033450
$REG23  001230          $SWRMK= 000000          $TMP16  001266          $TRAP2  035756          $$GET4= 000001
$REG3   001170          $SWRMS= 000200          $TMP17  001270          $TRP   = 000014         $OFILL  035025
$REG4   001172          $TAB    037215          $TMP2   001236          $TRPAD  035770          .LPER   036746
$REG5   001174          $TBIT   033430          $TMP20  001272          $TSTM   004326          .RSET   036754
$REG6   001176          $TERM = 000030          $TMP21  001274          $TSTNM  001102          .$X    = 004322
$REG7   001200

. ABS.  071710      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  58376 WORDS  ( 229 PAGES)
DYNAMIC MEMORY:  20434 WORDS  ( 78 PAGES)
ELAPSED TIME:  00:15:56
CKFPBA0.BIN,CKFPBA0.SEQ/CRF=CKFPBA0.MLB/ML,CKFPBA0.P11
```

CKFPBA0 CREATED BY MACRO ON 18-SEP-79 AT 13:50 PAGE 1
SYMBOL CROSS REFERENCE CREF SEQ 0172

```
SYMBOL  VALUE            REFERENCES
AAADON     014406         12-2020   #12-2106
AAA1       013404        #11-1821
AAA10      014022        #11-1952
AAA11      014060        #12-1970
AAA12      014116        #12-1987
AAA13      014154        #12-2004
AAA2       013442        #11-1832
AAA3       013500        #11-1845
AAA4       013536        #11-1858
AAA5       013574        #11-1872
AAA6       013632        #11-1885
AAA7       013670        #11-1902
AAA8       013726        #11-1919
AAA9       013764        #11-1936
ABASE   = 000000          8-890     8-890
ACDW1   = 000000          8-890     8-890
ACDW2   = 000000          8-890     8-890
ACPUOP  = 000000          8-890     8-890
AC0     =%000000         #7-879    *9-928    *9-930     9-933    *9-962    *9-964     9-967    *9-986    *9-988
                          9-991    *9-1016   *9-1018    9-1021   *9-1038   *9-1040    9-1043   *9-1158   *9-1160
                          9-1163   *9-1182   *9-1184    9-1187    9-1198   *9-1223   *9-1225    9-1228   *9-1247
                         *9-1249    9-1252    9-1263   *9-1289   *9-1291    9-1294   *11-1315  *11-1317   11-1320
                          11-1331   11-1362   11-1388    11-1414  *11-1486  *11-1489   11-1496  *11-1520  *11-1527
                          11-1537  *11-1568  *11-1583   *11-1586   11-1589  *11-1603  *11-1605   11-1608  *11-1633
                         *11-1634  *11-1638   11-1643   *11-1664  *11-1668   11-1673  *12-2053   12-2059   12-2087
                         *12-2129  *12-2132  *12-2152   *12-2154  *12-2174  *12-2179  *12-2452  *12-2459   12-2466
                         *13-2636  *13-2644   13-2651   *13-2887  *13-2894   13-2901  *13-3034  *13-3042   13-3049
                         *13-3221  *13-3240   13-3247    14-3319  *14-3473  *14-3492   14-3499   14-3570  *14-3719
                         *14-3738   14-3754   14-3832   *15-3971  *15-3990   15-4006   15-4084  *15-4373  *15-4382
                          15-4388  *16-4742  *16-4751    16-4757  *16-4980  *16-4989   16-4996  *16-5199  *16-5208
                          16-5215   17-5373  *17-5373   *17-5386  *17-5400  *17-5414  *17-5430  *17-5444  *17-5460
                         *17-5473  *17-5488  *17-5502   *17-5516  *17-5559   17-5585
AC1     =%000001         #7-880    *15-4375   15-4390   *16-4744   16-4759  *16-4982   16-4998  *16-5201   16-5217
AC2     =%000002         #7-881
AC3     =%000003         #7-882
AC4     =%000004         #7-883
AC5     =%000005         #7-884
AC6     =%000006         #7-885
AC7     =%000007         #7-886
ADDW0   = 000000          8-890     8-890
ADDW1   = 000000          8-890     8-890
ADDW10  = 000000          8-890     8-890
ADDW11  = 000000          8-890     8-890
ADDW12  = 000000          8-890     8-890
ADDW13  = 000000          8-890     8-890
ADDW14  = 000000          8-890     8-890
ADDW15  = 000000          8-890     8-890
ADDW2   = 000000          8-890     8-890
ADDW3   = 000000          8-890     8-890
ADDW4   = 000000          8-890     8-890
ADDW5   = 000000          8-890     8-890
ADDW6   = 000000          8-890     8-890
```

SYMBOL	VALUE	REFERENCES					
ADDW7	= 000000	8-890	8-890				
ADDW8	= 000000	8-890	8-890				
ADDW9	= 000000	8-890	8-890				
ADEVCT	= 000000	8-890	8-890				
ADEVM	= 000000	8-890	8-890				
AENV	= 000000	8-890	8-890				
AENVM	= 000000	8-890	8-890				
AFATAL	= 000000	8-890	8-890				
AMADR1	= 000000	8-890	8-890				
AMADR2	= 000000	8-890	8-890				
AMADR3	= 000000	8-890	8-890				
AMADR4	= 000000	8-890	8-890				
AMAMS1	= 000000	8-890	8-890				
AMAMS2	= 000000	8-890	8-890				
AMAMS3	= 000000	8-890	8-890				
AMAMS4	= 000000	8-890	8-890				
AMSGAD	= 000000	8-890	8-890				
AMSGLG	= 000000	8-890	8-890				
AMSGTY	= 000000	8-890	8-890				
AMTYP1	= 000000	8-890	8-890				
AMTYP2	= 000000	8-890	8-890				
AMTYP3	= 000000	8-890	8-890				
AMTYP4	= 000000	8-890	8-890				
APASS	= 000000	8-890	8-890				
APRIOR	= 000000	8-890					
APTCSU	= 000040	17-5651	#17-5655				
APTENV	= 000001	17-5647	17-5651	17-5655	#17-5655		
APTSIZ	= 000200	9-903	#17-5655				
APTSPO	= 000100	17-5651	17-5655	#17-5655			
ASWREG	= 000000	8-890	8-890				
ATESTN	= 000000	8-890	8-890				
AUNIT	= 000000	8-890	8-890				
AUSWR	= 000000	8-890	8-890				
AVECT1	= 000000	8-890	8-890				
AVECT2	= 000000	8-890	8-890				
BBBDON	015110	12-2203	12-2224	12-2232	12-2240	12-2253	#12-2260
BBBER1	014664	12-2126	#12-2207				
BBBER2	014750	12-2139	12-2161	12-2197	#12-2227		
BBBER3	014776	12-2143	12-2165	12-2201	#12-2235		
BBBER4	015024	12-2182	#12-2244				
BBBP1	015070	12-2128	12-2130	12-2153	12-2177	#12-2255	
BBBP2	015100	12-2151	12-2173	#12-2256			
BBB0	014414	#12-2122					
BBB1	014450	12-2127	#12-2132				
BBB2	014476	#12-2146					
BBB3	014526	12-2150	#12-2154				
BBB4	014554	#12-2168					
BBB5	014612	12-2172	#12-2179	12-2184			
BBB6	014620	12-2176	#12-2184				
BIT0	= 000001	#7-877					
BIT00	= 000001	#7-877	7-877				
BIT01	= 000002	#7-877	7-877				

```
SYMBOL  VALUE               REFERENCES
BIT02   = 000004            #7-877      7-877
BIT03   = 000010            #7-877      7-877
BIT04   = 000020            #7-877      7-877
BIT05   = 000040            #7-877      7-877
BIT06   = 000100            #7-877      7-877
BIT07   = 000200            #7-877      7-877
BIT08   = 000400            #7-877      7-877       17-5645
BIT09   = 001000            #7-877      7-877       17-5645     17-5647     17-5647
BIT1    = 000002            #7-877
BIT10   = 002000            #7-877      17-5647
BIT11   = 004000            #7-877      17-5645
BIT12   = 010000            #7-877      17-5643
BIT13   = 020000            #7-877      17-5647
BIT14   = 040000            #7-877      17-5645
BIT15   = 100000            #7-877
BIT2    = 000004            #7-877
BIT3    = 000010            #7-877
BIT4    = 000020            #7-877
BIT5    = 000040            #7-877
BIT6    = 000100            #7-877
BIT7    = 000200            #7-877
BIT8    = 000400            #7-877
BIT9    = 001000            #7-877
BPTVEC  = 000014            #7-877
CCCDON    016120             12-2420    #12-2507
CCC1      015114            #12-2274
CCC10     015510            #12-2376
CCC11     015544            #12-2387
CCC12     015600            #12-2399
CCC13     015634            #12-2410
CCC2      015150            #12-2285
CCC3      015204            #12-2296
CCC4      015240            #12-2308
CCC5      015274            #12-2319
CCC6      015330            #12-2330
CCC7      015364            #12-2341
CCC8      015420            #12-2353
CCC9      015454            #12-2365
CKSWR   = 104406             17-5645    17-5647     17-5647    #17-5659     17-5886
CMPSUB    014216             11-1822    11-1833     11-1846     11-1859     11-1873     11-1886     11-1903     11-1920     11-1937
                             11-1953    12-1971     12-1988     12-2005    #12-2046
CMPTMP    014376             12-2086    12-2098    #12-2102
CNT     = 000267            #9-892      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
                            #9-896      9-896       9-896      #9-896       9-896       9-896      #9-896       9-896       9-896
```

```
                                                                         G 14
CKFPBAO     CREATED BY  MACRO  ON 18-SEP-79 AT 13:50        PAGE 4
SYMBOL CROSS REFERENCE                                      CREF                                                          SEQ 0175
SYMBOL  VALUE                REFERENCES
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896     9-896     9-896
                             #9-896     9-896     9-896     #9-896     9-896     9-896     #9-896
CORDON   033142                17-5350    17-5359   17-5522    17-5618   #17-5635
```

SYMBOL	VALUE	REFERENCES								
CORFLG	033124	17-5375	17-5388	17-5402	17-5416	17-5432	17-5446	17-5461	17-5474	17-5490
		17-5504	17-5518	17-5553	*17-5569	17-5576	17-5615	#17-5620		
CORINT	033136	17-5353	17-5571	17-5578	#17-5623					
CORMES	037665	17-5366	#18-5933							
CORSUB	032642	17-5369	17-5382	17-5396	17-5410	17-5426	17-5440	17-5456	17-5469	17-5484
		17-5498	17-5512	#17-5553						
CORTMP	033126	17-5584	17-5586	17-5603	#17-5621					
CORTRP	033140	17-5568	#17-5628							
CORTV	032740	17-5568	#17-5576							
CORTV0	033102	17-5598	17-5601	17-5606	#17-5610					
CORTV1	033106	17-5577	#17-5612							
COR1	032010	17-5346	#17-5352							
COR10	032432	#17-5468								
COR11	032476	#17-5483								
COR12	032536	#17-5497								
COR13	032576	#17-5511								
COR2	032034	17-5352	#17-5357							
COR3	032050	17-5349	17-5355	#17-5362						
COR33	032062	17-5364	#17-5367							
COR4	032122	#17-5381								
COR5	032162	#17-5395								
COR6	032222	#17-5409								
COR7	032266	#17-5425								
COR8	032326	#17-5439								
COR9	032372	#17-5455								
CPSPUR	036712	11-1572	11-1584	11-1737	17-5354	17-5358	#17-5847	17-5891		
CPTWO	036730	#17-5858	17-5892							
CR	= 000015	#7-877	17-5651	17-5651						
CRLF	= 000200	#7-875	#7-877	9-904	9-904	17-5651	17-5651	18-5913	18-5913	18-5933
		18-5933	18-5942	18-5944	18-5946	18-5948	18-5950	18-5952	18-5954	18-5956
		18-5958	18-5960	18-5962	18-5964	18-5991	18-5993	18-6000	18-6002	18-6003
		18-6005	18-6006	18-6008	18-6010	18-6017	18-6019	18-6020	18-6022	18-6024
		18-6028	18-6030	18-6050	18-6052	18-6064	18-6066	18-6071	18-6073	18-6080
		18-6082	18-6102	18-6104	18-6114	18-6116	18-6121	18-6123	18-6135	18-6137
		18-6139	18-6141	18-6143	18-6144	18-6146	18-6147	18-6149	18-6151	18-6153
		18-6155	18-6167	18-6169	18-6171	18-6173	18-6175	18-6177	18-6179	18-6180
		18-6182	18-6183	18-6185	18-6187	18-6188	18-6190	18-6192	18-6194	18-6217
		18-6221	18-6223	18-6225	18-6233	18-6235	18-6237	18-6239	18-6241	18-6243
		18-6341	18-6341	18-6367	18-6367	18-6367	18-6368	18-6368	18-6368	18-6369
		18-6369	18-6369	18-6370	18-6370	18-6370	18-6372	18-6372	18-6375	18-6375
		18-6376	18-6376	18-6376	18-6376	18-6377	18-6377	18-6378	18-6378	18-6378
		18-6378	18-6379	18-6379	18-6380	18-6380	18-6380	18-6380	18-6381	18-6381
		18-6382	18-6382	18-6383	18-6383	18-6384	18-6384	18-6416	18-6419	18-6421
		18-6423	18-6427							
DDDDON	017054	12-2601	#13-2697							
DDD1	016124	#12-2520								
DDD2	016200	#12-2531								
DDD3	016254	#12-2542								
DDD4	016330	#12-2553								
DDD5	016404	#12-2564								
DDD6	016460	#12-2577								
DDD7	016534	#12-2589								

```
SYMBOL   VALUE          REFERENCES
DDISP    = 177570       #7-877    8-890    9-903
DF1        070056         9-896   #19-6663  19-6665  19-6666  19-6667  19-6668  19-6669  19-6670  19-6671
                         19-6672   19-6673  19-6674  19-6675  19-6676
DF10     = 070056         9-896   #19-6670
DF100    = 070277         9-896   #19-6771
DF101    = 070277         9-896   #19-6772
DF102    = 070277         9-896   #19-6773
DF103    = 070277         9-896   #19-6774
DF104    = 070277         9-896   #19-6775
DF105    = 070277         9-896   #19-6776
DF106    = 070277         9-896   #19-6777
DF107    = 070277         9-896   #19-6778
DF11     = 070056         9-896   #19-6671
DF110    = 070277         9-896   #19-6779
DF111    = 070125         9-896   #19-6706
DF112    = 070125         9-896   #19-6707
DF113    = 070125         9-896   #19-6708
DF114    = 070125         9-896   #19-6709
DF115      070175         9-896   #19-6710  19-6711  19-6712  19-6713  19-6714  19-6715  19-6728  19-6729
                         19-6730   19-6731  19-6732  19-6733
DF116    = 070175         9-896   #19-6711
DF117    = 070175         9-896   #19-6712
DF12     = 070056         9-896   #19-6672
DF120    = 070175         9-896   #19-6713
DF121    = 070175         9-896   #19-6714
DF122    = 070175         9-896   #19-6715
DF123    = 070151         9-896   #19-6716
DF124    = 070151         9-896   #19-6717
DF125    = 070151         9-896   #19-6718
DF126    = 070151         9-896   #19-6719
DF127      070221         9-896   #19-6720  19-6721  19-6722  19-6724  19-6725  19-6727  19-6738  19-6739
                         19-6740   19-6741  19-6742  19-6743  19-6745  19-6748
DF13     = 070056         9-896   #19-6673
DF130    = 070221         9-896   #19-6721
DF131    = 070221         9-896   #19-6722
DF132    = 070151         9-896   #19-6723
DF133    = 070221         9-896   #19-6724
DF134    = 070221         9-896   #19-6725
DF135    = 070151         9-896   #19-6726
DF136    = 070221         9-896   #19-6727
DF137    = 070175         9-896   #19-6728
DF14     = 070056         9-896   #19-6674
DF140    = 070175         9-896   #19-6729
DF141    = 070175         9-896   #19-6730
DF142    = 070175         9-896   #19-6731
DF143    = 070175         9-896   #19-6732
DF144    = 070175         9-896   #19-6733
DF145    = 070125         9-896   #19-6734
DF146    = 070125         9-896   #19-6735
DF147    = 070125         9-896   #19-6736
DF15     = 070056         9-896   #19-6675
DF150    = 070125         9-896   #19-6737
```

SYMBOL		VALUE	REFERENCES							
DF151	=	070221	9-896	#19-6738						
DF152	=	070221	9-896	#19-6739						
DF153	=	070221	9-896	#19-6740						
DF154	=	070221	9-896	#19-6741						
DF155	=	070221	9-896	#19-6742						
DF156	=	070221	9-896	#19-6743						
DF157	=	070151	9-896	#19-6744						
DF16	=	070056	9-896	#19-6676						
DF160	=	070221	9-896	#19-6745						
DF161	=	070151	9-896	#19-6746						
DF162	=	070151	9-896	#19-6747						
DF163	=	070221	9-896	#19-6748						
DF164	=	070151	9-896	#19-6749						
DF165		070331	9-896	#19-6780	19-6781	19-6782	19-6783	19-6784	19-6785	19-6786	19-6787
			19-6788								
DF166	=	070331	9-896	#19-6781						
DF167	=	070331	9-896	#19-6782						
DF17		070106	9-896	#19-6677						
DF170	=	070331	9-896	#19-6783						
DF171	=	070331	9-896	#19-6784						
DF172	=	070331	9-896	#19-6785						
DF173	=	070331	9-896	#19-6786						
DF174	=	070331	9-896	#19-6787						
DF175	=	070331	9-896	#19-6788						
DF176		070363	9-896	#19-6789	19-6790	19-6791	19-6792	19-6793	19-6794	19-6795	19-6796
			19-6797								
DF177	=	070363	9-896	#19-6790						
DF2		070074	9-896	#19-6664						
DF20		070115	9-896	#19-6678	19-6679	19-6680					
DF200	=	070363	9-896	#19-6791						
DF201	=	070363	9-896	#19-6792						
DF202	=	070363	9-896	#19-6793						
DF203	=	070363	9-896	#19-6794						
DF204	=	070363	9-896	#19-6795						
DF205	=	070363	9-896	#19-6796						
DF206	=	070363	9-896	#19-6797						
DF207		070425	9-896	#19-6799	19-6800	19-6805	19-6807	19-6808	19-6809	19-6810	19-6811
			19-6812								
DF21	=	070115	9-896	#19-6679						
DF210		070415	9-896	#19-6798						
DF211	=	070425	9-896	#19-6800						
DF212		070451	9-896	#19-6801	19-6802	19-6803	19-6804	19-6813	19-6814	19-6815	19-6816
			19-6817								
DF213	=	070451	9-896	#19-6802						
DF214	=	070451	9-896	#19-6803						
DF215	=	070451	9-896	#19-6804						
DF216	=	070425	9-896	#19-6805						
DF217		070501	9-896	#19-6806						
DF22	=	070115	9-896	#19-6680						
DF220	=	070425	9-896	#19-6807						
DF221	=	070425	9-896	#19-6808						
DF222	=	070425	9-896	#19-6809						

SYMBOL	VALUE	REFERENCES							
DF223	= 070425	9-896	#19-6810						
DF224	= 070425	9-896	#19-6811						
DF225	= 070425	9-896	#19-6812						
DF226	= 070451	9-896	#19-6813						
DF227	= 070451	9-896	#19-6814						
DF23	070125	9-896	#19-6681	19-6682	19-6683	19-6684	19-6685	19-6686	19-6687	19-6688
		19-6693	19-6694	19-6695	19-6696	19-6697	19-6698	19-6699	19-6706	19-6707
		19-6708	19-6709	19-6734	19-6735	19-6736	19-6737			
DF230	= 070451	9-896	#19-6815						
DF231	= 070451	9-896	#19-6816						
DF232	= 070451	9-896	#19-6817						
DF233	070510	9-896	#19-6818	19-6825	19-6827	19-6828	19-6829		
DF234	070525	9-896	#19-6819	19-6822	19-6823	19-6824	19-6826		
DF235	070534	9-896	#19-6821						
DF236	= 070525	9-896	#19-6822						
DF237	= 070525	9-896	#19-6823						
DF24	= 070125	9-896	#19-6682						
DF240	= 070525	9-896	#19-6824						
DF241	= 070510	9-896	#19-6825						
DF242	= 070525	9-896	#19-6826						
DF243	= 070510	9-896	#19-6827						
DF244	= 070510	9-896	#19-6828						
DF245	= 070510	9-896	#19-6829						
DF246	= 070151	9-896	#19-6700						
DF247	070540	9-896	#19-6830	19-6831	19-6832				
DF25	= 070125	9-896	#19-6684						
DF250	= 070540	9-896	#19-6831						
DF251	= 070540	9-896	#19-6832						
DF252	070546	9-896	#19-6834	19-6846					
DF253	070552	9-896	#19-6835	19-6836	19-6837	19-6838	19-6839	19-6840	19-6841	19-6842
		19-6843	19-6844	19-6845						
DF254	= 070552	9-896	#19-6836						
DF255	= 070552	9-896	#19-6837						
DF256	= 070552	9-896	#19-6838						
DF257	= 070552	9-896	#19-6839						
DF26	= 070125	9-896	#19-6685						
DF260	= 070552	9-896	#19-6840						
DF261	= 070552	9-896	#19-6841						
DF262	= 070552	9-896	#19-6842						
DF263	= 070552	9-896	#19-6843						
DF264	= 070552	9-896	#19-6844						
DF265	= 070552	9-896	#19-6845						
DF266	= 070546	9-896	#19-6846						
DF27	= 070125	9-896	#19-6686						
DF3	= 070056	9-896	#19-6665						
DF30	= 070125	9-896	#19-6687						
DF31	= 070125	9-896	#19-6688						
DF32	= 070125	9-896	#19-6683						
DF33	070151	9-896	#19-6689	19-6690	19-6691	19-6692	19-6700	19-6701	19-6702	19-6703
		19-6704	19-6705	19-6716	19-6717	19-6718	19-6719	19-6723	19-6726	19-6744
		19-6746	19-6747	19-6749						
DF34	= 070151	9-896	#19-6690							

SYMBOL	VALUE	REFERENCES							
DF35	= 070151	9-896	#19-6691						
DF36	= 070151	9-896	#19-6692						
DF37	= 070125	9-896	#19-6693						
DF4	= 070056	9-896	#19-6666						
DF40	= 070125	9-896	#19-6694						
DF41	= 070125	9-896	#19-6695						
DF42	= 070125	9-896	#19-6696						
DF43	= 070125	9-896	#19-6697						
DF44	= 070125	9-896	#19-6698						
DF45	= 070125	9-896	#19-6699						
DF46	= 070151	9-896	#19-6701						
DF47	= 070151	9-896	#19-6702						
DF5	= 070056	9-896	#19-6667						
DF50	= 070151	9-896	#19-6703						
DF51	= 070151	9-896	#19-6704						
DF52	= 070151	9-896	#19-6705						
DF53	070245	9-896	#19-6750	19-6751	19-6752	19-6753	19-6754	19-6755	19-6756	19-6757
		19-6758	19-6759	19-6760	19-6761	19-6762				
DF54	= 070245	9-896	#19-6751						
DF55	= 070245	9-896	#19-6752						
DF56	= 070245	9-896	#19-6753						
DF57	= 070245	9-896	#19-6754						
DF6	= 070056	9-896	#19-6668						
DF60	= 070245	9-896	#19-6755						
DF61	= 070245	9-896	#19-6756						
DF62	= 070245	9-896	#19-6757						
DF63	= 070245	9-896	#19-6758						
DF64	= 070245	9-896	#19-6759						
DF65	= 070245	9-896	#19-6760						
DF66	= 070245	9-896	#19-6761						
DF67	= 070245	9-896	#19-6762						
DF7	= 070056	9-896	#19-6669						
DF70	070277	9-896	#19-6763	19-6764	19-6765	19-6766	19-6767	19-6768	19-6769	19-6770
		19-6771	19-6772	19-6773	19-6774	19-6775	19-6776	19-6777	19-6778	19-6779
DF71	= 070277	9-896	#19-6764						
DF72	= 070277	9-896	#19-6765						
DF73	= 070277	9-896	#19-6766						
DF74	= 070277	9-896	#19-6767						
DF75	= 070277	9-896	#19-6768						
DF76	= 070277	9-896	#19-6769						
DF77	= 070277	9-896	#19-6770						
DH1	066262	9-896	#18-6455	18-6458	18-6459	18-6460	18-6461	18-6462	18-6463	18-6464
		18-6465	18-6466	18-6467	18-6468	18-6469	18-6471	18-6475	18-6476	18-6477
		18-6478	18-6479	18-6480	18-6481	18-6482	18-6483	18-6484	18-6485	18-6486
		18-6487	18-6488	18-6489	18-6490	18-6491	18-6492	18-6493	18-6494	18-6495
		18-6496	18-6497	18-6498	18-6499	18-6500	18-6501	18-6502	18-6503	18-6511
		18-6512	18-6513	18-6514	18-6518	18-6521	18-6529	18-6530	18-6531	18-6532
		18-6539	18-6541	18-6542	18-6544	18-6545	18-6546	18-6547	18-6548	18-6549
		18-6550	18-6551	18-6552	18-6553	18-6554	18-6555	18-6556	18-6557	18-6558
		18-6559	18-6560	18-6561	18-6562	19-6564	19-6565	19-6566	19-6567	19-6568
		19-6569	19-6570	19-6571	19-6572	19-6573	19-6574	19-6575	19-6648	19-6649
		19-6650	19-6651	19-6652	19-6653	19-6654	19-6655	19-6656	19-6657	19-6658

SYMBOL	VALUE	REFERENCES							
DH10	= 066262	9-896	#18-6463						
DH100	= 066262	9-896	#19-6567						
DH101	= 066262	9-896	#19-6568						
DH102	= 066262	9-896	#19-6569						
DH103	= 066262	9-896	#19-6570						
DH104	= 066262	9-896	#19-6571						
DH105	= 066262	9-896	#19-6572						
DH106	= 066262	9-896	#19-6573						
DH107	= 066262	9-896	#19-6574						
DH11	= 066262	9-896	#18-6464						
DH110	= 066262	9-896	#19-6575						
DH111	= 066262	9-896	#18-6500						
DH112	= 066262	9-896	#18-6501						
DH113	= 066262	9-896	#18-6502						
DH114	= 066262	9-896	#18-6503						
DH115	066623	9-896	#18-6504	18-6506	18-6507	18-6508	18-6509	18-6510	18-6515	18-6516
		18-6517	18-6519	18-6520	18-6522	18-6523	18-6524	18-6525	18-6526	18-6527
		18-6528	18-6533	18-6534	18-6535	18-6536	18-6537	18-6538	18-6540	18-6543
		19-6576	19-6577	19-6578	19-6579	19-6580	19-6581	19-6582	19-6583	19-6584
		19-6585	19-6586	19-6587	19-6588	19-6589	19-6590	19-6591	19-6592	19-6593
DH116	= 066623	9-896	#18-6506						
DH117	= 066623	9-896	#18-6507						
DH12	= 066262	9-896	#18-6465						
DH120	= 066623	9-896	#18-6508						
DH121	= 066623	9-896	#18-6509						
DH122	= 066623	9-896	#18-6510						
DH123	= 066262	9-896	#18-6511						
DH124	= 066262	9-896	#18-6512						
DH125	= 066262	9-896	#18-6513						
DH126	= 066262	9-896	#18-6514						
DH127	= 066623	9-896	#18-6515						
DH13	= 066262	9-896	#18-6466						
DH130	= 066623	9-896	#18-6516						
DH131	= 066623	9-896	#18-6517						
DH132	= 066262	9-896	#18-6518						
DH133	= 066623	9-896	#18-6519						
DH134	= 066623	9-896	#18-6520						
DH135	= 066262	9-896	#18-6521						
DH136	= 066623	9-896	#18-6522						
DH137	= 066623	9-896	#18-6523						
DH14	= 066262	9-896	#18-6467						
DH140	= 066623	9-896	#18-6524						
DH141	= 066623	9-896	#18-6525						
DH142	= 066623	9-896	#18-6526						
DH143	= 066623	9-896	#18-6527						
DH144	= 066623	9-896	#18-6528						
DH145	= 066262	9-896	#18-6529						
DH146	= 066262	9-896	#18-6530						
DH147	= 066262	9-896	#18-6531						
DH15	= 066262	9-896	#18-6468						
DH150	= 066262	9-896	#18-6532						
DH151	= 066623	9-896	#18-6533						

SYMBOL	VALUE	REFERENCES					
DH152	= 066623	9-896	#18-6534				
DH153	= 066623	9-896	#18-6535				
DH154	= 066623	9-896	#18-6536				
DH155	= 066623	9-896	#18-6537				
DH156	= 066623	9-896	#18-6538				
DH157	= 066262	9-896	#18-6539				
DH16	= 066262	9-896	#18-6469				
DH160	= 066623	9-896	#18-6540				
DH161	= 066262	9-896	#18-6541				
DH162	= 066262	9-896	#18-6542				
DH163	= 066623	9-896	#18-6543				
DH164	= 066262	9-896	#18-6544				
DH165	= 066623	9-896	#19-6576				
DH166	= 066623	9-896	#19-6577				
DH167	= 066623	9-896	#19-6578				
DH17	066413	9-896	#18-6470				
DH170	= 066623	9-896	#19-6579				
DH171	= 066623	9-896	#19-6580				
DH172	= 066623	9-896	#19-6581				
DH173	= 066623	9-896	#19-6582				
DH174	= 066623	9-896	#19-6583				
DH175	= 066623	9-896	#19-6584				
DH176	= 066623	9-896	#19-6585				
DH177	= 066623	9-896	#19-6586				
DH2	066352	9-896	#18-6457	19-6647			
DH20	= 066262	9-896	#18-6471				
DH200	= 066623	9-896	#19-6587				
DH201	= 066623	9-896	#19-6588				
DH202	= 066623	9-896	#19-6589				
DH203	= 066623	9-896	#19-6590				
DH204	= 066623	9-896	#19-6591				
DH205	= 066623	9-896	#19-6592				
DH206	= 066623	9-896	#19-6593				
DH207	066762	9-896	#19-6595	19-6599	19-6600	19-6601	19-6602
DH21	066466	9-896	#18-6472				
DH210	067052	9-896	#19-6597	19-6598	19-6603	19-6606	19-6608	19-6610
DH211	= 067052	9-896	#19-6598				
DH212	= 066762	9-896	#19-6599				
DH213	= 066762	9-896	#19-6600				
DH214	= 066762	9-896	#19-6601				
DH215	= 066762	9-896	#19-6602				
DH216	= 067052	9-896	#19-6603				
DH217	067113	9-896	#19-6604				
DH22	066555	9-896	#18-6474				
DH220	066722	9-896	#19-6594	19-6607	19-6609		
DH221	= 067052	9-896	#19-6606				
DH222	= 066722	9-896	#19-6607				
DH223	= 067052	9-896	#19-6608				
DH224	= 066722	9-896	#19-6609				
DH225	= 067052	9-896	#19-6610				
DH226	067166	9-896	#19-6611	19-6615			
DH227	067255	9-896	#19-6613	19-6616	19-6617		

SYMBOL	VALUE	REFERENCES					
DH23	= 066262	9-896	#18-6475				
DH230	= 067166	9-896	#19-6615				
DH231	= 067255	9-896	#19-6616				
DH232	= 067255	9-896	#19-6617				
DH233	067344	9-896	#19-6620	19-6638	19-6640	19-6641	19-6642
DH234	067405	9-896	#19-6622	19-6632			
DH235	067473	9-896	#19-6625				
DH236	067533	9-896	#19-6628				
DH237	= 067405	9-896	#19-6632				
DH24	= 066262	9-896	#18-6477				
DH240	067621	9-896	#19-6634	19-6639			
DH241	= 067344	9-896	#19-6638				
DH242	= 067621	9-896	#19-6639				
DH243	= 067344	9-896	#19-6640				
DH244	= 067344	9-896	#19-6641				
DH245	= 067344	9-896	#19-6642				
DH246	= 066262	9-896	#18-6494				
DH247	067711	9-896	#19-6643				
DH25	= 066262	9-896	#18-6478				
DH250	067756	9-896	#19-6644	19-6645			
DH251	= 067756	9-896	#19-6645				
DH252	= 066352	9-896	#19-6647				
DH253	= 066262	9-896	#19-6648				
DH254	= 066262	9-896	#19-6649				
DH255	= 066262	9-896	#19-6650				
DH256	= 066262	9-896	#19-6651				
DH257	= 066262	9-896	#19-6652				
DH26	= 066262	9-896	#18-6479				
DH260	= 066262	9-896	#19-6653				
DH261	= 066262	9-896	#19-6654				
DH262	= 066262	9-896	#19-6655				
DH263	= 066262	9-896	#19-6656				
DH264	= 066262	9-896	#19-6657				
DH265	= 066262	9-896	#19-6658				
DH266	070016	9-896	#19-6659				
DH27	= 066262	9-896	#18-6480				
DH3	= 066262	9-896	#18-6458				
DH30	= 066262	9-896	#18-6481				
DH31	= 066262	9-896	#18-6482				
DH32	= 066262	9-896	#18-6476				
DH33	= 066262	9-896	#18-6483				
DH34	= 066262	9-896	#18-6484				
DH35	= 066262	9-896	#18-6485				
DH36	= 066262	9-896	#18-6486				
DH37	= 066262	9-896	#18-6487				
DH4	= 066262	9-896	#18-6459				
DH40	= 066262	9-896	#18-6488				
DH41	= 066262	9-896	#18-6489				
DH42	= 066262	9-896	#18-6490				
DH43	= 066262	9-896	#18-6491				
DH44	= 066262	9-896	#18-6492				
DH45	= 066262	9-896	#18-6493				

SYMBOL	VALUE	REFERENCES								
DH46	= 066262	9-896	#18-6495							
DH47	= 066262	9-896	#18-6496							
DH5	= 066262	9-896	#18-6460							
DH50	= 066262	9-896	#18-6497							
DH51	= 066262	9-896	#18-6498							
DH52	= 066262	9-896	#18-6499							
DH53	= 066262	9-896	#18-6545							
DH54	= 066262	9-896	#18-6546							
DH55	= 066262	9-896	#18-6547							
DH56	= 066262	9-896	#18-6548							
DH57	= 066262	9-896	#18-6549							
DH6	= 066262	9-896	#18-6461							
DH60	= 066262	9-896	#18-6550							
DH61	= 066262	9-896	#18-6551							
DH62	= 066262	9-896	#18-6552							
DH63	= 066262	9-896	#18-6553							
DH64	= 066262	9-896	#18-6554							
DH65	= 066262	9-896	#18-6555							
DH66	= 066262	9-896	#18-6556							
DH67	= 066262	9-896	#18-6557							
DH7	= 066262	9-896	#18-6462							
DH70	= 066262	9-896	#18-6558							
DH71	= 066262	9-896	#18-6559							
DH72	= 066262	9-896	#18-6560							
DH73	= 066262	9-896	#18-6561							
DH74	= 066262	9-896	#18-6562							
DH75	= 066262	9-896	#19-6564							
DH76	= 066262	9-896	#19-6565							
DH77	= 066262	9-896	#19-6566							
DISPLA	001142	#8-890	*9-903	*9-903	17-5645	17-5647				
DISPRE	000174	#7-889	9-903							
DIVDSU	016614	12-2521	12-2532	12-2543	12-2554	12-2565	12-2578	12-2590	#13-2631	
DIVDT	017044	13-2650	13-2659	13-2665	13-2677	#13-2695				
DIVFSU	015674	12-2275	12-2286	12-2297	12-2309	12-2320	12-2331	12-2342	12-2354	12-2366
		12-2377	12-2388	12-2400	12-2411	#12-2448				
DIVFT	016110	12-2465	12-2474	#12-2505						
DSWR	= 177570	#7-877	8-890	9-903						
DT1	070604	9-896	#19-6853	19-6856	19-6857	19-6858	19-6859	19-6860	19-6861	19-6862
		19-6863	19-6864	19-6865	19-6866	19-6867				
DT10	= 070604	9-896	#19-6861							
DT100	= 071120	9-896	#19-6968							
DT101	= 071120	9-896	#19-6969							
DT102	= 071120	9-896	#19-6970							
DT103	= 071120	9-896	#19-6971							
DT104	= 071120	9-896	#19-6972							
DT105	= 071120	9-896	#19-6973							
DT106	= 071120	9-896	#19-6974							
DT107	= 071120	9-896	#19-6975							
DT11	= 070604	9-896	#19-6862							
DT110	= 071120	9-896	#19-6976							
DT111	= 070772	9-896	#19-6898							
DT112	= 070772	9-896	#19-6899							

SYMBOL	VALUE	REFERENCES							
DT113	= 070772	9-896	#19-6900						
DT114	= 070772	9-896	#19-6901						
DT115	071044	9-896	#19-6902	19-6905	19-6906	19-6907	19-6908	19-6909	19-6914	19-6915
		19-6916	19-6918	19-6919	19-6921	19-6922	19-6923	19-6924	19-6925	19-6926
		19-6927	19-6932	19-6933	19-6934	19-6935	19-6936	19-6937	19-6939	19-6942
DT116	= 071044	9-896	#19-6905							
DT117	= 071044	9-896	#19-6906							
DT12	= 070604	9-896	#19-6863							
DT120	= 071044	9-896	#19-6907							
DT121	= 071044	9-896	#19-6908							
DT122	= 071044	9-896	#19-6909							
DT123	= 070772	9-896	#19-6910							
DT124	= 070772	9-896	#19-6911							
DT125	= 070772	9-896	#19-6912							
DT126	= 070772	9-896	#19-6913							
DT127	= 071044	9-896	#19-6914							
DT13	= 070604	9-896	#19-6864							
DT130	= 071044	9-896	#19-6915							
DT131	= 071044	9-896	#19-6916							
DT132	= 070772	9-896	#19-6917							
DT133	= 071044	9-896	#19-6918							
DT134	= 071044	9-896	#19-6919							
DT135	= 070772	9-896	#19-6920							
DT136	= 071044	9-896	#19-6921							
DT137	= 071044	9-896	#19-6922							
DT14	= 070604	9-896	#19-6865							
DT140	= 071044	9-896	#19-6923							
DT141	= 071044	9-896	#19-6924							
DT142	= 071044	9-896	#19-6925							
DT143	= 071044	9-896	#19-6926							
DT144	= 071044	9-896	#19-6927							
DT145	= 070772	9-896	#19-6928							
DT146	= 070772	9-896	#19-6929							
DT147	= 070772	9-896	#19-6930							
DT15	= 070604	9-896	#19-6866							
DT150	= 070772	9-896	#19-6931							
DT151	= 071044	9-896	#19-6932							
DT152	= 071044	9-896	#19-6933							
DT153	= 071044	9-896	#19-6934							
DT154	= 071044	9-896	#19-6935							
DT155	= 071044	9-896	#19-6936							
DT156	= 071044	9-896	#19-6937							
DT157	= 070772	9-896	#19-6938							
DT16	= 070604	9-896	#19-6867							
DT160	= 071044	9-896	#19-6939							
DT161	= 070772	9-896	#19-6940							
DT162	= 070772	9-896	#19-6941							
DT163	= 071044	9-896	#19-6942							
DT164	= 070772	9-896	#19-6943							
DT165	071206	9-896	#19-6977	19-6981	19-6982	19-6983	19-6984	19-6985	19-6986	19-6987
		19-6988	19-6989	19-6990	19-6991	19-6992	19-6993	19-6994	19-6995	19-6996
		19-6997								

SYMBOL VALUE REFERENCES

DT166 = 071206 9-896 #19-6981
DT167 = 071206 9-896 #19-6982
DT17 070670 9-896 #19-6868
DT170 = 071206 9-896 #19-6983
DT171 = 071206 9-896 #19-6984
DT172 = 071206 9-896 #19-6985
DT173 = 071206 9-896 #19-6986
DT174 = 071206 9-896 #19-6987
DT175 = 071206 9-896 #19-6988
DT176 = 071206 9-896 #19-6989
DT177 = 071206 9-896 #19-6990
DT2 070642 9-896 #19-6855
DT20 070710 9-896 #19-6869
DT200 = 071206 9-896 #19-6991
DT201 = 071206 9-896 #19-6992
DT202 = 071206 9-896 #19-6993
DT203 = 071206 9-896 #19-6994
DT204 = 071206 9-896 #19-6995
DT205 = 071206 9-896 #19-6996
DT206 = 071206 9-896 #19-6997
DT207 071316 9-896 #19-7001 19-7004 19-7011 19-7013 19-7014 19-7015 19-7016 19-7017
19-7018 19-7023
DT21 070732 9-896 #19-6870
DT210 071274 9-896 #19-6999 19-7022
DT211 = 071316 9-896 #19-7004
DT212 071370 9-896 #19-7005 19-7008 19-7009 19-7010 19-7019 19-7020 19-7021
DT213 = 071370 9-896 #19-7008
DT214 = 071370 9-896 #19-7009
DT215 = 071370 9-896 #19-7010
DT216 = 071316 9-896 #19-7011
DT217 071452 9-896 #19-7012
DT22 070754 9-896 #19-6871
DT220 = 071316 9-896 #19-7013
DT221 = 071316 9-896 #19-7014
DT222 = 071316 9-896 #19-7015
DT223 = 071316 9-896 #19-7016
DT224 = 071316 9-896 #19-7017
DT225 = 071316 9-896 #19-7018
DT226 = 071370 9-896 #19-7019
DT227 = 071370 9-896 #19-7020
DT23 070772 9-896 #19-6872 19-6874 19-6875 19-6876 19-6877 19-6878 19-6879 19-6880
19-6881 19-6882 19-6883 19-6884 19-6885 19-6886 19-6887 19-6888 19-6889
19-6890 19-6891 19-6892 19-6893 19-6894 19-6895 19-6896 19-6897 19-6898
19-6899 19-6900 19-6901 19-6910 19-6911 19-6912 19-6913 19-6917 19-6920
19-6928 19-6929 19-6930 19-6931 19-6938 19-6940 19-6941 19-6943
DT230 = 071370 9-896 #19-7021
DT231 = 071274 9-896 #19-7022
DT232 = 071316 9-896 #19-7023
DT233 071472 9-896 #19-7029 19-7040 19-7042 19-7043 19-7044
DT234 071526 9-896 #19-7033 19-7037 19-7038 19-7039 19-7041
DT235 071546 9-896 #19-7035
DT236 = 071526 9-896 #19-7037

SYMBOL	VALUE	REFERENCES							
DT237	= 071526	9-896	#19-7038						
DT24	= 070772	9-896	#19-6875						
DT240	= 071526	9-896	#19-7039						
DT241	= 071472	9-896	#19-7040						
DT242	= 071526	9-896	#19-7041						
DT243	= 071472	9-896	#19-7042						
DT244	= 071472	9-896	#19-7043						
DT245	= 071472	9-896	#19-7044						
DT246	= 070772	9-896	#19-6892						
DT247	071560	9-896	#19-7045						
DT25	= 070772	9-896	#19-6876						
DT250	071576	9-896	#19-7046	19-7047					
DT251	= 071576	9-896	#19-7047						
DT252	071610	9-896	#19-7049	19-7066					
DT253	071622	9-896	#19-7051	19-7055	19-7056	19-7057	19-7058	19-7059	19-7060	19-7061
		19-7062	19-7063	19-7064						
DT254	= 071622	9-896	#19-7055						
DT255	= 071622	9-896	#19-7056						
DT256	= 071622	9-896	#19-7057						
DT257	= 071622	9-896	#19-7058						
DT26	= 070772	9-896	#19-6877						
DT260	= 071622	9-896	#19-7059						
DT261	= 071622	9-896	#19-7060						
DT262	= 071622	9-896	#19-7061						
DT263	= 071622	9-896	#19-7062						
DT264	= 071622	9-896	#19-7063						
DT265	= 071622	9-896	#19-7064						
DT266	= 071610	9-896	#19-7066						
DT27	= 070772	9-896	#19-6878						
DT3	= 070604	9-896	#19-6856						
DT30	= 070772	9-896	#19-6879						
DT31	= 070772	9-896	#19-6880						
DT32	= 070772	9-896	#19-6874						
DT33	= 070772	9-896	#19-6881						
DT34	= 070772	9-896	#19-6882						
DT35	= 070772	9-896	#19-6883						
DT36	= 070772	9-896	#19-6884						
DT37	= 070772	9-896	#19-6885						
DT4	= 070604	9-896	#19-6857						
DT40	= 070772	9-896	#19-6886						
DT41	= 070772	9-896	#19-6887						
DT42	= 070772	9-896	#19-6888						
DT43	= 070772	9-896	#19-6889						
DT44	= 070772	9-896	#19-6890						
DT45	= 070772	9-896	#19-6891						
DT46	= 070772	9-896	#19-6893						
DT47	= 070772	9-896	#19-6894						
DT5	= 070604	9-896	#19-6858						
DT50	= 070772	9-896	#19-6895						
DT51	= 070772	9-896	#19-6896						
DT52	= 070772	9-896	#19-6897						
DT53	071120	9-896	#19-6944	19-6948	19-6949	19-6950	19-6951	19-6952	19-6953	19-6954

SYMBOL	VALUE	REFERENCES								
		19-6955	19-6956	19-6957	19-6958	19-6959	19-6960	19-6961	19-6962	19-6963
		19-6964	19-6965	19-6966	19-6967	19-6968	19-6969	19-6970	19-6971	19-6972
		19-6973	19-6974	19-6975	19-6976					
DT54	= 071120	9-896	#19-6948							
DT55	= 071120	9-896	#19-6949							
DT56	= 071120	9-896	#19-6950							
DT57	= 071120	9-896	#19-6951							
DT6	= 070604	9-896	#19-6859							
DT60	= 071120	9-896	#19-6952							
DT61	= 071120	9-896	#19-6953							
DT62	= 071120	9-896	#19-6954							
DT63	= 071120	9-896	#19-6955							
DT64	= 071120	9-896	#19-6956							
DT65	= 071120	9-896	#19-6957							
DT66	= 071120	9-896	#19-6958							
DT67	= 071120	9-896	#19-6959							
DT7	= 070604	9-896	#19-6860							
DT70	= 071120	9-896	#19-6960							
DT71	= 071120	9-896	#19-6961							
DT72	= 071120	9-896	#19-6962							
DT73	= 071120	9-896	#19-6963							
DT74	= 071120	9-896	#19-6964							
DT75	= 071120	9-896	#19-6965							
DT76	= 071120	9-896	#19-6966							
DT77	= 071120	9-896	#19-6967							
EEEDON	020064	13-2855	#13-2942							
EEE1	017060	#13-2711								
EEE10	017454	#13-2812								
EEE11	017510	#13-2823								
EEE12	017544	#13-2834								
EEE13	017600	#13-2845								
EEE2	017114	#13-2722								
EEE3	017150	#13-2733								
EEE4	017204	#13-2744								
EEE5	017240	#13-2755								
EEE6	017274	#13-2767								
EEE7	017330	#13-2779								
EEE8	017364	#13-2790								
EEE9	017420	#13-2801								
EMTVEC	= 000030	#7-877	9-903	9-903						
EM1	037724	9-896	#18-5937							
EM10	040543	9-896	#18-5952							
EM100	053701	9-896	#18-6175							
EM101	053764	9-896	#18-6177							
EM102	054160	9-896	#18-6180							
EM103	054354	9-896	#18-6183							
EM104	054466	9-896	#18-6185							
EM105	054633	9-896	#18-6188							
EM106	054742	9-896	#18-6190							
EM107	055051	9-896	#18-6192							
EM11	040650	9-896	#18-5954							
EM110	055160	9-896	#18-6194							

CKFPBA0 CREATED BY MACRO ON 18-SEP-79 AT 13:50 PAGE 18
SYMBOL CROSS REFERENCE CREF SEQ 0189

SYMBOL		VALUE	REFERENCES		
EM111		043775	9-896	#18-6026	18-6078
EM112		044052	9-896	#18-6027	18-6079
EM113		044130	9-896	#18-6028	
EM114		044206	9-896	#18-6030	
EM115		044265	9-896	#18-6032	
EM116		044343	9-896	#18-6033	
EM117		044422	9-896	#18-6040	
EM12		040757	9-896	#18-5956	
EM120		044560	9-896	#18-6042	
EM121		044716	9-896	#18-6044	
EM122		045053	9-896	#18-6046	
EM123		045210	9-896	#18-6048	18-6100
EM124		045266	9-896	#18-6049	18-6101
EM125		045345	9-896	#18-6050	
EM126		045423	9-896	#18-6052	
EM127		045502	9-896	#18-6054	
EM13		041066	9-896	#18-5958	
EM130		045560	9-896	#18-6055	
EM131		045637	9-896	#18-6062	
EM132		045775	9-896	#18-6064	
EM133		046133	9-896	#18-6067	
EM134		046271	9-896	#18-6069	
EM135		046426	9-896	#18-6071	
EM136		046563	9-896	#18-6074	
EM137		046720	9-896	#18-6076	
EM14		041202	9-896	#18-5960	
EM140		047006	9-896	#18-6077	
EM141	=	043775	9-896	#18-6078	
EM142	=	044052	9-896	#18-6079	
EM143		047075	9-896	#18-6080	
EM144		047153	9-896	#18-6082	
EM145		047232	9-896	#18-6090	
EM146		047377	9-896	#18-6092	
EM147		047544	9-896	#18-6094	
EM15		041314	9-896	#18-5962	
EM150		047710	9-896	#18-6096	
EM151		050054	9-896	#18-6098	
EM152		050142	9-896	#18-6099	
EM153	=	045210	9-896	#18-6100	
EM154	=	045266	9-896	#18-6101	
EM155		050231	9-896	#18-6102	
EM156		050307	9-896	#18-6104	
EM157		050366	9-896	#18-6112	
EM16		041426	9-896	#18-5964	
EM160		050533	9-896	#18-6114	
EM161		050671	9-896	#18-6117	
EM162		051036	9-896	#18-6119	
EM163		051202	9-896	#18-6121	
EM164		051337	9-896	#18-6124	
EM165		055270	9-896	#18-6208	
EM166		055354	9-896	#18-6210	
EM167		055437	9-896	#18-6212	

SYMBOL		VALUE	REFERENCES		
EM17		041523	9-896	#18-5966	
EM170		055471	9-896	#18-6214	
EM171		055557	9-896	#18-6216	
EM172		055702	9-896	#18-6218	
EM173		055767	9-896	#18-6220	
EM174		056135	9-896	#18-6222	
EM175		056303	9-896	#18-6224	
EM176		056415	9-896	#18-6226	
EM177		056501	9-896	#18-6228	
EM2		037756	9-896	#18-5938	
EM20		041600	9-896	#18-5967	
EM200		056564	9-896	#18-6230	
EM201		056616	9-896	#18-6232	
EM202		056705	9-896	#18-6234	
EM203		057047	9-896	#18-6236	
EM204		057211	9-896	#18-6238	
EM205		057277	9-896	#18-6240	
EM206		057445	9-896	#18-6242	
EM207		057613	9-896	#18-6338	
EM21		041632	9-896	#18-5968	
EM210		057655	9-896	#18-6339	18-6385
EM211		057717	9-896	#18-6340	
EM212		060063	9-896	#18-6367	
EM213		060247	9-896	#18-6368	
EM214		060433	9-896	#18-6369	
EM215		060617	9-896	#18-6370	
EM216		061003	9-896	#18-6371	
EM217		061165	9-896	#18-6374	
EM22		041664	9-896	#18-5969	
EM220		061227	9-896	#18-6375	
EM221		061457	9-896	#18-6376	
EM222		061723	9-896	#18-6377	
EM223		062154	9-896	#18-6378	
EM224		062421	9-896	#18-6379	
EM225		062652	9-896	#18-6380	
EM226		063117	9-896	#18-6381	
EM227		063252	9-896	#18-6382	
EM23		041745	9-896	#18-5970	
EM230		063405	9-896	#18-6383	
EM231		063541	9-896	#18-6384	
EM232	=	057655	9-896	#18-6385	
EM233		063675	9-896	#18-6415	
EM234		063734	9-896	#18-6416	
EM235		064020	9-896	#18-6417	
EM236		064066	9-896	#18-6418	
EM237		064121	9-896	#18-6419	
EM24		042022	9-896	#18-5975	
EM240		064205	9-896	#18-6420	
EM241		064241	9-896	#18-6421	
EM242		064344	9-896	#18-6422	
EM243		064400	9-896	#18-6423	
EM244		064503	9-896	#18-6424	

SYMBOL	VALUE	REFERENCES		
EM245	064542	9-896	#18-6426	
EM246	043405	9-896	#18-6016	
EM247	064624	9-896	#18-6429	
EM25	042120	9-896	#18-5977	18-5981
EM250	064660	9-896	#18-6430	
EM251	064712	9-896	#18-6431	
EM252	064745	9-896	#18-6433	
EM253	065036	9-896	#18-6439	
EM254	065122	9-896	#18-6440	
EM255	065207	9-896	#18-6441	
EM256	065275	9-896	#18-6442	
EM257	065364	9-896	#18-6443	
EM26	042205	9-896	#18-5979	
EM260	065452	9-896	#18-6444	
EM261	065541	9-896	#18-6445	
EM262	065631	9-896	#18-6446	
EM263	065722	9-896	#18-6447	
EM264	066007	9-896	#18-6448	
EM265	066074	9-896	#18-6449	
EM266	066161	9-896	#18-6451	
EM27	= 042120	9-896	#18-5981	
EM3	040012	9-896	#18-5942	
EM30	042303	9-896	#18-5982	
EM31	042355	9-896	#18-5984	
EM32	041770	9-896	#18-5971	
EM33	042422	9-896	#18-5986	
EM34	042445	9-896	#18-5987	
EM35	042477	9-896	#18-5991	
EM36	042552	9-896	#18-5993	
EM37	042620	9-896	#18-5995	
EM4	040117	9-896	#18-5944	
EM40	042643	9-896	#18-5999	
EM41	042675	9-896	#18-6000	
EM42	042750	9-896	#18-6002	
EM43	043101	9-896	#18-6005	
EM44	043232	9-896	#18-6008	
EM45	043300	9-896	#18-6010	
EM46	043353	9-896	#18-6012	
EM47	043430	9-896	#18-6017	
EM5	040224	9-896	#18-5946	
EM50	043564	9-896	#18-6020	
EM51	043637	9-896	#18-6022	
EM52	043705	9-896	#18-6024	
EM53	051535	9-896	#18-6127	
EM54	051566	9-896	#18-6128	
EM55	051503	9-896	#18-6126	
EM56	051616	9-896	#18-6135	
EM57	051707	9-896	#18-6137	
EM6	040331	9-896	#18-5948	
EM60	051777	9-896	#18-6139	
EM61	052101	9-896	#18-6141	
EM62	052275	9-896	#18-6144	

CKFPBA0 CREATED BY MACRO ON 18-SEP-79 AT 13:50 PAGE 21
SYMBOL CROSS REFERENCE CREF SEQ 0192

SYMBOL	VALUE	REFERENCES								
EM63	052471	9-896	#18-6147							
EM64	052602	9-896	#18-6149							
EM65	052721	9-896	#18-6151							
EM66	053034	9-896	#18-6153							
EM67	053103	9-896	#18-6155							
EM7	040436	9-896	#18-5950							
EM70	053166	9-896	#18-6157							
EM71	053217	9-896	#18-6158							
EM72	053247	9-896	#18-6159	18-6160						
EM73	= 053247	9-896	#18-6160							
EM74	053301	9-896	#18-6167							
EM75	053410	9-896	#18-6169							
EM76	053501	9-896	#18-6171							
EM77	053571	9-896	#18-6173							
ERM10	034152	17-5647	17-5647	#17-5647						
ERROR	= 104000	#7-877	9-1055	9-1056	9-1057	9-1058	9-1059	9-1060	9-1061	9-1062
		9-1063	9-1064	9-1065	9-1066	9-1079	11-1363	11-1365	11-1367	11-1369
		11-1373	11-1375	11-1389	11-1391	11-1393	11-1395	11-1397	11-1399	11-1401
		11-1416	11-1418	11-1420	11-1422	11-1424	11-1426	11-1428	11-1692	11-1697
		11-1703	11-1709	11-1714	11-1719	11-1724	11-1730	11-1741	11-1750	11-1759
		11-1828	11-1842	11-1855	11-1868	11-1882	11-1898	11-1915	11-1932	11-1949
		12-1966	12-1983	12-2000	12-2017	12-2083	12-2099	12-2222	12-2231	12-2239
		12-2252	12-2282	12-2293	12-2305	12-2316	12-2327	12-2338	12-2349	12-2361
		12-2373	12-2384	12-2395	12-2407	12-2418	12-2498	12-2502	12-2528	12-2539
		12-2550	12-2561	12-2573	12-2586	12-2599	13-2688	13-2692	13-2719	13-2730
		13-2741	13-2752	13-2763	13-2775	13-2787	13-2798	13-2809	13-2820	13-2831
		13-2842	13-2853	13-2933	13-2937	13-2964	13-2976	13-2987	13-2999	13-3086
		13-3090	13-3121	13-3123	13-3137	13-3140	13-3154	13-3156	13-3170	13-3173
		13-3269	13-3273	14-3299	14-3303	14-3335	14-3339	14-3373	14-3375	14-3390
		14-3392	14-3407	14-3409	14-3425	14-3427	14-3521	14-3525	14-3550	14-3554
		14-3586	14-3590	14-3625	14-3627	14-3641	14-3643	14-3657	14-3659	14-3673
		14-3675	14-3778	14-3782	14-3790	14-3794	14-3819	14-3823	14-3870	14-3872
		14-3888	14-3890	14-3906	14-3908	14-3924	14-3926	15-4030	15-4034	15-4042
		15-4046	15-4071	15-4075	15-4120	15-4122	15-4136	15-4138	15-4152	15-4154
		15-4168	15-4170	15-4184	15-4186	15-4200	15-4203	15-4217	15-4220	15-4234
		15-4236	15-4250	15-4252	15-4266	15-4268	15-4282	15-4284	15-4298	15-4300
		15-4314	15-4316	15-4330	15-4332	15-4433	15-4447	15-4453	15-4487	15-4489
		15-4505	15-4507	15-4522	15-4524	15-4540	15-4542	15-4559	15-4561	15-4575
		15-4578	16-4593	16-4596	16-4610	16-4612	16-4628	16-4631	16-4648	16-4650
		16-4666	16-4668	16-4684	16-4686	16-4700	16-4702	16-4810	16-4827	16-4833
		16-4867	16-4869	16-4886	16-4901	16-4903	16-4921	16-4937	16-4939	16-5037
		16-5051	16-5057	16-5100	16-5102	16-5121	16-5138	16-5140	16-5159	17-5262
		17-5279	17-5285	17-5376	17-5378	17-5389	17-5391	17-5403	17-5405	17-5417
		17-5419	17-5433	17-5435	17-5447	17-5449	17-5462	17-5464	17-5475	17-5477
		17-5491	17-5493	17-5505	17-5507	17-5519	17-5521	17-5617	17-5838	17-5849
		17-5860								
ERRVEC	= 000004	#7-877	9-903	9-903	9-903	11-1566	11-1572	11-1584	17-5352	17-5354
		17-5358	17-5645	17-5645	17-5645	17-5645	17-5891			
ERTYPE	036204	17-5647	#17-5675							
ERT1	036370	#17-5726	17-5813							
ERT2	036624	17-5731	17-5744	17-5764	17-5773	17-5801	#17-5805			
ERT3	036630	17-5781	17-5789	#17-5810						

SYMBOL	VALUE	REFERENCES								
ERT4	036642	17-5722	17-5812	#17-5815						
ERT5	036654	17-5688	#17-5820							
FFFDON	020614	13-3001	#13-3095							
FFF1	020070	#13-2956								
FFF2	020144	#13-2968								
FFF3	020220	#13-2979								
FFF4	020274	#13-2990								
FPSPUR	036660	9-1014	9-1071	9-1193	9-1258	11-1326	11-1355	11-1381	11-1407	12-2189
		12-2212	14-3310	14-3561	14-3746	15-3998	#17-5832	17-5890		
FPVECT	= 000244	#7-871	9-960	9-1014	9-1156	9-1180	9-1221	9-1245	9-1287	11-1313
		12-2126	12-2176	13-3235	14-3487	14-3733	15-3985	17-5890		
GGDATO	011540	9-1162	9-1186	9-1197	9-1227	9-1251	9-1262	9-1293	11-1319	11-1330
		11-1361	11-1363	11-1365	11-1367	11-1369	11-1373	11-1375	11-1387	11-1389
		11-1391	11-1393	11-1395	11-1397	11-1399	11-1401	11-1413	11-1416	11-1418
		11-1420	11-1422	11-1424	11-1426	11-1428	#11-1430			
GGDONE	011660	11-1349	11-1363	11-1365	11-1367	11-1369	11-1373	11-1375	11-1389	11-1391
		11-1393	11-1395	11-1397	11-1399	11-1401	11-1416	11-1418	11-1420	11-1422
		11-1424	11-1426	11-1428	#11-1472					
GGER0	010036	9-1156	#11-1351							
GGER1	010134	9-1168	#11-1365	11-1371						
GGER10	010646	9-1253	#11-1395							
GGER11	010714	9-1268	#11-1397							
GGER12	010762	9-1273	#11-1399							
GGER13	011030	9-1280	#11-1401							
GGER14	011076	9-1287	#11-1403							
GGER15	011174	9-1299	#11-1418							
GGER16	011242	9-1304	#11-1420							
GGER17	011310	11-1321	#11-1422							
GGER18	011356	11-1336	#11-1424							
GGER19	011424	11-1347	#11-1426							
GGER2	010202	9-1173	#11-1367							
GGER20	011472	11-1341	#11-1428							
GGER3	010250	9-1188	#11-1369							
GGER4	010316	9-1203	#11-1371							
GGER5	010320	9-1214	#11-1373							
GGER6	010366	9-1208	#11-1375							
GGER7	010434	9-1221	#11-1377							
GGER8	010532	9-1233	#11-1391							
GGER9	010600	9-1238	#11-1393							
GGGDON	025676	15-4333	#15-4462							
GGG1	024224	#15-4110								
GGG10	024774	#15-4256								
GGG11	025044	#15-4272								
GGG12	025114	#15-4288								
GGG13	025164	#15-4304								
GGG14	025234	#15-4320								
GGG2	024274	#15-4126								
GGG3	024344	#15-4142								
GGG4	024414	#15-4158								
GGG5	024464	#15-4174								
GGG6	024534	#15-4190								
GGG7	024604	#15-4207								

SYMBOL	VALUE	REFERENCES								
GGG8	024654	#15-4224								
GGG9	024724	#15-4240								
GGP1	011550	11-1416	#11-1435							
GGP2	011560	9-1222	9-1246	9-1288	11-1314	11-1389	11-1391	11-1393	11-1395	11-1397
		11-1399	11-1401	11-1418	11-1420	11-1422	11-1424	11-1426	11-1428	#11-1439
GGP3	011570	9-1224	9-1248	11-1389	11-1391	11-1393	11-1395	11-1397	11-1399	11-1401
		11-1416	#11-1443							
GGP4	011600	#11-1447								
GGP5	011610	9-1157	9-1159	9-1181	9-1183	11-1363	11-1363	11-1365	11-1365	11-1367
		11-1367	11-1369	11-1369	11-1373	11-1373	11-1375	11-1375	#11-1451	
GGP6	011620	9-1164	9-1199	9-1229	9-1295	11-1363	11-1365	11-1367	11-1369	11-1373
		11-1375	11-1389	11-1391	11-1393	11-1416	11-1418	11-1420	#11-1455	
GGP7	011630	9-1264	11-1395	11-1397	11-1399	11-1401	#11-1460			
GGP8	011640	9-1290	11-1316	11-1418	11-1420	11-1422	11-1424	11-1426	11-1428	#11-1464
GGP9	011650	11-1332	11-1422	11-1424	11-1426	11-1428	#11-1468			
GG1	006746	#9-1152								
GG10	007242	9-1213	#9-1217							
GG11	007300	9-1220	#9-1225							
GG12	007322	#9-1231	9-1234							
GG13	007332	9-1232	#9-1234							
GG14	007350	9-1237	#9-1241							
GG15	007406	9-1244	#9-1249							
GG16	007424	9-1245	#9-1254							
GG17	007472	#9-1266	9-1269							
GG18	007502	9-1267	#9-1269							
GG19	007536	9-1279	#9-1283							
GG2	007004	9-1155	#9-1160							
GG20	007574	9-1286	#9-1291							
GG21	007616	#9-1297	9-1300							
GG22	007626	9-1298	#9-1300							
GG23	007644	9-1303	#11-1309							
GG24	007702	11-1312	#11-1317							
GG25	007720	11-1313	#11-1322							
GG26	007766	#11-1334	11-1337							
GG27	007776	11-1335	#11-1337							
GG28	010032	11-1346	#11-1349							
GG3	007026	#9-1166	9-1169							
GG4	007036	9-1167	#9-1169							
GG5	007054	9-1172	#9-1176							
GG6	007112	9-1179	#9-1184							
GG7	007130	9-1180	#9-1189							
GG8	007176	#9-1201	9-1204							
GG9	007206	9-1202	#9-1204							
GNS	= ******	7-889	7-889	9-904	17-5643	17-5643	17-5659	17-5659	17-5659	17-5659
		17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659
		17-5659	17-5659	17-5659	17-5659	17-5659	17-5660	17-5660	17-5661	17-5661
GTSWR	= 104405	9-904	#17-5659							
HHDATO	006572	9-932	9-938	9-966	9-990	9-996	9-1020	9-1042	9-1055	9-1056
		9-1057	9-1058	9-1059	9-1060	9-1061	9-1062	9-1063	9-1064	9-1065
		9-1066	#9-1081							
HHDONE	006742	9-1054	9-1055	9-1056	9-1057	9-1058	9-1059	9-1060	9-1061	9-1062
		9-1063	9-1064	9-1065	9-1066	9-1080	#9-1133			

SYMBOL	VALUE	REFERENCES								
HHER0	005612	9-943	#9-1055							
HHER00	005726	9-949	#9-1057							
HHER1	005660	9-945	#9-1056							
HHER10	006454	9-1053	#9-1066							
HHER2	005774	9-972	#9-1058							
HHER3	006042	9-977	#9-1059							
HHER4	006110	9-1001	#9-1060							
HHER5	006156	9-1003	#9-1061							
HHER6	006224	9-1008	#9-1062							
HHER7	006272	9-1026	#9-1063							
HHER8	006340	9-1031	#9-1064							
HHER9	006406	9-1048	#9-1065							
HHHDON	030004	16-4703	#16-4840							
HHH1	025702	#15-4477								
HHH10	027002	#16-4636								
HHH11	027102	#16-4654								
HHH12	027202	#16-4672								
HHH13	027302	#16-4690								
HHH2	026002	#15-4493								
HHH3	026102	#15-4511								
HHH4	026202	#15-4528								
HHH5	026302	#15-4546								
HHH6	026402	#15-4565								
HHH7	026502	#15-4582								
HHH8	026602	#16-4600								
HHH9	026702	#16-4616								
HHP0	006602	9-927	9-1055	9-1056	9-1057	#9-1085				
HHP1	006612	9-929	9-1055	9-1056	9-1057	#9-1089				
HHP10	006722	9-1022	9-1063	9-1064	#9-1125					
HHP11	006732	9-1044	9-1065	9-1066	#9-1129					
HHP2	006622	9-934	9-1055	9-1056	9-1057	#9-1093				
HHP3	006632	9-939	#9-1097							
HHP4	006642	9-968	9-997	9-1058	9-1059	#9-1101				
HHP5	006652	9-961	9-1037	9-1039	9-1058	9-1059	9-1065	9-1065	9-1066	9-1066
		#9-1105								
HHP6	006662	9-963	9-1058	9-1059	#9-1109					
HHP7	006672	9-992	9-1060	9-1061	9-1062	#9-1113				
HHP8	006702	9-985	9-1015	9-1017	9-1060	9-1061	9-1062	9-1063	9-1063	9-1064
		9-1064	#9-1117							
HHP9	006712	9-987	9-1060	9-1061	9-1062	#9-1121				
HHTRAP	006522	9-960	#9-1067							
HH1	005022	#9-923								
HH10	005234	9-971	#9-973							
HH11	005252	9-976	#9-980							
HH12	005302	9-984	#9-988							
HH13	005324	#9-994	9-1004							
HH14	005344	#9-999	9-1002							
HH15	005354	9-1000	#9-1002							
HH16	005362	9-995	#9-1004							
HH17	005400	9-1007	#9-1010							
HH18	005436	9-1013	#9-1018							
HH19	005460	#9-1024	9-1027							

SYMBOL	VALUE	REFERENCES								
HH2	005052	9-926	#9-930							
HH20	005470	9-1025	#9-1027							
HH21	005506	9-1030	#9-1033							
HH22	005536	9-1036	#9-1040							
HH23	005560	#9-1046	9-1049							
HH24	005570	9-1047	#9-1049							
HH25	005606	9-1052	#9-1054							
HH3	005074	#9-936	9-946							
HH4	005114	#9-941	9-944							
HH5	005124	9-942	#9-944							
HH6	005132	9-937	#9-946							
HH7	005144	9-948	#9-955							
HH8	005202	9-959	#9-964							
HH9	005224	#9-970	9-973							
HT	= 000011	#7-877	17-5651	17-5651						
HXDATO	013270	11-1495	11-1502	11-1535	11-1544	11-1588	11-1607	11-1616	11-1642	11-1672
		11-1702	11-1708	11-1723	11-1729	11-1758	#11-1766			
HXDONE	013400	11-1686	11-1693	11-1698	11-1704	11-1710	11-1715	11-1720	11-1725	11-1731
		11-1742	11-1751	11-1760	#11-1806					
HXER1	012656	11-1492	#11-1690							
HXER10	013146	11-1575	#11-1744							
HXER11	013200	11-1620	#11-1753							
HXER12	013220	11-1622	#11-1756							
HXER13	013250	11-1649	11-1679	#11-1762						
HXER2	012716	11-1506	11-1594	#11-1700						
HXER22	012732	#11-1702	11-1755	11-1764						
HXER3	012746	11-1508	#11-1706							
HXER33	012762	#11-1708								
HXER4	012776	11-1655	11-1685	#11-1712						
HXER5	012676	11-1531	#11-1695							
HXER6	013032	11-1548	#11-1721							
HXER66	013046	#11-1723								
HXER7	013062	11-1550	#11-1727							
HXER8	013014	11-1513	11-1557	#11-1717						
HXER9	013112	11-1566	#11-1733							
HXP1	013300	11-1485	11-1519	11-1632	11-1637	11-1645	11-1667	11-1675	11-1762	11-1763
		#11-1771								
HXP2	013310	11-1487	11-1490	11-1501	11-1522	11-1529	11-1585	11-1690	11-1695	11-1700
		11-1706	11-1721	11-1727	#11-1776					
HXP3	013320	#11-1780								
HXP4	013330	11-1604	11-1753	11-1756	#11-1784					
HXP5	013340	11-1610	11-1754	11-1757	#11-1788					
HXP6	013350	11-1582	11-1602	11-1663	#11-1792					
HXP7	013360	11-1497	11-1543	11-1590	11-1615	11-1701	11-1707	#11-1797		
HXP8	013370	11-1538	11-1722	11-1728	#11-1801					
HX1	011664	#11-1482								
HX10	012072	11-1530	#11-1533							
HX11	012114	#11-1540	11-1552							
HX12	012134	#11-1546	11-1549							
HX13	012144	11-1547	#11-1549							
HX14	012152	11-1541	#11-1552							
HX15	012170	11-1556	#11-1561							

SYMBOL		VALUE	REFERENCES		
HX16		012216	11-1565	#11-1568	
HX165		012222	#11-1569	11-1735	11-1746
HX17		012250	11-1574	#11-1577	
HX18		012306	11-1581	#11-1586	
HX19		012326	#11-1592	11-1595	
HX2		011714	11-1488	#11-1489	
HX20		012336	11-1593	#11-1595	
HX21		012340	#11-1598		
HX22		012370	11-1601	#11-1605	
HX23		012410	#11-1612	11-1624	
HX24		012430	#11-1618	11-1621	
HX25		012440	11-1619	#11-1621	
HX26		012446	11-1613	#11-1624	
HX27		012450	#11-1628		
HX28		012502	11-1636	#11-1638	
HX29		012524	#11-1647	11-1650	
HX3		011730	11-1491	#11-1493	
HX30		012534	11-1648	#11-1650	
HX31		012552	11-1654	#11-1659	
HX32		012602	11-1666	#11-1668	
HX33		012624	#11-1677	11-1680	
HX34		012634	11-1678	#11-1680	
HX35		012652	11-1684	#11-1686	
HX4		011750	#11-1499	11-1509	
HX5		011770	#11-1504	11-1507	
HX6		012000	11-1505	#11-1507	
HX7		012006	11-1500	#11-1509	
HX8		012024	11-1512	#11-1515	
HX9		012056	11-1523	#11-1527	
IIIDON		021454	13-3174	#14-3344	
III1		020620	#13-3111		
III2		020664	#13-3127		
III3		020730	#13-3144		
III4		020774	#13-3160		
IOTVEC	=	000020	#7-877	9-903	9-903
JJJDON		022414	14-3428	#14-3595	
JJJ1		021460	#14-3361		
JJJ2		021544	#14-3379		
JJJ3		021630	#14-3396		
JJJ4		021714	#14-3414		
KKKDON		023256	14-3676	#14-3842	
KKK1		022420	#14-3615		
KKK3		022464	#14-3631		
KKK4		022530	#14-3647		
KKK5		022574	#14-3663		
LF	=	000012	#7-877	17-5651	17-5651
LLLDON		024220	14-3927	#15-4094	
LLL1		023262	#14-3858		
LLL2		023346	#14-3876		
LLL3		023432	#14-3894		
LLL4		023516	#14-3912		
LOOP		005020	#9-906	17-5643	

SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE	REFERENCES								
LPERR	= 104413	9-923	9-955	9-980	9-1010	9-1033	9-1152	9-1176	9-1217	9-1241
		9-1283	11-1309	11-1482	11-1515	11-1561	11-1577	11-1598	11-1628	11-1659
		11-1821	11-1832	11-1845	11-1858	11-1872	11-1885	11-1902	11-1919	11-1936
		11-1952	12-1970	12-1987	12-2004	12-2121	12-2146	12-2168	12-2274	12-2285
		12-2296	12-2308	12-2319	12-2330	12-2341	12-2353	12-2365	12-2376	12-2387
		12-2399	12-2410	12-2520	12-2531	12-2542	12-2553	12-2564	12-2577	12-2589
		13-2711	13-2722	13-2733	13-2744	13-2755	13-2767	13-2779	13-2790	13-2801
		13-2812	13-2823	13-2834	13-2845	13-2956	13-2968	13-2979	13-2990	13-3111
		13-3127	13-3144	13-3160	14-3361	14-3379	14-3396	14-3414	14-3615	14-3631
		14-3647	14-3663	14-3858	14-3876	14-3894	14-3912	15-4110	15-4126	15-4142
		15-4158	15-4174	15-4190	15-4207	15-4224	15-4240	15-4256	15-4272	15-4288
		15-4304	15-4320	15-4477	15-4493	15-4511	15-4528	15-4546	15-4565	15-4582
		16-4600	16-4616	16-4636	16-4654	16-4672	16-4690	16-4856	16-4873	16-4890
		16-4908	16-4926	16-5087	16-5106	16-5125	16-5144	17-5368	17-5381	17-5395
		17-5409	17-5425	17-5439	17-5455	17-5468	17-5483	17-5497	17-5511	#17-5661
MMMDON	030732	16-4940	#16-5071							
MMM1	030010	#16-4856								
MMM2	030062	#16-4873								
MMM3	030134	#16-4890								
MMM4	030206	#16-4908								
MMM5	030260	#16-4926								
MNUMBE	= 000266	#5-620	9-894							
MODDD0	031740	16-5214	16-5227	16-5234	#17-5295					
MODDD1	031750	16-5216	16-5228	16-5242	17-5266	#17-5297				
MODDOV	031352	16-5088	16-5107	16-5126	16-5145	#16-5195				
MODDSU	027406	15-4478	15-4494	15-4512	15-4529	15-4547	15-4566	15-4583	16-4601	16-4617
		16-4637	16-4655	16-4673	16-4691	#16-4738				
MODDT0	027764	16-4756	16-4769	16-4774	16-4797	#16-4836				
MODDT1	027774	16-4758	16-4770	16-4782	16-4814	#16-4838				
MODFD0	030712	16-4995	16-5008	16-5015	#16-5067					
MODFD1	030722	16-4997	16-5009	16-5021	#16-5069					
MODFOV	030336	16-4857	16-4874	16-4891	16-4909	16-4927	#16-4976			
MODFSU	025310	15-4111	15-4127	15-4143	15-4159	15-4175	15-4191	15-4208	15-4225	15-4241
		15-4257	15-4273	15-4289	15-4305	15-4321	#15-4369			
MODFT0	025646	15-4387	15-4400	15-4405	#15-4456					
MODFT1	025656	15-4389	15-4401	15-4411	#15-4458					
MODP1	025666	15-4374	#15-4460	16-4743	16-4981	16-5200				
MSA1	037024	#17-5904	19-7052							
MSA2	037042	#17-5905	19-7052							
MSA3	037065	#17-5906	19-7053							
MSA4	037077	#18-5908	19-7053							
MSA5	037115	#18-5909	19-7054							
MSA6	037130	#18-5910	19-7054							
MS1	037222	#18-5916	19-6854	19-6872	19-6903	19-6945	19-6978			
MS10	037421	#18-5923	19-6947	19-6980						
MS11	037443	#18-5924	19-6946	19-6979						
MS12	037467	#18-5925	19-6946	19-6979						
MS2	037240	#18-5917	19-6854	19-6872	19-6903	19-6945	19-6978			
MS3	037260	#18-5918	19-6855							
MS37	037512	#18-5926	19-7030							
MS4	037307	#18-5919	19-6855							
MS40	037530	#18-5927	19-7030							

SYMBOL	VALUE	REFERENCES								
MS41	037564	#18-5929	19-7001	19-7006	19-7031					
MS415	037544	#18-5928								
MS42	037610	#18-5930	19-7002	19-7006						
MS43	037626	#18-5931	19-7002	19-7007						
MS44	037643	#18-5932	19-7003	19-7007						
MS5	037335	#18-5920	19-6873	19-6904						
MS6	037357	#18-5921	19-6873	19-6904						
MS7	037375	#18-5922	19-6947	19-6980						
MULDSU	020354	13-2957	13-2969	13-2980	13-2991	#13-3029				
MULDT	020604	13-3048	13-3057	13-3063	13-3075	#13-3093				
MULFSU	017640	13-2712	13-2723	13-2734	13-2745	13-2756	13-2768	13-2780	13-2791	13-2802
		13-2813	13-2824	13-2835	13-2846	#13-2883				
MULFT	020054	13-2900	13-2909	#13-2940						
NNNDON	031760	16-5160	#17-5299							
NNN1	030736	#16-5087								
NNN2	031040	#16-5106								
NNN3	031142	#16-5125								
NNN4	031244	#16-5144								
OVDNTT	022404	14-3482	14-3498	14-3503	14-3529	14-3569	#14-3593			
OVDTT	024210	15-3980	15-4005	15-4010	15-4050	15-4083	#15-4092			
OVFNTT	021444	13-3230	13-3246	13-3251	14-3278	14-3318	#14-3342			
OVFTT	023246	14-3728	14-3753	14-3758	14-3798	14-3831	#14-3840			
OVUNDN	022004	14-3362	14-3380	14-3397	14-3415	#14-3468				
OVUNDT	023606	14-3859	14-3877	14-3895	14-3913	#15-3966				
OVUNFN	021044	13-3112	13-3128	13-3145	13-3161	#13-3216				
OVUNFT	022644	14-3616	14-3632	14-3648	14-3664	#14-3714				
PIRQ	= 177772	#7-877								
PIRQVE	= 000240	#7-877								
POWERM	037150	17-5664	#18-5913							
PROGNU	= 000002	#5-621	9-904							
PR0	= 000000	#7-877								
PR1	= 000040	#7-877								
PR2	= 000100	#7-877								
PR3	= 000140	#7-877								
PR4	= 000200	#7-877								
PR5	= 000240	#7-877								
PR6	= 000300	#7-877								
PR7	= 000340	#7-877								
PS	= 177776	#7-877	7-877							
PSW	= 177776	#7-877	17-5567							
PWRVEC	= 000024	#7-877	9-903	9-903	17-5664	17-5664	17-5664	17-5664	17-5664	17-5664
RDCHR	= 104407	#17-5659								
RESREG	= 104411	#17-5659								
RESVEC	= 000010	#7-877	9-903	9-903	9-903					
RSETUP	= 104412	9-1133	11-1472	11-1806	12-2106	12-2260	12-2507	13-2697	13-2942	13-3095
		14-3344	14-3595	14-3842	15-4094	15-4462	16-4840	16-5071	17-5299	17-5635
		#17-5660	17-5839	17-5850	17-5861					
R6	=%000006	#7-877	*9-903	*9-903	9-903					
R7	=%000007	#7-877								
SAVREG	= 104410	#17-5659								
SCOPE	= 000004	#7-877	9-919	9-1149	11-1480	11-1818	12-2118	12-2271	12-2517	13-2708
		13-2953	13-3108	14-3358	14-3612	14-3855	15-4107	15-4474	16-4853	16-5084

SYMBOL		VALUE	REFERENCES								
			17-5343	17-5643							
SPACE		037217	17-5741	17-5753	17-5757	17-5761	17-5799	#18-5915			
STACK	=	001100	#7-877	9-903	17-5894						
START		004336	7-889	#9-903	17-5664						
STKLMT	=	177774	#7-877								
SWR		001140	#8-890	9-903	*9-903	9-903	*9-903	*9-903	9-904	17-5347	17-5643
			17-5645	17-5645	17-5645	17-5645	17-5645	17-5647	17-5647	17-5647	17-5647
			17-5647	17-5657	17-5657	17-5664	17-5664	17-5883			
SWREG		000176	#7-889	9-903	9-904	17-5657	17-5657				
SW0	=	000001	#7-877								
SW00	=	000001	#7-877	7-877							
SW01	=	000002	#7-877	7-877							
SW02	=	000004	#7-877	7-877							
SW03	=	000010	#7-877	7-877							
SW04	=	000020	#7-877	7-877							
SW05	=	000040	#7-877	7-877							
SW06	=	000100	#7-877	7-877							
SW07	=	000200	#7-877	7-877							
SW08	=	000400	#7-877	7-877							
SW09	=	001000	#7-877	7-877							
SW1	=	000002	#7-877								
SW10	=	002000	#7-877								
SW11	=	004000	#7-877								
SW12	=	010000	#7-877								
SW13	=	020000	#7-877								
SW14	=	040000	#7-877								
SW15	=	100000	#7-877								
SW2	=	000004	#7-877								
SW3	=	000010	#7-877								
SW4	=	000020	#7-877								
SW5	=	000040	#7-877								
SW6	=	000100	#7-877								
SW7	=	000200	#7-877								
SW8	=	000400	#7-877								
SW9	=	001000	#7-877								
TAB	=	000011	#7-874	17-5904	17-5904	17-5905	17-5906	17-5906	18-5908	18-5909	18-5909
			18-5910	18-5910	18-5914	18-5916	18-5916	18-5917	18-5917	18-5918	18-5919
			18-5920	18-5921	18-5921	18-5922	18-5923	18-5924	18-5925	18-6455	18-6455
			18-6455	18-6456	18-6457	18-6457	18-6470	18-6470	18-6470	18-6470	18-6472
			18-6472	18-6472	18-6473	18-6474	18-6474	18-6474	18-6504	18-6504	18-6504
			19-6594	19-6594	19-6595	19-6595	19-6596	19-6596	19-6597	19-6597	19-6604
			19-6604	19-6605	19-6605	19-6611	19-6611	19-6612	19-6612	19-6613	19-6613
			19-6614	19-6614	19-6620	19-6620	19-6622	19-6622	19-6623	19-6623	19-6625
			19-6625	19-6626	19-6628	19-6628	19-6629	19-6630	19-6634	19-6634	19-6635
			19-6636	19-6643	19-6643	19-6643	19-6644	19-6644	19-6659	19-6659	
TBITVE	=	000014	#7-877	9-903	9-903						
TKVEC	=	000060	#7-877								
TPVEC	=	000064	#7-877								
TRAPVE	=	000034	#7-877	9-903	9-903						
TRTVEC	=	000014	#7-877								
TST1		005020	#9-919								
TST10		017056	#13-2708								

```
SYMBOL  VALUE              REFERENCES
TST11        020066       #13-2953
TST12        020616       #13-3108
TST13        021456       #14-3358
TST14        022416       #14-3612
TST15        023260       #14-3855
TST16        024222       #15-4107
TST17        025700       #15-4474
TST2         006744       #9-1149
TST20        030006       #16-4853
TST21        030734       #16-5084
TST22        031762       #17-5343
TST23        033144       #17-5639
TST3         011662       #11-1480
TST4         013402       #11-1818
TST5         014410       #12-2118
TST6         015112       #12-2271
TST7         016122       #12-2517
TYPE       = 104401        9-904        17-5365      17-5643      17-5643      17-5643      17-5647      17-5647      17-5651      17-5653
                           17-5657      17-5657      17-5657      17-5657      17-5657      17-5657     #17-5659      17-5664      17-5675
                           17-5703      17-5705      17-5710      17-5712      17-5740      17-5752      17-5756      17-5760      17-5779
                           17-5787      17-5798      17-5805      17-5815
TYPOC      = 104402        17-5657     #17-5659      17-5687      17-5730      17-5739      17-5743      17-5751      17-5755      17-5759
                           17-5763      17-5797
TYPON      = 104404       #17-5659
TYPOS      = 104403        17-5643      17-5643     #17-5659      17-5770
$APTHD       004322        9-900       #9-900
$ASTAT     = ******        17-5655      17-5655
$ATYC        035054        17-5655     #17-5655
$ATY1        035030       #17-5655
$ATY3        035036        17-5651     #17-5655
$ATY4        035046        17-5647     #17-5655
$AUTOB       001134       #8-890       *9-904        17-5657      17-5657      17-5657
$BASE        001372       #8-890
$BDADR       001122       #8-890
$BDDAT       001126       #8-890
$BELL        001306       #8-890        17-5647      17-5647      17-5647
$CDW1        001376       #8-890
$CDW2        001400       #8-890
$CHARC       034576       *17-5651     *17-5651      17-5651     *17-5651     #17-5651
$CKSWR       035276       #17-5657      17-5659      17-5659
$CLR.T       033346        17-5643     #17-5643
$CMTAG       001100       #8-890        9-903        9-903        9-903        9-903        9-903        9-903        9-903
$CM1       = 000024       #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
                          #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
                          #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
                          #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
                          #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
                          #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
                          #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890
$CM2       = 000050       #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
                          #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
                          #8-890        8-890        8-890       #8-890        8-890        8-890       #8-890        8-890        8-890
```

SYMBOL	VALUE	REFERENCES
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890
$CM3	= 000024	#8-890 8-890 8-890
$CM4	= 000024	#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
		#8-890 8-890 8-890 #8-890 8-890 8-890 #8-890 8-890 8-890
$CNTLG	035705	17-5657 #17-5657
$CNTLU	035700	17-5657 #17-5657
$CPUOP	001344	#8-890
$CRLF	001313	#8-890 17-5643 17-5647 17-5647 17-5647 17-5651 17-5651 17-5651 17-5657
		17-5676 17-5706 17-5713 17-5816 19-6854 19-6854 19-6855 19-6855 19-6872
		19-6872 19-6873 19-6873 19-6903 19-6903 19-6904 19-6904 19-6945 19-6945
		19-6946 19-6946 19-6947 19-6947 19-6978 19-6978 19-6979 19-6979 19-6980
		19-6980 19-7001 19-7001 19-7002 19-7002 19-7002 19-7002 19-7003 19-7003
		19-7006 19-7006 19-7006 19-7006 19-7007 19-7007 19-7007 19-7007 19-7029
		19-7030 19-7031 19-7052 19-7052 19-7053 19-7053 19-7054 19-7054
$DDW0	001402	#8-890
$DDW1	001404	#8-890
$DDW10	001426	#8-890
$DDW11	001430	#8-890
$DDW12	001432	#8-890
$DDW13	001434	#8-890
$DDW14	001436	#8-890
$DDW15	001440	#8-890
$DDW2	001406	#8-890
$DDW3	001410	#8-890
$DDW4	001412	#8-890
$DDW5	001414	#8-890
$DDW6	001416	#8-890
$DDW7	001420	#8-890
$DDW8	001422	#8-890
$DDW9	001424	#8-890
$DEVCT	001326	#8-890
$DEVM	001374	#8-890
$DOAGN	033366	17-5643 17-5643 17-5643 #17-5643
$ENDAD	033356	9-899 9-904 #17-5643 17-5647
$ENDCT	033200	9-903 #17-5643
$ENULL	033432	#17-5643
$ENV	001336	#8-890 9-904 17-5647 17-5651 17-5655 17-5655
$ENVM	001337	#8-890 9-903 17-5345 17-5651 17-5651 17-5655
$EOP	033144	#17-5643 17-5840 17-5851 17-5862
$EOPCT	033172	*9-903 #17-5643 17-5643
$ERFLG	001103	#8-890 17-5645 17-5645 17-5645 *17-5645 17-5645 17-5645 *17-5647 17-5647
		17-5647
$ERMAX	001115	#8-890 *9-903 17-5645 *17-5645 17-5645 17-5645
$ERROR	033716	9-903 #17-5647

SYMBOL	VALUE	REFERENCES							
$ERRPC	001116	#8-890	*17-5647	*17-5647	17-5647	17-5647	17-5647	17-5679	17-5686
$ERRTB	001442	#9-890	17-5699						
$ERTTL	001112	#8-890	17-5643	*17-5643	*17-5647	17-5647	17-5647		
$ESCAP	001304	#8-890	*9-903	*17-5645	17-5647	17-5647	17-5647	17-5647	
$ETABL	001336	#8-890							
$ETEND	001442	#8-890	9-900						
$FATAL	001320	#8-890	*17-5655						
$FFLG	035274	*17-5655	*17-5655	17-5655	*17-5655	#17-5655			
$FILLC	001156	#8-890	17-5651	17-5651	17-5651				
$FILLS	001155	#8-890	17-5651	17-5651					
$GDADR	001120	#8-890							
$GDDAT	001124	#8-890							
$GET42	033330	#17-5643							
$GTSWR	035346	#17-5657	17-5659	17-5659					
$HD =	000003	7-868	7-868	7-868					
$HIBTS	004322	#9-900							
$ICNT	001104	#8-890	*17-5645	17-5645	*17-5645	17-5645	17-5645		
$ILLUP	036176	17-5664	17-5664	#17-5664					
$INTAG	001135	#8-890	17-5657	17-5657	17-5657				
$ITEMB	001114	#8-890	*17-5647	17-5647	17-5647	17-5647	17-5682		
$LF	001314	#8-890	17-5647	17-5647	17-5651	17-5651			
$LFLG	035273	*17-5655	#17-5655						
$LOOP	033424	17-5643	#17-5643						
$LPADR	001106	#8-890	*9-903	*17-5645	*17-5645	17-5645	17-5645	17-5645	
$LPERR	001110	#8-890	*9-903	17-5645	*17-5645	17-5645	17-5645	17-5647	17-5871
$MADR1	001350	#8-890							
$MADR2	001354	#8-890							
$MADR3	001360	#8-890							
$MADR4	001364	#8-890							
$MAIL	001316	#8-890	9-900	9-900	9-903	9-904	17-5645	17-5647	17-5651
$MAMS1	001346	#8-890							
$MAMS2	001352	#8-890							
$MAMS3	001356	#8-890							
$MAMS4	001362	#8-890							
$MBADR	004324	#9-900							
$MFLG	035272	*17-5655	17-5655	*17-5655	#17-5655				
$MNEW	035723	17-5657	#17-5657						
$MSGAD	001332	#8-890	*17-5655	17-5655					
$MSGLG	001334	#8-890	*17-5655						
$MSGTY	001316	#8-890	17-5655	*17-5655	17-5655	*17-5655			
$MSWR	035712	17-5657	#17-5657						
$MTYP1	001347	#8-890							
$MTYP2	001353	#8-890							
$MTYP3	001357	#8-890							
$MTYP4	001363	#8-890							
$MXCNT	033714	17-5645	17-5645	17-5645	#17-5645				
$NULL	001154	#8-890	17-5651	17-5651	17-5651				
$NWTST =	000001	#9-919	9-919	#9-919	9-919	#9-1149	9-1149	#9-1149	9-1149	#11-1480
		11-1480	#11-1480	11-1480	#11-1818	11-1818	#11-1818	11-1818	#12-2118	12-2118
		#12-2118	12-2118	#12-2271	12-2271	#12-2271	12-2271	#12-2517	12-2517	#12-2517
		12-2517	#13-2708	13-2708	#13-2708	13-2708	#13-2953	13-2953	#13-2953	13-2953
		#13-3108	13-3108	#13-3108	13-3108	#14-3358	14-3358	#14-3358	14-3358	#14-3612

SYMBOL		VALUE	REFERENCES								
			14-3612	#14-3612	14-3612	#14-3855	14-3855	#14-3855	14-3855	#15-4107	15-4107
			#15-4107	15-4107	#15-4474	15-4474	#15-4474	15-4474	#16-4853	16-4853	#16-4853
			16-4853	#16-5084	16-5084	#16-5084	16-5084	#17-5343	17-5343	#17-5343	17-5343
$OCNT		035024	*17-5653	*17-5653	#17-5653						
$OMODE		035026	*17-5653	*17-5653	17-5653	*17-5653	*17-5653	#17-5653			
$CVER		033700	17-5645	17-5645	17-5645	17-5645	#17-5645				
$PASS		001324	#8-890	*9-903	*17-5643	*17-5643	17-5643	17-5643	17-5643	17-5645	17-5645
			17-5645								
$PASTM		004330	#9-900								
$PWRAD		036160	#17-5664								
$PWRDN		036020	9-903	#17-5664	17-5664						
$PWRMG		036154	#17-5664								
$PWRUP		036072	17-5664	#17-5664							
$QUES		001312	#8-890	17-5647	17-5647	17-5651	17-5651	17-5657			
$RDCHR		035560	#17-5657	17-5659	17-5659						
$RDDEC	=	******	17-5659								
$RDLIN	=	******	17-5659								
$RDOCT	=	******	17-5659								
$RDSZ	=	000001	#17-5657	17-5657							
$REGAD		001160	#8-890								
$REG0		001162	#8-890	17-5647	17-5647	17-5647					
$REG1		001164	#8-890								
$REG10		001202	#8-890								
$REG11		001204	#8-890								
$REG12		001206	#8-890								
$REG13		001210	#8-890								
$REG14		001212	#8-890								
$REG15		001214	#8-890								
$REG16		001216	#8-890								
$REG17		001220	#8-890								
$REG2		001166	#8-890								
$REG20		001222	#8-890								
$REG21		001224	#8-890								
$REG22		001226	#8-890								
$REG23		001230	#8-890								
$REG3		001170	#8-890								
$REG4		001172	#8-890								
$REG5		001174	#8-890								
$REG6		001176	#8-890								
$REG7		001200	#8-890								
$RESRE		034212	#17-5649	17-5659							
$RTNAD		033426	#17-5643								
$RTRN		033422	9-903	*9-903	*9-903	17-5643	#17-5643				
$R2A	=	******	17-5659								
$SAVRE		034154	#17-5649	17-5659	17-5659						
$SAVR6		036202	*17-5664	17-5664	*17-5664	*17-5664	#17-5664				
$SCOPE		033436	9-903	#17-5645							
$SETUP	=	000137	#7-887	7-887	#7-887	7-887	#7-887	7-887	#7-887	7-887	#7-887
			7-887	#7-887	7-887	#7-887	9-903	9-903	9-903	9-903	9-903
			9-903	9-903	9-903	9-903	9-903	9-903	9-903	9-904	9-904
			9-904	17-5643	17-5643	17-5645	17-5647	17-5647	17-5647	17-5647	17-5657
			17-5657	17-5664							

SYMBOL	VALUE	REFERENCES								
$STUP	= 177777	#7-887	#7-887	7-887	#7-887	#7-887	7-887	#7-887	#7-887	7-887
		#7-887	#7-887	7-887	#7-887	#7-887	7-887	#7-887	#7-887	7-887
$SVLAD	033644	17-5645	#17-5645							
$SVPC	= 004322	#9-899	9-899							
$SWR	= 177400	7-868	#7-868	#7-872	8-890	8-890	8-890	9-903	9-903	9-903
		9-903	9-903	9-919	9-1149	11-1480	11-1818	12-2118	12-2271	12-2517
		13-2708	13-2953	13-3108	14-3358	14-3612	14-3855	15-4107	15-4474	16-4853
		16-5084	17-5343	17-5643	17-5643	17-5643	17-5643	17-5643	17-5645	17-5645
		17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645
		17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645
		17-5647	17-5647	17-5647	17-5647	17-5647	17-5647	17-5647	17-5647	17-5647
		17-5647	17-5647	17-5664						
$SWREG	001340	#8-890	9-903							
$SWRMK	= 000000	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645	17-5645
		17-5645								
$SWRMS	= 000200	#7-873								
$TAB	037215	17-5806	#18-5914	19-6853	19-6853	19-6853	19-6855	19-6868	19-6868	19-6869
		19-6869	19-6869	19-6870	19-6870	19-6870	19-6871	19-6871	19-6872	19-6872
		19-6872	19-6902	19-6902	19-6902	19-6944	19-6944	19-6944	19-6977	19-6977
		19-6999	19-6999	19-7000	19-7001	19-7005	19-7005	19-7005	19-7012	19-7012
		19-7029	19-7033	19-7033	19-7035	19-7045	19-7045	19-7046	19-7049	19-7051
		19-7051	19-7051							
$TBIT	033430	*9-903	*17-5643	17-5643	17-5643	#17-5643	*17-5664			
$TERM	= 000030	#17-5662								
$TESTN	001322	#8-890	*17-5645							
$TIMES	001302	#8-890	*9-903	*17-5643	*17-5645	17-5645	*17-5645	17-5645	17-5645	
$TKB	001146	#8-890	17-5651	17-5651	17-5657	17-5657	17-5657	17-5657	17-5657	17-5657
$TKS	001144	#8-890	17-5651	17-5651	17-5657	17-5657	17-5657	17-5657	17-5657	17-5657
		17-5657								
$TMP0	001232	#8-890	17-5677	17-5678	19-6853	19-6855	19-6868	19-6869	19-6870	19-6871
		19-6872	19-6902	19-6944	19-6977	19-6999	19-7001	19-7005	19-7012	19-7029
		19-7033	19-7035	19-7045	19-7046	19-7049	19-7051			
$TMP1	001234	#8-890	17-5679	19-6853	19-6855	19-6868	19-6869	19-6870	19-6871	19-6872
		19-6902	19-6944	19-6977	19-6999	19-7001	19-7005	19-7012	19-7029	19-7033
		19-7035	19-7045	19-7046	19-7049	19-7051				
$TMP10	001252	#8-890	9-1055	9-1056	9-1057	9-1058	9-1059	9-1060	9-1061	9-1062
		9-1063	9-1064	9-1065	9-1066	11-1365	11-1367	11-1369	11-1373	11-1375
		11-1391	11-1393	11-1395	11-1397	11-1399	11-1401	11-1418	11-1420	11-1422
		11-1424	11-1426	11-1428	12-2251	12-2476	13-2661	13-2911	13-3059	13-3229
		14-3481	14-3727	15-3979	15-4401	16-4770	16-5009	16-5228	17-5594	19-6872
		19-6902	19-6947	19-6980	19-7005	19-7051				
$TMP11	001254	#8-890	9-1055	9-1056	9-1057	9-1058	9-1059	9-1060	9-1061	9-1062
		9-1063	9-1064	9-1065	9-1066	11-1365	11-1367	11-1369	11-1373	11-1375
		11-1391	11-1393	11-1395	11-1397	11-1399	11-1401	11-1418	11-1420	11-1422
		11-1424	11-1426	11-1428	13-3249	14-3320	14-3501	14-3571	14-3756	14-3834
		15-4008	15-4086	15-4402	15-4451	16-4772	16-4831	16-5010	16-5055	16-5229
		17-5283	17-5589	19-6902	19-6944	19-6977	19-7005	19-7052		
$TMP12	001256	#8-890	15-4403	15-4452	16-4771	16-4832	16-5011	16-5056	16-5230	17-5284
		17-5595	19-6944	19-6977	19-7052					
$TMP13	001260	#8-890	16-5012	16-5061	16-5231	17-5289	19-6977			
$TMP14	001262	#8-890	16-5013	16-5062	16-5232	17-5290				
$TMP15	001264	#8-890								

SYMBOL	VALUE	REFERENCES
$TMP16	001266	#8-890
$TMP17	001270	#8-890
$TMP2	001236	#8-890 9-926 9-959 9-984 9-1013 9-1036 9-1067 9-1072 9-1155 9-1179 9-1189 9-1194 9-1220 9-1244 9-1254 9-1259 9-1286 11-1312 11-1322 11-1327 11-1351 11-1377 11-1403 11-1488 11-1523 11-1565 11-1581 11-1601 11-1636 11-1666 11-1739 12-2051 12-2127 12-2150 12-2172 12-2207 12-2210 12-2455 13-2640 13-2890 13-3038 13-3234 14-3288 14-3313 14-3486 14-3539 14-3564 14-3732 14-3808 15-3984 15-4060 15-4378 16-4747 16-4985 16-5204 17-5565 17-5612 17-5832 17-5847 17-5858 19-6853 19-6855 19-6868 19-6869 19-6870 19-6871 19-6872 19-6902 19-6944 19-6977 19-6999 19-7001 19-7005 19-7012 19-7029 19-7033 19-7035 19-7045 19-7046 19-7049 19-7051
$TMP20	001272	#8-890
$TMP21	001274	#8-890
$TMP22	001276	#8-890
$TMP23	001300	#8-890
$TMP3	001240	#8-890 9-1055 9-1056 9-1057 9-1058 9-1059 9-1060 9-1061 9-1062 9-1063 9-1064 9-1065 9-1066 9-1076 11-1363 11-1365 11-1367 11-1369 11-1373 11-1375 11-1389 11-1391 11-1393 11-1395 11-1397 11-1399 11-1401 11-1416 11-1418 11-1420 11-1422 11-1424 11-1426 11-1428 11-1691 11-1696 11-1712 11-1717 11-1749 12-2065 12-2097 12-2217 12-2236 12-2247 12-2469 13-2654 13-2904 13-3052 13-3224 14-3476 14-3722 15-3974 15-4393 16-4762 16-5001 16-5220 17-5586 17-5835 19-6854 19-6855 19-6870 19-6872 19-6903 19-6945 19-6978 19-6999 19-7001 19-7006 19-7012 19-7033 19-7045 19-7054
$TMP4	001242	#8-890 9-1055 9-1056 9-1057 9-1058 9-1059 9-1060 9-1061 9-1062 9-1063 9-1064 9-1065 9-1066 9-1078 11-1363 11-1365 11-1367 11-1369 11-1373 11-1375 11-1389 11-1391 11-1393 11-1395 11-1397 11-1399 11-1401 11-1416 11-1418 11-1420 11-1422 11-1424 11-1426 11-1428 11-1690 11-1695 11-1713 11-1718 11-1747 12-2066 12-2067 12-2098 12-2218 12-2227 12-2235 12-2246 12-2471 13-2656 13-2906 13-3054 13-3226 14-3478 14-3724 15-3976 15-4395 16-4764 16-5003 16-5222 17-5591 17-5837 19-6854 19-6855 19-6868 19-6869 19-6870 19-6872 19-6903 19-6945 19-6978 19-7000 19-7002 19-7006 19-7012 19-7033 19-7054
$TMP5	001244	#8-890 9-1055 9-1056 9-1057 9-1058 9-1059 9-1060 9-1061 9-1062 9-1063 9-1064 9-1065 9-1066 11-1363 11-1365 11-1367 11-1369 11-1373 11-1375 11-1389 11-1391 11-1393 11-1395 11-1397 11-1399 11-1401 11-1416 11-1418 11-1420 11-1422 11-1424 11-1426 11-1428 11-1700 11-1706 11-1721 11-1727 11-1753 11-1756 11-1762 12-2068 12-2219 12-2228 12-2248 12-2473 13-2658 13-2908 13-3056 13-3228 14-3480 14-3726 15-3978 15-4397 16-4766 16-5005 16-5224 17-5587 19-6853 19-6868 19-6869 19-6871 19-6873 19-6904 19-6946 19-6979 19-7002 19-7007 19-7030 19-7053
$TMP6	001246	#8-890 9-1055 9-1056 9-1057 9-1058 9-1059 9-1060 9-1061 9-1062 9-1063 9-1064 9-1065 9-1066 11-1363 11-1365 11-1367 11-1369 11-1373 11-1375 11-1389 11-1391 11-1393 11-1395 11-1397 11-1399 11-1401 11-1416 11-1418 11-1420 11-1422 11-1424 11-1426 11-1428 11-1702 11-1708 11-1723 11-1729 11-1758 12-2069 12-2221 12-2229 12-2237 12-2249 12-2474 13-2659 13-2909 13-3057 13-3230 14-3482 14-3728 15-3980 15-4399 16-4768 16-5007 16-5226 17-5593 19-6853 19-6873 19-6904 19-6946 19-6979 19-7003 19-7007 19-7030 19-7053
$TMP7	001250	#8-890 11-1701 11-1707 11-1722 11-1728 11-1754 11-1757 11-1763 12-2220 12-2230 12-2238 12-2250 12-2475 13-2660 13-2910 13-3058 13-3248 14-3500 14-3755 14-3833 15-4007 15-4085 15-4400 16-4769 16-5008 16-5227 17-5588 19-6872 19-6902 19-6947 19-6980 19-7031 19-7051

SYMBOL	VALUE	REFERENCES								
$TN	= 000023	7-868	#7-868	9-919	9-919	#9-919	9-1149	9-1149	#9-1149	11-1480
		11-1480	#11-1480	11-1818	11-1818	#11-1818	12-2118	12-2118	#12-2118	12-2271
		12-2271	#12-2271	12-2517	12-2517	#12-2517	13-2708	13-2708	#13-2708	13-2953
		13-2953	#13-2953	13-3108	13-3108	#13-3108	14-3358	14-3358	#14-3358	14-3612
		14-3612	#14-3612	14-3855	14-3855	#14-3855	15-4107	15-4107	#15-4107	15-4474
		15-4474	#15-4474	16-4853	16-4853	#16-4853	16-5084	16-5084	#16-5084	17-5343
		17-5343	#17-5343	17-5639						
$TPB	001152	#8-890	17-5651	17-5651	17-5651					
$TPFLG	001157	#8-890	17-5651	17-5651	17-5651					
$TPS	001150	#8-890	17-5651	17-5651	17-5651					
$TRAP	035734	9-903	#17-5659							
$TRAP2	035756	#17-5659	17-5659							
$TRP	= 000014	#17-5659	17-5659	17-5659	17-5659	17-5659	#17-5659	17-5659	17-5659	17-5659
		17-5659	#17-5659	17-5659	17-5659	17-5659	17-5659	#17-5659	17-5659	17-5659
		17-5659	17-5659	#17-5659	17-5659	17-5659	17-5659	17-5659	#17-5659	17-5659
		17-5659	17-5659	17-5659	#17-5659	17-5659	17-5659	17-5659	17-5659	#17-5659
		17-5659	17-5659	17-5659	17-5659	#17-5659	17-5659	17-5659	17-5659	17-5659
		#17-5659	17-5660	17-5660	17-5660	17-5660	#17-5660	17-5661	17-5661	17-5661
		17-5661	#17-5661							
$TRPAD	035770	17-5659	#17-5659	17-5662						
$TSTM	004326	#9-900								
$TSTNM	001102	#8-890	*17-5643	17-5645	17-5645	*17-5645	17-5645	17-5645	17-5645	17-5645
		17-5647	17-5647	17-5647	17-5677					
$TYPBN	= ******	17-5659								
$TYPDS	= ******	17-5659								
$TYPE	034250	#17-5651	17-5655	17-5659	17-5659					
$TYPEC	034462	17-5651	17-5651	17-5651	#17-5651	17-5651	17-5657			
$TYPEX	034600	17-5651	17-5651	#17-5651						
$TYPOC	034626	#17-5653	17-5659	17-5659						
$TYPON	034642	17-5653	#17-5653	17-5659						
$TYPOS	034602	#17-5653	17-5659							
$UNIT	001330	#8-890								
$UNITM	004332	#9-900								
$USWR	001342	#8-890								
$VECT1	001366	#8-890								
$VECT2	001370	#8-890								
$XTSTR	033450	#17-5645								
$$GET4	= 000001	#17-5643	#17-5643	17-5643						
$OFILL	035025	*17-5653	*17-5653	17-5653	#17-5653					
$40CAT	= ******	17-5645	17-5647							
.LPER	036746	17-5661	#17-5871							
.RSET	036754	17-5660	#17-5883							
.$ASTA	= ******	17-5655	17-5655							
.$X	= 004322	#9-900	9-900							

```
MACRO NAME      REFERENCES
ACCMAC          #7-780
AMAC1           #18-6246
ASMAC1          #7-737       #9-1055      #9-1056      #9-1057      #9-1058      #9-1059      #9-1060      #9-1061      #9-1062      #9-1063
                #9-1064      #9-1065      #9-1066      #11-1363     #11-1365     #11-1367     #11-1369     #11-1373     #11-1375     #11-1389
                #11-1391     #11-1393     #11-1395     #11-1397     #11-1399     #11-1401     #11-1416     #11-1418     #11-1420     #11-1422
                #11-1424     #11-1426     #11-1428
ASMAC2          #7-745        9-1055       9-1056       9-1057       9-1058       9-1059       9-1060       9-1061       9-1062       9-1063
                 9-1064       9-1065       9-1066       11-1365      11-1367      11-1369      11-1373      11-1375      11-1391      11-1393
                 11-1395      11-1397      11-1399      11-1401      11-1418      11-1420      11-1422      11-1424      11-1426      11-1428
AS1             #18-6299
AS10            #18-6328
AS11            #18-6334
AS2             #18-6302
AS3             #18-6308
AS4             #18-6313
AS5             #18-6318      18-6338
AS6             #18-6321      18-6339
AS7             #18-6324
COMMEN          #7-877
CORMAC          #18-6435     #18-6439     #18-6440     #18-6441     #18-6442     #18-6443     #18-6444     #18-6445     #18-6446     #18-6447
                #18-6448     #18-6449
DM1             #18-5972     #18-5975     #18-5977     #18-5979     #18-5982     #18-5984
DM2             #18-5988      18-5991      18-5993
ENDCOM          #7-877
ENDPAS          #5-626        17-5643
ERMAC           #5-658        17-5647
ESCAPE          #7-877
FPM1            #18-5939     #18-5942     #18-5944     #18-5946     #18-5948     #18-5950     #18-5952     #18-5954     #18-5956     #18-5958
                #18-5960     #18-5962     #18-5964
GETPRI          #7-877       #17-5643
GETSWR          #7-877       #9-904       #9-904
HMAC1           #7-750
HMAC2           #7-760
HMAC3           #7-766
ITEMAC          #7-730       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
                #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896       #9-896
```

MACRO NAME	REFERENCES									
	#9-896	#9-896	#9-896							
LCM1	#18-6389	18-6416	18-6419							
LCM2	#18-6394	#18-6415	#18-6424							
LCM3	#18-6398	#18-6421	#18-6423							
LCM4	#18-6405	18-6420	18-6422							
LCM5	#18-6408	18-6417								
LDM1	#18-6263									
LDM2	#18-6269									
LDM3	#18-6273									
LDM4	#18-6277									
LDM5	#18-6281									
LDM6	#18-6284	#18-6411	#18-6418							
LDM7	#18-6288									
LDM8	#18-6292									
LDM9	#18-6296									
LD1M1	#18-6250									
LD1M2	#18-6254									
LD1M3	#18-6260									
LOADTP	#5-634									
LPER	#7-853	#9-923	#9-955	#9-980	#9-1010	#9-1033	#9-1152	#9-1176	#9-1217	#9-1241
	#9-1283	#11-1309	#11-1482	#11-1515	#11-1561	#11-1577	#11-1598	#11-1628	#11-1659	#11-1821
	#11-1832	#11-1845	#11-1858	#11-1872	#11-1885	#11-1902	#11-1919	#11-1936	#11-1952	#12-1970
	#12-1987	#12-2004	#12-2121	#12-2146	#12-2168	#12-2274	#12-2285	#12-2296	#12-2308	#12-2319
	#12-2330	#12-2341	#12-2353	#12-2365	#12-2376	#12-2387	#12-2399	#12-2410	#12-2520	#12-2531
	#12-2542	#12-2553	#12-2564	#12-2577	#12-2589	#13-2711	#13-2722	#13-2733	#13-2744	#13-2755
	#13-2767	#13-2779	#13-2790	#13-2801	#13-2812	#13-2823	#13-2834	#13-2845	#13-2956	#13-2968
	#13-2979	#13-2990	#13-3111	#13-3127	#13-3144	#13-3160	#14-3361	#14-3379	#14-3396	#14-3414
	#14-3615	#14-3631	#14-3647	#14-3663	#14-3858	#14-3876	#14-3894	#14-3912	#15-4110	#15-4126
	#15-4142	#15-4158	#15-4174	#15-4190	#15-4207	#15-4224	#15-4240	#15-4256	#15-4272	#15-4288
	#15-4304	#15-4320	#15-4477	#15-4493	#15-4511	#15-4528	#15-4546	#15-4565	#15-4582	#16-4600
	#16-4616	#16-4636	#16-4654	#16-4672	#16-4690	#16-4856	#16-4873	#16-4890	#16-4908	#16-4926
	#16-5087	#16-5106	#16-5125	#16-5144	#17-5368	#17-5381	#17-5395	#17-5409	#17-5425	#17-5439
	#17-5455	#17-5468	#17-5483	#17-5497	#17-5511					
MDO1	#18-6056	#18-6069	#18-6074							
MDO2	#18-6106	#18-6119	#18-6124							
MDU1	#18-6059	18-6062	18-6067							
MDU2	#18-6109	#18-6112	#18-6117							
MERROR	#7-756									
MFM4	#18-6196	#18-6208	#18-6226							
MFM5	#18-6199	#18-6210	#18-6216	#18-6220	#18-6222	#18-6224	#18-6228	#18-6234	#18-6236	#18-6240
	#18-6242									
MFM6	#18-6202	#18-6212	#18-6230							
MFM7	#18-6205	18-6214	18-6218	18-6232	18-6238					
MFO1	#18-6034	#18-6044	#18-6046							
MFO2	#18-6084	18-6094	18-6096							
MFU1	#18-6037	#18-6040	#18-6042							
MFU2	#18-6087	18-6090	18-6092							
MM1	#18-5996	#18-6000	#18-6002	#18-6005	#18-6008	#18-6010	#18-6028	#18-6030	#18-6080	#18-6082
MM2	#18-6013	#18-6017	#18-6020	#18-6022	#18-6024	#18-6050	#18-6052	#18-6064	#18-6071	#18-6102
	#18-6104	#18-6114	#18-6121							
MODDM1	#18-6161	18-6169	18-6173	18-6177	18-6180	18-6183	18-6194			
MODDM2	#18-6164	18-6167	18-6171	18-6175	18-6185	18-6188	18-6190	18-6192		

MACRO NAME	REFERENCES									
MODM1	#18-6129	18-6135	18-6141	18-6144	18-6147	18-6149	18-6151	18-6153		
MODM2	#18-6132	18-6137	18-6139	18-6155						
MSG	#9-909	#9-919	#9-1135	#9-1149	#11-1474	#11-1480	#11-1810	#11-1818	#12-2110	#12-2118
	#12-2263	#12-2271	#12-2510	#12-2517	#13-2700	#13-2708	#13-2945	#13-2953	#13-3098	#13-3108
	#14-3348	#14-3358	#14-3599	#14-3612	#14-3846	#14-3855	#15-4099	#15-4107	#15-4466	#15-4474
	#16-4844	#16-4853	#16-5075	#16-5084	#17-5301	#17-5343				
MULT	#7-877									
NAMEP	#5-648	9-904								
NEWTST	#7-877	9-919	9-1149	11-1480	11-1818	12-2118	12-2271	12-2517	13-2708	13-2953
	13-3108	14-3358	14-3612	14-3855	15-4107	15-4474	16-4853	16-5084	17-5343	
NTMPM	#5-644									
OU1	#18-6349	18-6375	18-6377	18-6379						
OU2	#18-6355	#18-6376	#18-6378	#18-6380						
OU3	#18-6362	18-6381	18-6382	18-6383	18-6384					
POP	#7-877	17-5649	17-5655	17-5655	17-5664	17-5664				
PUSH	#7-877	17-5649	17-5655	17-5655	17-5655	17-5664	17-5664			
REPORT	#7-877									
ROM	#7-679									
ROMAC	#7-811									
ROM1	#7-682									
ROM10	#7-703									
ROM11	#7-706									
ROM12	#7-709									
ROM13	#7-712									
ROM14	#7-715									
ROM15	#7-718									
ROM16	#7-721									
ROM17	#7-724									
ROM2	#7-685									
ROM20	#7-727									
ROM3	#7-688									
ROM30	#7-673									
ROM31	#7-676									
ROM4	#7-691									
ROM5	#7-694									
ROM6	#7-697									
ROM7	#7-700									
RSET	#5-651	#9-1133	#11-1472	#11-1806	#12-2106	#12-2260	#12-2507	#13-2697	#13-2942	#13-3095
	#14-3344	#14-3595	#14-3842	#15-4094	#15-4462	#16-4840	#16-5071	#17-5299	#17-5635	#17-5839
	#17-5850	#17-5861								
RTMAC	#18-6343	#18-6367	#18-6368	#18-6369	#18-6370					
RTMPM	#5-639									
SETPRI	#7-877									
SETTRA	#17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659
	17-5660	17-5661								
SETUP	#7-877	9-903								
SKIP	#7-877									
SLASH	#7-877									
SPACE	#7-860	#7-877								
SSKAD	#7-850									
STARS	#7-877	8-890	8-890	8-890	9-899	9-900	9-900	9-900	9-919	9-919
	9-1149	9-1149	11-1480	11-1480	11-1818	11-1818	12-2118	12-2118	12-2271	12-2271

CKFPBAO CREATED BY MACRO ON 18-SEP-79 AT 13:50 PAGE 40
MACRO CROSS REFERENCE CREF SEQ 0211

MACRO NAME	REFERENCES									
	12-2517	12-2517	13-2708	13-2708	13-2953	13-2953	13-3108	13-3108	14-3358	14-3358
	14-3612	14-3612	14-3855	14-3855	15-4107	15-4107	15-4474	15-4474	16-4853	16-4853
	16-5084	16-5084	17-5343	17-5343	17-5643	17-5645	17-5647	17-5649	17-5651	17-5653
	17-5655	17-5657	17-5657	17-5657	17-5659	17-5664	17-5664	17-5668	17-5827	17-5844
	17-5855	17-5869	17-5875							
SUMMAC	#7-838									
SWRSU	#7-877	#9-903	#9-903							
TMAC	#7-670	17-5639								
TRMTRP	#17-5659	17-5662								
TYPBIN	#7-877									
TYPDEC	#7-877									
TYPNAM	#7-877	#9-904								
TYPNUM	#7-877									
TYPOCS	#7-877	17-5643	17-5643							
TYPOCT	#7-877	#17-5657								
TYPTXT	#7-877	17-5643	17-5643							
$$CMRE	#7-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890
	8-890	8-890	8-890	8-890	8-890	8-890	8-390	8-890	8-890	8-890
	8-890									
$$CMTM	#7-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890
	8-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890	8-890
	8-890									
$$ESCA	#7-877									
$$NEWT	#7-877	#9-919	#9-1149	#11-1480	#11-1818	#12-2118	#12-2271	#12-2517	#13-2708	#13-2953
	#13-3108	#14-3358	#14-3612	#14-3855	#15-4107	#15-4474	#16-4853	#16-5084	#17-5343	
$$SET	#17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659	17-5659
	17-5660	17-5661								
$$SETM	#9-903	#9-903								
$$SETU	#9-903	9-903								
$$SKIP	#7-877									
.EQUAT	#7-863	7-877								
.EQUIV	#7-857	#7-867	#7-877	#7-877	#7-877	#7-877	#7-877	#7-877	#7-877	#7-877
	#7-877	#7-877	#7-877	#7-877	#7-877	#7-877	#7-877	#7-877	#7-877	#7-877
	#7-877	#7-877	#7-877	#7-877	#7-877					
.HEADE	#7-863	7-868								
.SETUP	#7-863	7-887								
.SWRHI	#7-863									
.$ACT1	#7-863	#9-899								
.$APTB	#7-865	#8-890	8-890							
.$APTH	#7-865	9-900								
.$APTY	#7-866	17-5655								
.$CATC	#7-863	7-889								
.$CMTA	#7-863	#7-890								
.$EOP	#7-864	17-5643								
.$ERRO	#7-864	#17-5647								
.$POWE	#7-865	#17-5664								
.$READ	#7-866	17-5657								
.$SAVE	#7-864	17-5649								
.$SCOP	#7-864	#17-5645								
.$TRAP	#7-865	#17-5659								
.$TYPD	#7-865									
.$TYPE	#7-864	#17-5651								

MACRO CROSS REFERENCE

MACRO NAME	REFERENCES
.$TYPO	#7-864 #17-5653