IDENTIFICATION

PRODUCT CODE: AC-F241B-MC

PRODUCT NAME: CJKDCB0 KEF11-AA DIAG #1

DATE CREATED: NOV-79

MAINTAINER: DIAGNOSTIC ENGINEERING

PROGRAM HISTORY

DATE	REVISION	REASON FOR REVISION
JUNE-79	A	FIRST RELEASE
NOVEMBER-79	B	CORRECTIONS WERE MADE TO MULTI-TESTER SUPPORT CODE.

CONTENTS

ABSTRACT

THE TWO PROGRAMS:

CJKDCB, CJKDDB

ARE DESIGN TO DETECT AND REPORT LOGIC FAULTS IN THE F-11 MMU AND FLOATING POINT CHIP SET. THE DESIGN IS AN ATTEMPT TO REACH ALL MIRCO-CODE LOCATIONS. TESTS ARE PARTITIONED INTO TWO STAND-ALONE PROGRAMS DESCRIBED BELOW.

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

A. CJKDCB

CJKDCB TESTS: (FLOATING POINT TEST 1)
- 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)
- ADDF, ADDD AND SUBD (ALL CONDITIONS NOT TESTED IN DFFPA)
- CMPD AND CMPF
- DIVD AND DIVF
- MULD AND MULF
- MODD AND MODF

B. CJKDDA

CJKDDA TESTS: (FLOATING POINT TEST 2)
- 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 PROCESSOR WITH THE DCF11-AA,KTF11-AA AND KEF11-A CHIP SET.

2.2 STORAGE

BOTH PROGRAMS REQUIRE A MEMORY SYSTEM OF AT LEAST 16K TO LOAD AND RUN.

2.3 PRELIMINARY PROGRAMS

THESE TWO DIAGNOSTICS WILL ASSUME THAT THE BASIC CENTRAL PROCESSOR IS FAULTLESS, THEREFORE WHEN IN DOUBT RUN THE DCF11-AA PROCESSOR DIAGNOSTICS BEFORE THESE FLOATING POINT DIAGNOSTICS.

3. LOADING PROCEDURE

THE PROGRAMS WILL BE SUPPLIED ON THE 11/23 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 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>

6. ERRORS

6.1 SUMMARIES

WHEN AN ERROR IS ENCOUNTERED, AN ERROR MESSAGE ACCOMPANIED BY THE ERROR PC ARE TYPED.
THERE ARE FOUR STANDARD ERROR MESSAGES USED, DESCRIBING THE PROBABLE CAUSE OF FAILURE, SUCH AS: PROBABLY BAD MMU CHIP; BAD FP1 CHIP; BAD HYBRID FP CHIP; FLOATING POINT ERROR.

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 2 SECONDS FOR EACH PROGRAM ON ANY PASS.

8.2 STACK POINTER

THE STACK POINTER IS INITIALIZED TO 1100 IN EACH OF THE TWO 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.

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 TWO 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 ACT, APT AND XXDP COMPATIBILTY

THESE PROGRAMS ARE FULLY COMPATIBLE WITH:
- APT
- ACT
- XXDP MONITOR AND CHAIN PROGRAMS.

9. PROGRAM DESCRIPTION

TEST 1 LDFPS, STFPS AND DATA PATHS TEST

THIS IS A TEST OF THE LDFPS (LOAD FLOATING POINT STATUS) AND STFPS (STORE FLOATING POINT STATUS) INSTRUCTIONS. VARIOUS PATTERN ARE USED AND RUN THROUGH THE FLOATING POINT STATUS REGISTER. ONLY DMO AND SMO ARE USED. NOTE THAT A MASK MUST BE USED BECAUSE SOME OF THE FPS BITS CANNOT BE SET.

TEST 2 CFCC TEST

THIS IS A TEST OF THE COPY CONDITION CODES INSTRUCTION, CFCC.

TEST 3 SETF, SETD, SETI AND SETL TEST

THIS IS A TEST OF THE SETF, SETD, SETI AND SETL INSTRUCTIONS. EACH INSTRUCTION IS EXECUTED WITH THE FPS CONTAINING ALL ONES AND ALSO WITH THE FPS CLEAR. THE RESULT OF EACH SITUATION IS CHECKED.

TEST 4 ILLEGAL FPP OP CODES AND STST TEST

THIS IS A TEST OF THE FPP OPERATION CODES:

170003
170004
:
170010
170013
170014
:
170077

THESE ARE ILLEGAL INSTRUCTIONS AND WITH INTERRUPTS ENABLED SHOULD CAUSE A TRAP TO 244. ALSO TESTED HERE IS THE INSTRUCTION: STST R1, WHICH SHOULD PUT THE FEC CODE 2 IN R1, AFTER ANY OF THE ABOVE OP CODES IS EXECUTED.

TEST 5 FID, INTERRUPT DISABLE, BIT TEST

THIS IS A TEST OF FPS BIT 14 (FID) OR FLOATING INTERRUPT DISABLE. AN ILLEGAL INSTRUCTION IS EXECUTED WITH FID-1. NO INTERRUPT SHOULD OCCUR.

TEST 6 LDD AND STD, WITH SRC AND DST MODE 1, TEST

THIS IS A TEST OF BOTH THE INSTRUCTION:
LDD (R0),AC0
AND THE INSTRUCTION:
STD AC0,(R0) MOST OF THE FAILURES ARE ISOLATED TO THE SRC OR DST FLOWS. NOTE THAT THE INTEGRITY OF AC0 HAS NOT BEEN ASSURED. THIS MEANS THAT IN SOME CASES IT WILL BE IMPOSSIBLE TO ISOLATE CERTAIN DATA PATTERN FAILURES TO EITHER THE FLOWS OR THIS ACCUMULATOR.

TEST 7 FSRC MODE 0 TEST

THIS IS A TEST OF FSRC MODE ZERO USING THE LDD AND LDF INSTRUCTIONS.

TEST 10 FDST MODE 0 TEST

THIS IS A TEST OF THE STORE INSTRUCTIONS, STD AND STF, WITH FDST MODE 0.

TEST 11 ACCUMULATORS DATA PATTERNS TEST

THIS IS A TEST OF THE FLOATING POINT PROCESSOR ACCUMULATORS.

EACH ACCUMULATOR IS TESTED IN TWO WAYS:
1 TEST PATTERN GENERATED BY FLOATING A ONE ACROSS A FIELD OF ZEROES.
2 TEST PATTERN GENERATED BY FLOATING A ZERO ACROSS A FIELD OF ONES.
EACH OF ACCUMULATORS AC0 THROUGH AC5 IS TESTED.

TEST 12 FPP ACCUMULATORS DUAL ADDRESS TEST

THIS TEST PERFORMS A DUAL ADDRESSING TEST ON THE FLOATING ACCUMULATORS. NOTE THAT ACCUMULATOR ZERO IS USED TO ACCESS ALL THE OTHERS.

TEST 13 FSRC MODE 0 WITH ILLEGAL ACCUMULATOR TEST

THIS IS A TEST OF FSRC MODE 0 WITH ACCUMULATORS 6 AND 7. USE OF EITHER OF THESE NON-EXISTENT ACCUMULATORS SHOULD RESULT IN A TRAP TO 244 WITH FEC=2 (ILLEGAL FPP INSTRUCTION).

TEST 14 FSRC MODE 2 TEST

THIS IS A TEST OF FSRC MODE 2, AUTO INCREMENT MODE.

TEST 15 FSRC MODE 4 TEST

THIS IS A TEST OF FSRC MODE 4, AUTO DECREMENT MODE.

TEST 16 FSRC MODE 2, WITH FD=0, TEST

THIS IS A TEST OF FSRC MODE 2 WITH FD=0. (AUTO INCREMENT)

TEST 17 FSRC MODE 2 WITH GR7, IMMEDIATE MODE, TEST

THIS IS A TEST OF FSRC MODE 2 USING GR7 (THE PC). THIS IS IMMEDIATE MODE.

TEST 20 FSRC MODE 3 TEST

THIS IS A TEST OF FSRC MODE 3, AUTO INCREMENT DEFERRED

TEST 21 FSRC MODE 5 TEST

THIS IS A TEST OF FSRC MODE 5, AUTO DECREMENT DEFERRED.

TEST 22 FSRC MODE 6 TEST

THIS IS A TEST OF FSRC MODE 6, INDEX MODE

TEST 23 FSRC MODE 7 TEST

THIS IS A TEST OF FSRC MODE 7, INDEX DEFERRED MODE.

TEST 24 (BUT EZBT Y8),(BUT ENBT) AND (BUT FIUV) TEST

THIS IS A TEST OF THE (BUT EZBT Y8) FORK, THE (BUT ENBT) FORK AND (BUT FIUV) FORK IN THE LOAD INSTRUCTION FLOWS.
EACH OF THE PATTERNS:

0
+NUM
-NUM
-0

IS LOADED TWICE, ONCE WITH AC>0 THEN WITH AC=0.
AFTER EACH LOAD THE FPS IS CHECK TO INSURE THAT
CONTROL WAS PASSED THROUGH WITH THE FORKS PROPERLY.

TEST 25 ADDF,ADDD,SUBF AND SUBD WITH FSRC=AC=0 TEST

THIS IS A TEST OF ADD AND SUB WITH FSRC=AC=0

TEST 26 ADDD AND SUB WITH FSRC=0

THIS IS A TEST OF ADD AND SUB WITH FSRC=0.

TEST 27 SUBD WITH AC=0 TEST

THIS IS A TEST OF SUBD WITH AC=0. BOTH POSITIVE AND
NEGATIVE FSRC'S ARE TRIED.

TEST 30 ADDD WITH AC=0 TEST

POSITIVE AND NEGATIVE FSRC'S ARE TRIED.

TEST 31 ADDF AND ADDD WITH E(AC)=E(FSRC) AND (BUT FT) TEST

THIS IS A TEST OF THE ADD INSTRUCTION WITH THE
OPERANDS HAVING EQUAL EXPONENTS. THE (BUF FT) FORK
IN THE ROUND/TRUNK FLOWS IS ALSO TESTED.

TEST 32 ADDF AND ADDD WITH E(AC) LESS THAN E(FSRC) TEST

THIS IS ATEST OF THE ADDD AND ADDF INSTRUCTIONS AND
THE ALIGN AC ALGORITHM FLOWS. THE CONSTANT (25 FOR
FLOATING, 57 FOR DOUBLE) USED IS CHECKED. THEN
SIMPLE AND WORST CASE ALIGNMENT SITUATIONS ARE
TRIED. NOTE E(AC) IS LESS THEN E(FSRC)

TEST 33 ADDF AND ADDD WITH E(AC) GREATER THAN E(FSRC) TEST

THIS IS A TEST OF THE ADDD AND ADDF INSTRUCTIONS AND
THE ALIGN FSRC ALGORITHM FLOWS. FIRST THE CONSTANT
USED IS CHECKED. THEN SIMPLE AND WORST CASE
ALIGNMENT SITUATIONS ARE TRIED. NOTE E(AC) IS
GREATER THAN E(FSRC).

TEST 34 ADDD WITH NEGATVE OPRANDS TEST

THIS IS A TEST OF THE ADDD INSTRUCTION WITH NEGATIVE
OPERANDS. EVERY COMBINATION OF OPERAND SIGNS IS
TRIED.

TEST 35 SUBD TEST

THIS IS A TEST OF THE SUBD INSTRUCTION. BOTH A POSITIVE AND A NEGATIVE NUMBER IS SUBTRACTED FROM IT SELF

TEST 36 NORMALIZE ALGORITHM TEST

THIS IS A TEST OF THE NORMALIZE FLOW ALGORITHM. TWO PATTERNS ARE USED, FIRST THE MINIMUM SITUATION REQUIRING ONE LEFT SHIFT AND THEN THE MAXIMUM SITUATION REQUIRING 56 SHIFTS.

TEST 37 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 40 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 41 LDCFD AND LDCDF TEST

THIS IS A TEST OF LDCFD AND LDCDF.

TEST 42 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 43 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 44 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 45 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 46 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 47 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 50 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 51 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 52 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 53 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 54 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 55 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 56 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 57 UNDER\OVER FLOW, USING MODD WITH TRAPS DISABLED, TEST

TEST 60 MORE MICROCODES COVERAGE, TEST

'0. LISTING

&

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

                        .TITLE  CJKDCB-A KEF11-A FP DIAG PART 1
                        ;*COPYRIGHT (C) NOV  1979
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY DIAGNOSTIC ENGINEERING
                        ;*
                        ;*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

000244                  FPVECT=244
177400                  $SWR=177400
000200                  $SWRMSK=200
000011                  TAB=11
000015                  CRLF=15

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
```

```
                                ;*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
                        .EQUIV  PS,PSW
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
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
```

```
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*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
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*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
                                .SBTTL  FPP REGISTER DEFINITIONS
000000                  AC0     =%0
000001                  AC1     =%1
000002                  AC2     =%2
000003                  AC3     =%3
000004                  AC4     =%4
000005                  AC5     =%5
000006                  AC6     =%6
```

```
                        000007                  AC7     =%7
                                                .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  001466                                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
```

```
                                .SBTTL  COMMON TAGS

                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

        001100                          .=1100
001100                          $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                          .WORD   0
001102  000                     $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000                     $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000                     $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001                     $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                          .WORD   0
001134  000                     $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                          .WORD   0
001140  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560                  $TKS:   177560                  ;;TTY KBD STATUS
001146  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000                     $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002                     $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012                     $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
001157  000                     $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07> 0 YES)
001160  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                ;;WHICH  ($REG0) WAS OBTAINED
001162  000000                  $REG0:  .WOR    0               ;;CONTAINS (($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)
```

```
 238  001224  000000                $REG21: .WORD   0               ;;CONTAINS (($REGAD)+42)
 239  001226  000000                $REG22: .WORD   0               ;;CONTAINS (($REGAD)+44)
 240  001230  000000                $REG23: .WORD   0               ;;CONTAINS (($REGAD)+46)
 241  001232  000000                $TMP0:  .WORD   0               ;;USER DEFINED
 242  001234  000000                $TMP1:  .WORD   0               ;;USER DEFINED
 243  001236  000000                $TMP2:  .WORD   0               ;;USER DEFINED
 244  001240  000000                $TMP3:  .WORD   0               ;;USER DEFINED
 245  001242  000000                $TMP4:  .WORD   0               ;;USER DEFINED
 246  001244  000000                $TMP5:  .WORD   0               ;;USER DEFINED
 247  001246  000000                $TMP6:  .WORD   0               ;;USER DEFINED
 248  001250  000000                $TMP7:  .WORD   0               ;;USER DEFINED
 249  001252  000000                $TMP10: .WORD   0               ;;USER DEFINED
 250  001254  000000                $TMP11: .WORD   0               ;;USER DEFINED
 251  001256  000000                $TMP12: .WORD   0               ;;USER DEFINED
 252  001260  000000                $TMP13: .WORD   0               ;;USER DEFINED
 253  001262  000000                $TMP14: .WORD   0               ;;USER DEFINED
 254  001264  000000                $TMP15: .WORD   0               ;;USER DEFINED
 255  001266  000000                $TMP16: .WORD   0               ;;USER DEFINED
 256  001270  000000                $TMP17: .WORD   0               ;;USER DEFINED
 257  001272  000000                $TMP20: .WORD   0               ;;USER DEFINED
 258  001274  000000                $TMP21: .WORD   0               ;;USER DEFINED
 259  001276  000000                $TMP22: .WORD   0               ;;USER DEFINED
 260  001300  000000                $TMP23: .WORD   0               ;;USER DEFINED
 261  001302  000000                $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
 262  001304  000000                $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
 263  001306  177607  000377        $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
 264  001312     077                $QUES:  .ASCII  /?/             ;;QUESTION MARK
 265  001313     015                $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
 266  001314  000012                $LF:    .ASCIZ  <12>            ;;LINE FEED
 267                                ;;**************************************************************
 268                                .SBTTL  APT MAILBOX-ETABLE
 269
 270                                ;;**************************************************************
 271                                .EVEN
 272  001316                        $MAIL:                          ;;APT MAILBOX
 273  001316  000000                $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
 274  001320  000000                $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
 275  001322  000000                $TESTN: .WORD   ATESTN          ;;TEST NUMBER
 276  001324  000000                $PASS:  .WORD   APASS           ;;PASS COUNT
 277  001326  000000                $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
 278  001330  000000                $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
 279  001332  000000                $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
 280  001334  000000                $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
 281  001336                        $ETABLE:                        ;;APT ENVIRONMENT TABLE
 282  001336     000                $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
 283  001337     000                $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
 284  001340  000000                $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
 285  001342  000000                $USWR:  .WORD   AUSWR           ;;USER SWITCHES
 286  001344  000000                $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
 287                                ;*                              BITS 15-11=CPU TYPE
 288                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
 289                                ;*                                      11/70=06,PDQ=07,Q=10
 290                                ;*                              BIT 10=REAL TIME CLOCK
 291                                ;*                              BIT  9=FLOATING POINT PROCESSOR
 292                                ;*                              BIT  8=MEMORY MANAGEMENT
 293  001346     000                $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
```

```
                                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                              900 NSEC CORE=001
                                ;*                              300 NSEC BIPOLAR=002
                                ;*                              500 NSEC MOS=003
                                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
                                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
                                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
                                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
                                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
                                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
                                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
                                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
                                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
                                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
                                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
                                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
                                $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
                                $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
                                $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
                                $DDW^:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
                                $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
                                $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
                                $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
                                $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
                                $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
                                $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
                                $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
                                $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
                                $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
                                $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
                                $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
                                $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
                                $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15


                                $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:
                                ;ITEM NUMBER
001442  062743                  .WORD   EM1
001444  062771                  .WORD   EM2
001446  063025                  .WORD   EM3
001450  063053                  .WORD   EM4


                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
        001452                          $SVPC=.                 ;SAVE PC
        000046                          .=46
000046  057322                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
        000052                          .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
        001452                          .=$SVPC                 ;; RESTORE PC
                                .SBTTL  APT PARAMETER BLOCK

                                ;;******************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;******************************************************************
        001452                          .$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  001452                          $APTHDR ;;POINT TO APT HEADER BLOCK
        001452                          .=.$X   ;;RESET LOCATION COUNTER
                                ;;******************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001452                          $APTHD:
001452  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001454  001316                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001456  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
001460  000040                  $PASTM: .WORD   40      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001462  000000                  $UNITM: .WORD   0       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001464  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
392  001466                                START:
393                                        .SBTTL  INITIALIZE THE COMMON TAGS
394                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
395  001466  012706  001100                       MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
396  001472  005026                               CLR     (R6)+           ;;CLEAR MEMORY LOCATION
397  001474  022706  001140                       CMP     #SWR,R6 ;;DONE?
398  001500  001374                               BNE     .-6             ;;LOOP BACK IF NO
399  001502  012706  001100                       MOV     #STACK,SP       ;;SETUP THE STACK POINTER
400                                        ;;INITIALIZE A FEW VECTORS
401  001506  012737  057454  000020               MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
402  001514  012737  000340  000022               MOV     #340,@#IOTVEC+2 ;;LEVEL 7
403  001522  012737  057734  000030               MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERPOR ROUTINE
404  001530  012737  000340  000032               MOV     #340,@#EMTVEC+2 ;;LEVEL 7
405  001536  012737  062070  000034               MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
406  001544  012737  000340  000036               MOV     #340,@#TRAPVEC+2;LEVEL 7
407  001552  012737  062156  000024               MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
408  001560  012737  000340  000026               MOV     #340,@#PWRVEC+2 ;;LEVEL 7
409  001566  016767  055466  055456               MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
410  001574  005067  177502                       CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
411  001600  005067  177500                       CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
412  001604  112767  000001  177303               MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
413                                        ;;INITIALIZE THE "T-BIT" TRAP VECTOR. THEN LOAD LOCATION "$RTRN", IN
414                                        ;;THE "END-OF-PASS" ($EOP) ROUTINE, WITH A "RTI" OR "RTT".
415  001612  012737  057424  000014               MOV     #$RTRN,@#TBITVEC ;;SET "T" BIT VECTOR TO $RTRN
416  001620  012737  000340  000016               MOV     #340,@#TBITVEC+2 ;;LEVEL 7
417  001626  012767  000002  055570               MOV     #RTI,$RTRN      ;;SET $RTRN TO A RTI
418  001634  012737  001662  000010               MOV     #65$,@#RESVEC   ;;TRY TO DO A RTT
419  001642  005046                               CLR     -(SP)           ;;DUMMY PS
420  001644  012746  001652                       MOV     #64$,-(SP)      ;;AND PC
421  001650  000006                               RTT                     ;;TRY THE RTT
422  001652  012767  000006  055544  64$:        MOV     #RTT,$RTRN      ;;RTT IS LEGAL--SET $RTRN TO A RTT
423  001660  000402                               BR      66$
424  001662  062706  000010          65$:        ADD     #10,SP          ;;RTT ILLEGAL--CLEAN OFF THE STACK
425  001666  012737  000012  000010  66$:        MOV     #RESVEC+2,@#RESVEC ;;RESTORE TRAP CATCHER
426  001674  005067  055532                       CLR     $TBIT           ;;CLEAR "T" BIT SWITCH
427  001700  012767  001700  177200               MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
428  001706  012767  001706  177174               MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
429                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
430                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
431  001714  013746  000004                       MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
432  001720  012737  001754  000004               MOV     #67$,@#ERRVEC   ;;SET UP ERROR VECTOR
433  001726  012767  177570  177204               MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
434  001734  012767  177570  177200               MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
435  001742  022777  177777  177170               CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
436  001750  001012                               BNE     69$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
437                                                                       ;;AND  THE HARDWARE SWR IS NOT = -1
438  001752  000403                               BR      68$             ;;BRANCH IF NO TIMEOUT
439  001754  012716  001762          67$:        MOV     #68$,(SP)       ;;SET UP FOR TRAP RETURN
440  001760  000002                               RTI
441  001762  012767  000176  177150  68$:        MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
442  001770  012767  000174  177144               MOV     #DISPREG,DISPLAY
443  001776  012637  000004          69$:        MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
444
445  002002  005067  177316                       CLR     $PASS           ;;CLEAR PASS COUNT
446  002006  132767  000200  177323               BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
447  002014  001403                               BEQ     70$             ;;YES,USE NON-APT SWITCH
```

```
                                        MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
                                70$:
                                .SBTTL  TYPE PROGRAM NAME
                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
002024  005227  177777                  INC     #-1             ;;FIRST TIME?
002030  001053                          BNE     71$             ;;BRANCH IF NO
002032  022737  057322  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
002040  001447                          BEQ     71$             ;;BRANCH IF YES
002042  104401  002110                  TYPE    ,72$            ;;TYPE ASCIZ STRING
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
002046  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
002052  001012                          BNE     73$             ;;BRANCH IF YES
002054  126727  177256  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
002062  001406                          BEQ     73$             ;;BRANCH IF YES
002064  026727  177050  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
002072  001005                          BNE     74$             ;;BRANCH IF NO
002074  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
002076  000403                          BR      74$
002100  112767  000001  177026  73$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
002106                          74$:
002106  000424                          BR      71$             ;;GET OVER THE ASCIZ
                                ;;72$:  .ASCIZ  <CRLF>*CJKDCB, KEF11-A FP DIAGNOSTIC PART 1*<CRLF>
002160                          71$:

002160                          LOOP:
002160  012706  001100                  MOV     #STACK,SP       ;SET UP STACK POINTER
002164  012737  002220  000004          MOV     #1$,@#4         ;SET UP FOR TIMEOUT IF NO MULTI-TESTER
002172  012737  000340  000006          MOV     #340,@#6
002200  012737  000002  164000          MOV     #2,@#164000     ;SET 'GO' ON MULTI-TESTER
002206  012737  000006  000004          MOV     #6,@#4          ;RESTORE TRAP CATCHER
002214  005037  000006                  CLR     @#6
002220                          1$:


                                ;;******************************************************************
                                ;*TEST 1         LDFPS, STFPS AND DATA PATHS TEST
                                ;*
                                ;*THIS IS A TEST OF THE LDFPS (LOAD FLOATING POINT STATUS) AND STFPS
                                ;*(STORE FLOATING POINT STATUS) INSTRUCTIONS. A COUNT PATTERN IS GENERATED
                                ;*AND RUN THROUGH THE FLOATING POINT STATUS REGISTER.
                                ;*DMO AND SMO ARE USED.
                                ;*NOTE THAT A MASK MUST BE USED BECAUSE SOME OF THE FPS BITS CANNOT
                                ;*BE SET.
                                ;*
                                ;;******************************************************************
002220  000004                  TST1:   SCOPE
002222  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
002224  012700  177777                  MOV     #-1,R0          ;INITIALIZE THE COUNT PATTERN.
002230  012737  002336  000244          MOV     #AERR1,@#FPVECT ;SET UP FOR UNABLE TO DECODE
002236  012737  002350  000010          MOV     #AERR2,@#10     ;FPP INSTRUCTION TRAP TO 244 OR 10.
002244  012737  002402  000004          MOV     #AERR3,@#ERRVECT ;IF EITHER INSTRUCTION
                                                                ;FAILS TO GO THROUGH THE
                                                                ;CORRECT SRC OR DST MODE AN
```

```
                                                                ;ODD ADDRESS TRAP WILL OCCUR.
002252                          A1:
002252  010004                  A11:    MOV     R0,R4
002254  042704  030020                  BIC     #30020,R4
002260  170104                          LDFPS   R4              ;TEST INSTRUCTION.

002262  012701  177777                  MOV     #-1,R1
002266  170201                  A12:    STFPS   R1              ;TEST INSTRUCTION.
002270  012737  062524  000244          MOV     #FPSPUR,@#FPVECT ;SET UP FOR UNEXPECTED TRAPS.
002276  010004                          MOV     R0,R4           ;MASK OFF UNSETTABLE BITS.
002300  042704  030020                  BIC     #30020,R4
002304  012737  062556  000004          MOV     #CPSPUR,@#ERRVECT
002312  012737  062574  000010          MOV     #CPTWO,@#10
002320  020401                          CMP     R4,R1           ;COMPARE DATA EXPECTED WITH
                                                                ;THE DATA READ.
002322  001401                          BEQ     A2              ;IF NOT EQUAL GO REPORT ERROR.
002324  104004                          ERROR   4

002326  012700  000001          A2:     MOV     #1,R0           ;NEXT PATTERN WILL BE ALL ZERO
002332  077031                          SOB     R0,A1           ;DECREMENT COUNT PATTERN
002334  000443                          BR      ADONE           ;UNTIL IT IS ZERO.


                                ;UNABLE TO DECODE FPP INSTRUCTION. TRAPPED TO 244.
002336  011637  001236          AERR1:  MOV     (SP),@#$TMP2    ;SAVE PC OF TRAP.

002342  022626                          CMP     (SP)+,(SP)+
002344  104004                  1$:     ERROR   4
002346  000436                          BR      ADONE

                                ;UNABLE TO DECODE INSTRUCTION. TRAPPED TO 10.
002350  021627  002254          AERR2:  CMP     (SP),#A11+2     ;DID TRAP OCCUR OF FPP INSTRUCTION?
002354  001405                          BEQ     1$
002356  021627  002270                  CMP     (SP),#A12+2
002362  001402                          BEQ     1$
002364  000137  062574                  JMP     @#CPTWO         ;IF NOT FPP INSTRUCTION THEN
                                                                ;REPORT SPURIOUS TRAP TO 10.

002370  011637  001236          1$:     MOV     (SP),@#$TMP2    ;OTHERWISE REPORT IR DECODE ERROR.
002374  022626                          CMP     (SP)+,(SP)+
002376  104004                  2$:     ERROR   4
002400  000421                          BR      ADONE

                                ;TRAP TO 4 HANDLER:
002402  021627  002254          AERR3:  CMP     (SP),#A11+2     ;DID THE TRAP OCCUR ON THE
002406  001405                          BEQ     1$              ;LDFPS INSTRUCTION?
002410  021627  002270                  CMP     (SP),#A12+2     ;OR THE STFPS INSTRUCTION?
002414  001407                          BEQ     2$
002416  000137  062556                  JMP     @#CPSPUR        ;IF NEITHER THEN REPORT
                                                                ;UNEXPECTED TRAP TO 4.

002422  011637  001236          1$:     MOV     (SP),@#$TMP2
002426  022626                          CMP     (SP)+,(SP)+
002430  104004                  15$:    ERROR   4
002432  000404                          BR      ADONE
```

```
                                2$:     MOV     (SP),@#$TMP2
                                        CMP     (SP)+,(SP)+
                                25$:    ERROR   4

                                ADONE:
                                        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 2         CFCC TEST
                                ;*
                                ;*THIS IS A TEST OF THE COPY CONDITION CODES INSTRUCTION, CFCC.
                                ;*
                                ;;*****************************************************************
002446  000004                TST2:   SCOPE
002450  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
002452  012700  000017                MOV     #17,R0                  ;R0 CONTAINS TO TEST PATTERN.

002456                          B1:
002456  170100                        LDFPS   R0                      ;LOAD THE TEST PATTERN

002460                          B2:
002460  170000                        CFCC                            ;COPY CONDITION CODES.

002462  013703  177776                MOV     @#PSW,R3                ;SEE IF PATTERN TRANSFERED.
002466  042703  177760                BIC     #177760,R3
002472  020003                        CMP     R0,R3
002474  001002                        BNE     BERR

002476  077011                B3:     SOB     R0,B1
002500  000422                        BR      BDONE

002502                          BERR:
002502  170201                        STFPS   R1                      ;WAS FPS MODIFIED BY CFCC?
002504  012737  002460  001236        MOV     #B2,@#$TMP2

002512  020001                        CMP     R0,R1
002514  001006                        BNE     BERR1

002516  010337  001240                MOV     R3,@#$TMP3
002522  010037  001242                MOV     R0,@#$TMP4
002526  104001                1$:     ERROR   1
002530  000762                        BR      B3

002532                          BERR1:
002532  010037  001240                MOV     R0,@#$TMP3
002536  010137  001242                MOV     R1,@#$TMP4
002542  104001                1$:     ERROR   1
002544  000754                        BR      B3

002546                          BDONE:
002546  104413                        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 3         SETF, SETD, SETI AND SETL TEST
                                ;*
                                ;*THIS IS A TEST OF THE SETF, SETD, SETI AND SETL INSTRUCTIONS.
                                ;*EACH INSTRUCTION IS EXECUTED WITH THE FPS CONTAINING
                                ;*ALL ONES AND ALSO WITH THE FPS CLEAR. THE RESULT OF EACH
                                ;* SITUATION IS CHECKED.
                                ;*$TEMP2 CONTAINS INSTRUCTION TRIED TO PERFORM
                                ;*$BDDAT CONTAINS BAD FPS CONTAIN.
                                ;;**************************************************************
002550  000004               TST3:  SCOPF
002552  104414                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
002554  012737  000760 001244       MOV     #760,@#$TMP5
002562  012737  000202 001250 C1:    MOV     #202,@#$TMP7
002570  005000                      CLR     R0

002572  170100                      LDFPS   R0                      ;CLEAR THE FPS.
002574  012737  002602 001236       MOV     #C15,@#$TMP2

002602  170001              C15:    SETF                            ;TEST INSTRUCTION.


002604  170201                      STFPS   R1                      ;GET RESULT.
002606  005002                      CLR     R2
002610  020201                      CMP     R2,R1                   ;DID AN ERROR OCCUR?
002612  001403                      BEQ     1$
002614  010137  001126              MOV     R1,@#$BDDAT     ;STORE BAD DATA
002620  104001                      ERROR   1

002622                       1$:
002622  104414                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
002624  012700  147757       C2:    MOV     #147757,R0

002630  170100                      LDFPS   R0                      ;PUT 147757 IS FPS
002632  012737  002640 001236       MOV     #C25,@#$TMP2
002640  170001              C25:    SETF                            ;CLEAR FD BIT.

002642  170201                      STFPS   R1                      ;GET RESULT
002644  012702  147557              MOV     #147557,R2
002650  020102                      CMP     R1,R2                   ;RESULT CORRECT.
002652  001403                      BEQ     1$
002654  010137  001126              MOV     R1,@#$BDDAT     ;STORE BAD DATA
002660  104001                      ERROR   1

002662                       1$:
002662  104414                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
002664  012737  000203 001250 C3:    MOV     #203,@#$TMP7
002672  012700  147757              MOV     #147757,R0

002676  170100                      LDFPS   R0                      ;LOAD 147757 INTO FPS.
```

```
672  002700  012737  002706  001236          MOV     #C35,@#$TMP2
673  002706  170011                  C35:    SETD                            ;SETD FD BIT.
674
675  002710  170201                          STFPS   R1
676  002712  012702  147757                  MOV     #147757,R2
677  002716  020102                          CMP     R1,R2                   ;RESULT CORRECT?
678  002720  001403                          BEQ     1$
679  002722  010137  001126                  MOV     R1,@#$BDDAT     ;STORE BAD DATA
680  002726  104001                          ERROR   1
681
682  002730                          1$:
683  002730  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
684  002732  005000                  C4:     CLR     R0
685  002734  170100                          LDFPS   R0                      ;CLEAR FPS.
686  002736  012737  002744  001236          MOV     #C45,@#$TMP2
687
688  002744  170011                  C45:    SETD                            ;SET FD BIT.
689
690  002746  170201                          STFPS   R1                      ;GET RESULT.
691  002750  012702  000200                  MOV     #200,R2
692  002754  020102                          CMP     R1,R2                   ;RESULT CORRECT?
693  002756  001403                          BEQ     1$
694  002760  010137  001126                  MOV     R1,@#$BDDAT     ;STORE BAD DATA
695  002764  104001                          ERROR   1
696
697  002766                          1$:
698  002766  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
699  002770  012737  000204  001250  C5:     MOV     #204,@#$TMP7
700  002776  005000                          CLR     R0
701
702  003000  170100                          LDFPS   R0                      ;CLEAR FPS
703  003002  012737  003010  001236          MOV     #C55,@#$TMP2
704
705  003010  170002                  C55:    SETI                            ;CLEAR FL BIT.
706
707  003012  170201                          STFPS   R1                      ;GET RESULT.
708  003014  005002                          CLR     R2
709  003016  020201                          CMP     R2,R1                   ;RESULT CORRECT?
710  003020  001403                          BEQ     1$
711  003022  010137  001126                  MOV     R1,@#$BDDAT     ;STORE BAD DATA
712  003026  104001                          ERROR   1
713
714  003030                          1$:
715  003030  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
716  003032  012700  147757          C6:     MOV     #147757,R0
717  003036  170100                          LDFPS   R0                      ;PUT 147757 INTO FPS
718  003040  012737  003046  001236          MOV     #C65,@#$TMP2
719
720  003046  170002                  C65:    SETI                            ;CLEAR FL BIT.
721
722  003050  170201                          STFPS   R1                      ;GET THE RESULT.
723  003052  012702  147657                  MOV     #147657,R2
724  003056  020102                          CMP     R1,R2                   ;RESULT CORRECT?
725  003060  001403                          BEQ     1$
726  003062  010137  001126                  MOV     R1,@#$BDDAT     ;STORE BAD DATA
727  003066  104001                          ERROR   1
```

```
003070                            1$:
003070  104414                            LPERR                      ;SET UP THE LOOP ON ERROR ADDRESS.
003072  012737  000205  001250    C7:     MOV     #205,@#$TMP7
003100  012700  147757                    MOV     #147757,R0
003104  170100                            LDFPS   R0                          ;SET FPS TO 147757.
003106  012737  003114  001236            MOV     #C75,@#$TMP2

003114  170012                    C75:    SETL                                ;SET FL BIT.

003116  170201                            STFPS   R1                          ;GET THE RESULT.
003120  012702  147757                    MOV     #147757,R2
003124  020102                            CMP     R1,R2                       ;RESULT CORRECT?
003126  001403                            BEQ     1$
003130  010137  001126                    MOV     R1,@#$BDDAT        ;STORE BAD DATA
003134  104001                            ERROR   1

003136                            1$:
003136  104414                            LPERR                      ;SET UP THE LOOP ON ERROR ADDRESS.
003140  005000                    C8:     CLR     R0
003142  170100                            LDFPS   R0                          ;CLEAR FPS.
003144  012737  003152  001236            MOV     #C85,@#$TMP2

003152  170012                    C85:    SETL                                ;SET FL BIT.

003154  170201                            STFPS   R1
003156  012702  000100                    MOV     #100,R2
003162  020102                            CMP     R1,R2                       ;RESULT CORRECT.
003164  001403                            BEQ     1$
003166  010137  001126                    MOV     R1,@#$BDDAT        ;STORE BAD DATA
003172  104001                            ERROR   1

003174                            1$:
003174                            CDONE:
003174  104413                            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 4         ILLEGAL FPP OP CODES AND STST TEST
                                  ;*
                                  ;*THIS IS A TEST OF THE FPP OPERATION CODES:
                                  ;*                       170003
                                  ;*                       170004
                                  ;*                          :
                                  ;*                       170010
                                  ;*                       170013
                                  ;*                       170014
                                  ;*                          :
                                  ;*                       170077
                                  ;*THESE ARE ILLEGAL INSTRUCTIONS AND (WITH INTERRUPTS ENABLED)
                                  ;*SHOULD CAUSE A TRAP TO 244.
                                  ;*ALSO TESTED HERE IS THE INSTRUCTION:
```

```
                                        ;*                  STST    R1
                                        ;*WHICH SHOULD PUT THE FEC CODE 2 IN R1, AFTER ANY OF THE ABOVE
                                        ;*OP CODES IS EXECUTED.
                                        ;*
                                        ;;*****************************************************************
003176  000004                  TST4:   SCOPE
003200  104414                          LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
003202  012705  170003                   MOV     #170003,R5          ;INITIAL OP CODE.
003206  012737  003412  000004          MOV     #DERR2,@#ERRVECT
003214  012737  003316  000244          MOV     #DERR1,@#FPVECT

003222  005000                  D1:     CLR     R0
003224  170100                          LDFPS   R0                  ;CLEAR FPS.
003226  005002                          CLR     R2
003230  010537  003246                   MOV     R5,@#D2             ;SET UP THE ILLEGAL INSTRUCTION.
003234  010537  001244                   MOV     R5,@#$TMP5
003240  012737  003246  001236          MOV     #D2,@#$TMP2
003246  000000                  D2:     .WORD   0
003250  170000                  D3:     CFCC
003252  005202                          INC     R2
003254  005202                  D4:     INC     R2

003256  170201                          STFPS   R1                  ;REPORT FAILURE. DID NOT TRAP.
003260  010137  001240                   MOV     R1,@#$TMP3
003264  104001                  1$:     ERROR   1

003266  022705  170010          D5:     CMP     #170010,R5          ;COMPUTE NEXT OP CODE
003272  001003                          BNE     D6
003274  012705  170013                   MOV     #170013,R5
003300  000750                          BR      D1

003302  022705  170077          D6:     CMP     #170077,R5
003306  001001                          BNE     D7
003310  000452                          BR      DDONE
003312  005205                  D7:     INC     R5
003314  000742                          BR      D1

003316  022716  003250          DERR1:  CMP     #D3,(SP)            ;DID TRAP OCCUR ON TEST INSTRUCTION?
003322  001402                          BEQ     1$
003324  000137  062524                   JMP     @#FPSPUR

003330  022626                  1$:     CMP     (SP)+,(SP)+
003332  170201                          STFPS   R1                  ;GET THE FPS AND SEE IF IT IS
003334  022701  100000                   CMP     #100000,R1          ;SET CORRECTLY.
003340  001406                          BEQ     3$

003342  012737  100000  001240          MOV     #100000,@#$TMP3
003350  010137  001242                   MOV     R1,@#$TMP4
003354  104001                  2$:     ERROR   1

003356  012704  000001          3$:     MOV     #1,R4
003362  170304                  D8:     STST    R4                  ;GET THE FEC CODE. NOTE THAT
                                                                    ;IF THE DESTINATION MODE IS
                                                                    ;IMPROPERLY DECODED AN ODD
                                                                    ;ADDRESS TRAP TO 4 SHOULD OCCUR.
003364  022704  000002                   CMP     #2,R4               ;WAS FEC CORRECT?
```

```
 840  003370  001001                          BNE     D9
 841  003372  000735                          BR      D5
 842
 843  003374                          D9:                                     ;REPORT STST FAILURE
 844  003374  012737  003362  001240          MOV     #D8,@#$TMP3
 845  003402  010437  001242                  MOV     R4,@#$TMP4
 846  003406  104001                  1$:     ERROR   1
 847  003410  000726                          BR      D5
 848
 849  003412  022716  003364          DERR2:  CMP     #D8+2,(SP)              ;DID THE TRAP OCCUR ON THE
 850  003416  001402                          BEQ     D10                     ;STST INSTRUCTION?
 851  003420  000137  062556                  JMP     @#CPSPUR
 852
 853  003424                          D10:
 854  003424  011637  001236                  MOV     (SP),@#$TMP2
 855  003430  022626                          CMP     (SP)+,(SP)+
 856  003432  104001                  1$:     ERROR   1
 857  003434  000714                          BR      D5.
 858
 859  003436                          DDONE:
 860  003436  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 861                                                                  ;SEE IF THE USER HAS EXPRESSED
 862                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 863                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 864                                                                  ;THE USER TYPED CONTROL G?).
 865
 866
 867                                  ;;*****************************************************************
 868                                  ;*TEST 5        FID, INTERRUPT DISABLE, BIT TEST
 869                                  ;*
 870                                  ;*THIS IS A TEST OF FPS BIT 14 (FID) OR FLOATING INTERRUPT DISABLE.
 871                                  ;*AN ILLEGAL INSTRUCTION IS EXECUTED WITH FID=1. NO INTERRUPT SHOULD
 872                                  ;*OCCUR.
 873                                  ;*
 874                                  ;;*****************************************************************
 875  003440  000004                  TST5:   SCOPE
 876  003442  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 877  003444  012737  003546  000244          MOV     #EERR2,@#FPVECT         ;SETUP FOR THE INTERRUPT.
 878
 879  003452  012700  040000          E1:     MOV     #40000,R0
 880  003456  170100                          LDFPS   R0                      ;SET FID.
 881  003460  012737  003466  001236          MOV     #E3,@#$TMP2
 882  003466                          E2:
 883  003466  170020                  E3:     .WORD   170020                  ;ILLEGAL FPP INSTRUCTION.
 884  003470  170000                  E4:     CFCC
 885
 886  003472  170201                          STFPS   R1                      ;SEE IF ERROR WAS DETECTED.
 887  003474  022701  140000                  CMP     #140000,R1
 888  003500  001005                          BNE     EERR0
 889
 890  003502  170304                          STST    R4                      ;SEE IF FEC=2
 891  003504  022704  000002                  CMP     #2,R4
 892  003510  001010                          BNE     EERR1
 893  003512  000431                          BR      EDONE
 894
 895  003514                          EERR0:                                  ;REPORT FPS INCORRECTLY SET.
```

```
                                        MOV     R1,@#$TMP3
                                        MOV     #140000,@#$TMP4
                                1$:     ERROR   1
                                        BR      EDONE

                                EERR1:                                  ;REPORT FEC NOT 2.
                                        MOV     R5,@#$TMP3
                                        MOV     R4,@#$TMP4
                                1$:     ERROR   1
                                        BR      EDONE

                                EERR2:  CMP     (SP),#E4                ;DID THE ILLEGAL INSTRUCTION TRAP?
                                        BEQ     1$
                                        JMP     @#FPSPUR

                                1$:
                                        MOV     (SP),@#$TMP2
                                        CMP     (SP)+,(SP)+
                                        STFPS   R1
                                        MOV     R1,@#$TMP3
                                2$:     ERROR   1

                                EDONE:
                                        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 6         LDD AND STD, WITH SRC AND DST MODE 1, TEST
                                ;*
                                ;*THIS IS A TEST OF BOTH THE INSTRUCTION:
                                ;*              LDD     (R0),AC0
                                ;*AND THE INSTRUCTION:
                                ;*              STD     AC0,(R0)
                                ;*MOST OF THE FAILURES ARE ISOLATED TO THE SRC OR DST FLOWS. NOTE
                                ;*THAT THE INTEGRITY OF AC0 HAS NOT BEEN ASSURED. THIS MEANS THAT
                                ;*IN SOME CASES IT WILL BE IMPOSSIBLE TO ISOLATE CERTAIN DATA PATTERN
                                ;*FAILURES TO EITHER THE FLOWS OR THIS ACCUMULATOR.
                                ;*
                                ;;*****************************************************************
003600  000004                  TST6:   SCOPE

003602                          F1:
003602  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
003604  012737  003654  001236          MOV     #F3,@#$TMP2
003612  005000                          CLR     R0
003614  170100                          LDFPS   R0
003616  170011                          SETD
003620  012701  005372                  MOV     #FDATI0,R1              ;SET UP THE LOAD DATA.
003624  012702  005436                  MOV     #FXDAT0,R2
003630  012703  000010                  MOV     #10,R3

003634  012221                  F2:     MOV     (R2)+,(R1)+
```

```
 952  003636  077302                        SOB     R3,F2
 953
 954  003640  012700  005402                MOV     #FDATI4,R0              ;SETUP R0 FOR THE LDD (R0),AC0.
 955  003644  012737  005072  000004        MOV     #FERR20,@#ERRVECT       ;IF THE SRC FLOWS FAIL THEN
 956                                                                          ;AN ODD ADDRESS MAY OCCUR.
 957  003652  005003                        CLR     R3
 958
 959  003654  172410                F3:     LDD     (R0),AC0
 960  003656  005203                F4:     INC     R3
 961  003660  005203                        INC     R3
 962
 963  003662  020027  005402                CMP     R0,#FDATI4              ;WAS R0 AFFECTED?
 964  003666  001402                        BEQ     F5
 965  003670  000137  004236                JMP     @#FERR1
 966
 967  003674  020327  000002        F5:     CMP     R3,#2                   ;SEE IF THE PC WAS ADVERSELY
 968  003700  001402                        BEQ     1$                      ;AFFECTED DURING THE INSTRUCTION.
 969  003702  000137  004334                JMP     @#FERR2
 970
 971  003706  012701  005372        1$:     MOV     #FDATI0,R1              ;MAKE SURE THE SOURCE DATA WAS
 972  003712  012702  005436                MOV     #FXDAT0,R2              ;NOT AFFECTED.
 973  003716  012703  000010                MOV     #10,R3
 974  003722  022122                2$:     CMP     (R1)+,(R2)+
 975  003724  001402                        BEQ     3$
 976  003726  000137  004200                JMP     @#FERR0
 977  003732  077305                3$:     SOB     R3,2$
 978
 979  003734  170201                        STFPS   R1                      ;MAKE SURE THE FPS IS CORRECT.
 980  003736  022701  000200                CMP     #200,R1
 981  003742  001402                        BEQ     F6
 982  003744  000137  005052                JMP     @#FERR11
 983
 984  003750                        F6:
 985  003750  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 986  003752  012737  004014  001236        MOV     #F10,@#$TMP2
 987
 988  003760  012703  177777                MOV     #-1,R3
 989  003764  012704  000010                MOV     #10,R4
 990  003770  012705  005414                MOV     #FDAT00,R5              ;SET UP THE OUTPUT DATA BUFFER.
 991  003774  010325                F7:     MOV     R3,(R5)+
 992  003776  077402                        SOB     R4,F7
 993
 994  004000  012700  005424                MOV     #FDATI4,R0              ;SET UP R0 FOR DST MODE 1 REG 0.
 995  004004  012737  005232  000004        MOV     #FERR25,@#ERRVECT       ;IF THE DST FLOWS FAIL AN ODD
 996                                                                          ;ADDRESS COULD OCCUR.
 997  004012  005003                        CLR     R3
 998
 999  004014  174010                F10:    STD     AC0,(R0)                ;TEST INSTRUCTION.
1000  004016  005203                F11:    INC     R3
1001  004020  005203                        INC     R3
1002
1003  004022  020027  005424                CMP     R0,#FDATI4              ;WAS R0 MODIFIED?
1004  004026  001402                        BEQ     F12
1005  004030  000137  004374                JMP     @#FERR3
1006
1007  004034  020327  000002        F12:    CMP     R3,#2                   ;WAS THE PC AFFECTED CORRECTLY?
```

```
1008  004040  001402                      BEQ     F135
1009  004042  000167  000320              JMP     FERR4
1010
1011  004046  012701  005414      F135:   MOV     #FDAT00,R1
1012  004052  012702  005436              MOV     #FXDAT0,R2
1013
1014  004056  022122                      CMP     (R1)+,(R2)+             ;SEE IF THE DATA WAS OUTPUT
1015  004060  001402                      BEQ     F13                     ;TO THE TARGET AREA CORRECTLY.
1016  004062  000137  004472              JMP     @#FERR5
1017
1018  004066  022122      F13:    CMP     (R1)+,(R2)+
1019  004070  001402              BEQ     F14
1020  004072  000137  004472              JMP     @#FERR5
1021
1022  004076  022122      F14:    CMP     (R1)+,(R2)+
1023  004100  001402              BEQ     F15
1024  004102  000137  004472              JMP     @#FERR5
1025
1026  004106  022122      F15:    CMP     (R1)+,(R2)+
1027  004110  001402              BEQ     F16
1028  004112  000137  004472              JMP     @#FERR5
1029
1030  004116  022122      F16:    CMP     (R1)+,(R2)+
1031  004120  001402              BEQ     F17
1032  004122  000137  005016              JMP     @#FERR10
1033
1034  004126  022122      F17:    CMP     (R1)+,(R2)+
1035  004130  001402              BEQ     F20
1036  004132  000137  004526              JMP     @#FERR6
1037
1038  004136  022122      F20:    CMP     (R1)+,(R2)+
1039  004140  001402              BEQ     F21
1040  004142  000167  000514              JMP     FERR7
1041
1042  004146  022122      F21:    CMP     (R1)+,(R2)+
1043  004150  001402              BEQ     F22
1044  004152  000137  005016              JMP     @#FERR10
1045
1046  004156  005001      F22:    CLR     R1
1047  004160  170201              STFPS   R1                              ;MAKE SURE FPS IS CORRECT.
1048  004162  022701  000200              CMP     #200,R1
1049  004166  001402              BEQ     F23
1050  004170  000137  005052              JMP     @#FERR11
1051  004174  000137  005456      F23:    JMP     @#FDONE
1052
1053  004200              FERR0:                                          ;SOURCE DATA AFFECTED BY
1054  004200  012737  005436  001240      MOV     #FXDAT0,@#$TMP3         ;THE LDD INSTRUCTION.
1055  004206  012737  005450  001242      MOV     #FXDAT0+12,@#$TMP4
1056  004214  012737  005372  001244      MOV     #FDATI0,@#$TMP5
1057  004222  012737  005404  001246      MOV     #FDATI0+12,@#$TMP6
1058  004230  104001              1$:     ERROR   1
1059  004232  000137  005456              JMP     @#FDONE
1060
1061  004236  012737  005402  001242  FERR1:  MOV     #FDATI4,@#$TMP4         ;FSRC FLOWS FAILURE.
1062  004244  010037  001240              MOV     R0,@#$TMP3
1063  004250  012737  000762  001244      MOV     #762,@#$TMP5
```

```
1064  004256  012737  000321  001250          MOV     #321,@#$TMP7
1065
1066  004264  022700  005372                  CMP     #FDATI0,R0              ;FSRC MODE 4?
1067  004270  001004                          BNE     1$
1068  004272  012737  000324  001246          MOV     #324,@#$TMP6
1069  004300  000412                          BR      4$
1070
1071  004302  022700  005412          1$:     CMP     #FDATI4+10,R0           ;FSRC MODF 2?
1072  004306  001004                          BNE     2$
1073  004310  012737  000322  001246          MOV     #322,@#$TMP6
1074  004316  000403                          BR      4$
1075
1076  004320                           2$:
1077  004320                           3$:
1078  004320  104001                          ERROR   1               ;R0 WAS MODIFED, LDD
1079  004322  000137  005456                  JMP     @#FDONE
1080  004326                           4$:
1081  004326                           5$:
1082  004326  104001                          ERROR   1               ;R0 WAS MODIFED, LDD
1083  004330  000137  005456                  JMP     @#FDONE
1084
1085
1086  004334  012701  003656          FERR2:  MOV     #F4,R1                  ;THE PC WAS INCORRECTLY AFFECTED
1087                                                                          ;DURING THE INSTRUCTION.
1088  004340  010137  001242          FER2:   MOV     R1,@#$TMP4
1089  004344  162701  000004                  SUB     #4,R1
1090  004350  006303                          ASL     R3
1091  004352  060301                          ADD     R3,R1
1092  004354  010137  001240                  MOV     R1,@#$TMP3
1093  004360  104001                  1$:     ERROR   1
1094  004362  000137  005456                  JMP     @#FDONE
1095
1096  004366  012701  00401o          FERR4:  MOV     #F11,R1
1097  004372  000762                          BR      FER2
1098
1099  004374  012737  005424  001242  FERR3:  MOV     #FDAT04,@#$TMP4         ;FAILURE IN THE FDST FLOWS.
1100  004402  010037  001240                  MOV     R0,@#$TMP3
1101  004406  012737  000527  001244          MOV     #527,@#$TMP5
1102  004414  012737  000641  001250          MOV     #641,@#$TMP7
1103
1104  004422  022700  005414                  CMP     #FDAT00,R0              ;DST MODE 4?
1105  004426  001004                          BNE     1$
1106  004430  012737  000644  001246          MOV     #644,@#$TMP6
1107  004436  000412                          BR      4$
1108
1109  004440  022700  005434          1$:     CMP     #FDAT04+10,R0           ;DST MODE 2?
1110  004444  001004                          BNE     2$
1111  004446  012737  000642  001246          MOV     #642,@#$TMP6
1112  004454  000403                          BR      4$
1113
1114  004456                           2$:
11    004456  104001                  3$:     ERROR   1               ;R0 WAS MODIFIED, STD
111o  004460  000137  005456                  JMP     @#FDONE
1117
1118  004464                           4$:
1119  004464  104001                  5$:     ERROR   1               ;R0 WAS MODIFIED, STD
```

```
1120
1121  004466  000137  005456            JMP     @#FDONE
1122
1123  004472                    FERR5:                          ;FAILURE OF STD.
1124  004472  010037  001240            MOV     R0,@#$TMP3
1125  004476  012737  005414  001242    MOV     #FDAT00,@#$TMP4
1126  004504  012737  005432  001244    MOV     #FDAT07,@#$TMP5
1127  004512  012737  005436  001246    MOV     #FXDAT0,@#$TMP6
1128  004520  104003            1$:     ERROR   3               ;OUTPUT BAD, STD
1129  004522  000137  005456            JMP     @#FDONE
1130
1131  004526  012701  005426    FERR6:  MOV     #FDAT05,R1              ;DID (BUT GR7) FAIL IN THE FDST
1132  004532  012702  177777            MOV     #-1,R2                  ;FLOWS?
1133  004536  012703  000003            MOV     #3,R3
1134  004542  020221            1$:     CMP     R2,(R1)+
1135  004544  001017                    BNE     5$
1136  004546  077303                    SOB     R3,1$
1137
1138                                                                    ;REPORT FAILURE OF (BUT GR7) IN
1139  004550  010037  001240            MOV     R0,@#$TMP3              ;THE FDST FLOWS.
1140  004554  012737  000412  001244    MOV     #412,@#$TMP5
1141  004562  012737  000147  001246    MOV     #147,@#$TMP6
1142  004570  012737  000145  001250    MOV     #145,@#$TMP7
1143  004576  104001            2$:     ERROR   1
1144  004600  000137  005456            JMP     @#FDONE
1145
1146  004604  012701  005426    5$:     MOV     #FDAT05,R1              ;DID (BUT GR7) FAIL IN THE SRC FLOWS?
1147  004610  012703  000003            MOV     #3,R3
1148  004614  005721            6$:     TST     (R1)+
1149  004616  001402                    BEQ     7$
1150  004620  000137  005016            JMP     @#FERR10
1151  004624  077305            7$:     SOB     R3,6$
1152
1153                                                                    ;REPORT FAILURE OF (BUT GR7) IN
1154  004626  010037  001240            MOV     R0,@#$TMP3              ;THE FSRC FLOWS.
1155  004632  012737  000207  001244    MOV     #207,@#$TMP5
1156  004640  012737  000176  001246    MOV     #176,@#$TMP6
1157  004646  012737  000174  001250    MOV     #174,@#$TMP7
1158
1159  004654  104001            10$:    ERROR   1
1160  004656  000137  005456            JMP     @#FDONE
1161
1162  004662  012701  005430    FERR7:  MOV     #FDAT06,R1              ;DID (BUT FD) FAIL IN THE FDST FLOWS?
1163  004666  012702  177777            MOV     #-1,R2
1164  004672  012703  000002            MOV     #2,R3
1165  004676  020221            1$:     CMP     R2,(R1)+
1166  004700  001017                    BNE     5$
1167  004702  077303                    SOB     R3,1$
1168
1169                                                                    ;REPORT FAILURE OF (BUT FD) IN THE
1170  004704  010037  001240            MOV     R0,@#$TMP3              ;FDST FLOWS.
1171  004710  012737  000707  001244    MOV     #707,@#$TMP5
1172  004716  012737  000244  001246    MOV     #244,@#$TMP6
1173  004724  012737  000245  001250    MOV     #245,@#$TMP7
1174  004732  104001            2$:     ERROR   1
1175  004734  000137  005456            JMP     @#FDONE
```

```
1176
1177  004740  012701  005430          5$:     MOV     #FDAT06,R1              ;DID (BUT FD) FAIL IN THE FSRC FLOWS?
1178  004744  012703  000002                  MOV     #2,R3
1179  004750  005721                  6$:     TST     (R1)+
1180  004752  001402                          BEQ     7$
1181  004754  000137  005016                  JMP     @#FERR10
1182  004760  077305                  7$:     SOB     R3,6$
1183
1184                                                                          ;REPORT FAILURE OF (BUT FD) IN THE
1185  004762  010037  001240                  MOV     R0,@#$TMP3              ;FSRC FLOWS.
1186  004766  012737  000441  001244          MOV     #441,@#$TMP5
1187  004774  012737  000076  001246          MOV     #76,@#$TMP6
1188  005002  012737  000077  001290          MOV     #77,@#$TMP7
1189  005010  104003                  10$:    ERROR   3                       ;BAD DATA
1190  005012  000137  005456                  JMP     @#FDONE
1191
1192  005016                          FERR10:                                 ;REPORT DATA ERROR.
1193  005016  010037  001240                  MOV     R0,@#$TMP3
1194  005022  012737  005424  001242          MOV     #FDAT04,@#$TMP4
1195  005030  012737  005432  001244          MOV     #FDAT07,@#$TMP5
1196  005036  012737  005446  001246          MOV     #FXDAT4,@#$TMP6
1197  005044  104001                  1$:     ERROR   1
1198  005046  000137  005456                  JMP     @#FDONE
1199
1200  005052                          FERR11:                                 ;REPORT BAD FPS.
1201  005052  010137  001240                  MOV     R1,@#$TMP3
1202  005056  012737  000200  001242          MOV     #200,@#$TMP4
1203  005064  104001                  1$:     ERROR   1
1204  005066  000137  005456                  JMP     @#FDONE
1205
1206  005072  012737  062735  001264  FERR20: MOV     #NULL,@#$TMP15          ;THE EXECUTION OF THE LDD
1207  005100  005037  001252                  CLR     @#$TMP10                ;CAUSED A TRAP TO 4, BECAUSE
1208  005104  011637  001236                  MOV     (SP),@#$TMP2            ;A FSRC FLOW FAILURE RESULTED
1209  005110  012737  005402  001240          MOV     #FDATI4,@#$TMP3         ;IN AN ODD ADDRESS.
1210  005116  012737  000321  001250          MOV     #321,@#$TMP7
1211  005124  012737  000762  001244          MOV     #762,@#$TMP5
1212
1213  005132  021627  003660                  CMP     (SP),#F4+2              ;SEE IF FSRC MODE 6 OR 7 WAS
1214  005136  001424                          BEQ     FERR21                  ;EXECUTED.
1215
1216  005140  020027  005400                  CMP     R0,#FDATI3              ;FSRC MODE 5?
1217  005144  001006                          BNE     2$
1218
1219                                                                          ;REPORT FSRC FLOW FAILURE TO
1220  005146  012737  000325  001246          MOV     #325,@#$TMP6            ;MODE 5.
1221  005154  022626                          CMP     (SP)+,(SP)+
1222  005156  104001                  1$:     ERROR   1
1223  005160  000536                          BR      FDONE
1224
1225  005162  020027  005404          2$:     CMP     R0,#FDATI5              ;FSRC MODE 3?
1226  005166  001402                          BEQ     3$
1227  005170  000137  062556                  JMP     @#CPSPUR
1228
1229  005174                          3$:                                     ;REPORT FSRC     FLOW FAILURE TO
1230  005174  012737  000323  001246          MOV     #323,@#$TMP6            ;MODE 3.
1231  005202  022626                          CMP     (SP)+,(SP)+
```

```
1232  005204  104001                  4$:     ERROR   1
1233  005206  000523                          BR      FDONE
1234
1235  005210  022626                  FERR21: CMP     (SP)+,(SP)+             ;REPORT FSRC FLOW FAILURE TO
1236                                                                          ;MODE 6 OR MODE 7.
1237  005212  012737  000326  001246          MOV     #326,@#$TMP6
1238  005220  012737  000327  001252          MOV     #327,@#$TMP10
1239  005226  104001                  1$:     ERROR   1
1240  005230  000512                          BR      FDONE
1241
1242  005232  012737  062735  001264  FERR25: MOV     #NULL,@#$TMP15          ;THE EXECUTION OF THE STD INSTRUCTION
1243  005240  005037  001252                  CLR     @#$TMP10                ;TRAPPED TO 4, BECAUSE A FAILURE
1244  005244  012737  005424  001240          MOV     #FDAT04,@#$TMP3         ;IN THE FDST FLOWS RESULTED
1245  005252  011637  001236                  MOV     (SP),@#$TMP2            ;IN AN ODD ADDRESS.
1246  005256  012737  000527  001244          MOV     #527,@#$TMP5
1247  005264  012737  000641  001250          MOV     #641,@#$TMP7
1248
1249  005272  021627  004016                  CMP     (SP),#F10+2             ;FLOW FAILURE TO FDST MODE 6 OR 7?
1250  005276  001424                          BEQ     FERR26
1251
1252  005300  020027  005422                  CMP     R0,#FDAT03              ;DID FDST FLOW FAIL TO MODE 5?
1253  005304  001006                          BNE     2$
1254
1255                                                                          ;REPORT FLOW FAILURE TO FDST
1256  005306  012737  000645  001246          MOV     #645,@#$TMP6            ;MODE 5.
1257  005314  022626                          CMP     (SP)+,(SP)+
1258  005316  104001                  1$:     ERROR   1
1259  005320  000456                          BR      FDONE
1260
1261  005322  020027  005426          2$:     CMP     R0,#FDAT05              ;DID FDST FLOW FAIL TO MODE 3?
1262  005326  001402                          BEQ     3$
1263  005330  000137  062556                  JMP     @#CPSPUR
1264
1265  005334                          3$:                                     ;REPORT FDST FLOW FAILED TO MODE 3.
1266  005336  012737  000643  001246          MOV     #643,@#$TMP6
1267  005342  022626                          CMP     (SP)+,(SP)+
1268  005344  104001                  4$:     ERROR   1
1269  005346  000443                          BR      FDONE
1270
1271  005350                          FERR26:                                 ;REPORT FDST FLOW FAILURE TO MODE
1272  005350  012737  000646  001246          MOV     #646,@#$TMP6
1273  005356  012737  000647  001252          MOV     #647,@#$TMP10
1274  005364  022626                          CMP     (SP)+,(SP)+
1275  005366  104001                  1$:     ERROR   1
1276  005370  000432                          BR      FDONE
1277
1278  005372  177777                  FDATI0: -1
1279  005374  177777                  FDATI1: -1
1280  005376  177777                  FDATI2: -1
1281  005400  177777                  FDATI3: -1
1282  005402  177777                  FDATI4: -1
1283  005404  177777                  FDATI5: -1
1284  005406  177777                  FDATI6: -1
1285  005410  177777                  FDATI7: -1
1286  005412  177777                          -1
1287  005414  177777                  FDAT00: -1
```

```
 1288  005416  177777                FDAT01: -1
 1289  005420  177777                FDAT02: -1
 1290  005422  177777                FDAT03: -1
 1291  005424  177777                FDAT04: -1
```

```
1292  005426  177777                  FDAT05: -1
1293  005430  177777                  FDAT06: -1
1294  005432  177777                  FDAT07: -1
1295  005434  177777                          -1
1296  005436  177777                  FXDAT0: -1
1297  005440  177777                  FXDAT1: -1
1298  005442  177777                  FXDAT2: -1
1299  005444  177777                  FXDAT3: -1
1300  005446  052525                  FXDAT4: 052525
1301  005450  031463                  FXDAT5: 031463
1302  005452  007417                  FXDAT6: 007417
1303  005454  000477                  FXDAT7: 000477
1304
1305
1306  005456                          FDONE:
1307  005456  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
1308                                                                  ;SEE IF THE USER HAS EXPRESSED
1309                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
1310                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
1311                                                                  ;THE USER TYPED CONTROL G?).
1312
1313
1314                                  ;;*****************************************************************
1315                                  ;*TEST 7        FSRC MODE 0 TEST
1316                                  ;*
1317                                  ;*THIS IS A TEST OF FSRC MODE ZERO USING THE LDD AND LDF INSTRUCTIONS.
1318                                  ;*
1319                                  ;;*****************************************************************
1320  005460  000004                  TST7:   SCOPE
1321  005462  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1322
1323  005464                          I1:
1324  005464  170011                          SETD                            ;SET FD.
1325  005466  012700  006274                  MOV     #IDAT10,R0
1326  005472  012701  006244                  MOV     #IPAT10,R1
1327  005476  012702  000004                  MOV     #4,R2
1328  005502  012120                  I2:     MOV     (R1)+,(R0)+             ;SET UP THE INPUT DATA BUFFER.
1329  005504  077202                          SOB     R2,I2
1330
1331  005506  012700  006274                  MOV     #IDAT10,R0              ;LOAD AC1
1332  005512  172510                          LDD     (R0),AC1
1333
1334  005514  012700  006254                  MOV     #IPAT20,R0              ;LOAD AC0
1335  005520  172410                          LDD     (R0),AC0
1336
1337  005522  012701  000001                  MOV     #1,R1                   ;IN CASE THE FSRC FLOWS FAIL
1338  005526  012737  006044  000004          MOV     #IERR0,@#ERRVECT        ;AN ODD ADDRESS TRAP TO 4 MAY OCCUR.
1339  005534  012737  005542  001236          MOV     #I3,@#$TMP2
1340  005542  172401                  I3:     LDD     AC1,AC0                 ;TEST INSTRUCTION.
1341  005544  000240                  I4:     NOP
1342  005546  000240                  I5:     NOP
1343
1344  005550  012700  006264                  MOV     #IDAT00,R0
1345  005554  174010                          STD     AC0,(R0)                ;GET AC0, THE RESULTS.
1346
1347  005556  012700  006264                  MOV     #IDAT00,R0              ;SEE IF DATA IS CORRECT.
```

```
1348  005562  012701  006274          MOV     #IDATI0,R1
1349  005566  012702  000004          MOV     #4,R2
1350  005572  022021            I6:   CMP     (R0)+,(R1)+
1351  005574  001424                  BEQ     I105
1352
1353  005576  012700  006270          MOV     #IDATO2,R0              ;SEE IF (BUT FD) FAILED.
1354  005602  012702  000002          MOV     #2,R2
1355  005606  005720            I7:   TST     (R0)+
1356  005610  001413                  BEQ     I10
1357
1358  005612  012700  006270          MOV     #IDATO2,R0
1359  005616  012702  000002          MOV     #2,R2
1360  005622  022720  177777    1$:   CMP     #-1,(R0)+
1361  005626  001402                  BEQ     2$
1362  005630  000137  006126          JMP     @#IERR1
1363  005634  077206            2$:   SOB     R2,1$
1364  005636  000401                  BR      I106
1365  005640  077216            I10:  SOB     R2,I7
1366  005642  000137  006146    I106: JMP     @#IERR2
1367
1368  005646  077227            I105: SOB     R2,I6
1369
1370                            ;NOW TEST THE LOAD INSTRUCTION WITH FSRC MODE ZERO AND FD CLEAR.
1371
1372  005650                    I11:
1373  005650  104414                  LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
1374
1375  005652  012700  006244    I12:  MOV     #IPAT10,R0
1376  005656  012701  006274          MOV     #IDATI0,R1
1377  005662  012702  000004          MOV     #4,R2
1378  005666  012021            I13:  MOV     (R0)+,(R1)+
1379  005670  077202                  SOB     R2,I13
1380
1381  005672  012700  006274          MOV     #IDATI0,R0              ;SET UP AC1
1382  005676  172510                  LDD     (R0),AC1
1383
1384  005700  012700  006254          MOV     #IPAT20,R0              ;SET UP AC0
1385  005704  172410                  LDD     (R0),AC0
1386
1387  005706  012701  000001          MOV     #1,R1
1388  005712  012737  005722  001236  MOV     #I14,@#$TMP2
1389  005720  170001                  SETF                            ;CLEAR FD.
1390
1391  005722  172401            I14:  LDF     AC1,AC0                 ;TEST INSTRUCTION.
1392  005724  000240            I15:  NOP
1393  005726  000240            I16:  NOP
1394
1395  005730  170200                  STFPS   R0                      ;SEE IF FPS IS STILL CLEAR.
1396  005732  022700  000004          CMP     #4,R0
1397  005736  001402                  BEQ     I17
1398  005740  000137  006220          JMP     @#IERR3
1399
1400  005744                    I17:                                  ;RESET TO DOUBLE MODE.
1401  005744  170011                  SETD
1402
1403  005746  012700  006264          MOV     #IDATO0,R0
```

SEQ 0042

```
1404  005752  174010                          STD     AC0,(R0)                ;GET AC0
1405
1406  005754  012737  177777  006300          MOV     #-1,@#IDATI2
1407  005762  012737  177777  006302          MOV     #-1,@#IDATI3
1408  005770  012700  006264                  MOV     #IDAT00,R0
1409  005774  012701  006274                  MOV     #IDATI0,R1
1410  006000  012702  000004                  MOV     #4,R2
1411  006004  022021                  I20:    CMP     (R0)+,(R1)+             ;SEE IF AC0 WAS CORRECT.
1412  006006  001414                          BEQ     I23
1413
1414  006010  023737  006270  006250          CMP     @#IDAT02,@#IPAT12       ;DID (BUT FD) FAIL?
1415  006016  001402                          BEQ     I22
1416  006020  000137  006126          I21:    JMP     @#IERR1
1417  006024  023737  006272  006252  I22:    CMP     @#IDAT03,@#IPAT13
1418  006032  001372                          BNE     I21
1419  006034  000137  006174                  JMP     @#IERR4
1420
1421  006040  077217                  I23:    SOB     R2,I20
1422
1423  006042  000520                          BR      IDONE                   ;NO ERRORS.
1424
1425                                  ;IF AN ODD ADDRESS TRAP OCCURS COME HERE TO ANALYZE THE FSRC FAILURE.
1426  006044  022716  005544          IERR0:  CMP     #I4,(SP)                ;MAKE SURE THE TRAP OCCURRED
1427  006050  001413                          BEQ     1$                      ;ON THE INSTRUCTION BEING TESTED.
1428  006052  022716  005546                  CMP     #I5,(SP)
1429  006056  001410                          BEQ     1$
1430  006060  022716  005724                  CMP     #I15,(SP)
1431  006064  001405                          BEQ     1$
1432  006066  022716  005726                  CMP     #I16,(SP)
1433  006072  001402                          BEQ     1$
1434  006074  000137  062556                  JMP     @#CPSPUR
1435
1436  006100  011637  001236          1$:     MOV     (SP),@#$TMP2            ;REPORT FAILURE.
1437  006104  012737  000627  001240          MOV     #627,@#$TMP3
1438  006112  012737  000320  001242          MOV     #320,@#$TMP4
1439  006120  022626                          CMP     (SP)+,(SP)+
1440  006122  104001                  2$:     ERROR   1
1441  006124  000467                          BR      IDONE
1442
1443                                  ;REPORT DATA ERROR.
1444  006126                          IERR1:
1445  006126  012737  006274  001242          MOV     #IDATI0,@#$TMP4
1446  006134  012737  006264  001244          MOV     #IDAT00,@#$TMP5
1447  006142  104004                  1$:     ERROR   4
1448  006144  000457                          BR      IDONE
1449
1450                                  ;REPORT FAILURE OF (BUT FD)
1451  006146  012737  000153  001244  IERR2:  MOV     #153,@#$TMP5
1452  006154  012737  000434  001246          MOV     #434,@#$TMP6
1453  006162  012737  000435  001250          MOV     #435,@#$TMP7
1454  006170                          IERR25:
1455  006170  104001                  1$:     ERROR   1
1456  006172  000444                          BR      IDONE
1457  006174  012737  000153  001244  IERR4:  MOV     #153,@#$TMP5
1458  006202  012737  000435  001246          MOV     #435,@#$TMP6
1459  006210  012737  000434  001250          MOV     #434,@#$TMP7
```

```
1460  006216  000764                          BR      IERR25
1461
1462                                  ;REPORT INCORRECT FPS AFTER LOAD INSTRUCTION.
1463  006220                          IERR3:
1464  006220  012737  005722  001236          MOV     #I14,@#$TMP2
1465  006226  010037  001240                  MOV     R0,@#$TMP3
1466  006232  012737  000004  001242          MOV     #4,@#$TMP4
1467  006240  104001                  1$:     ERROR   1
1468  006242  000420                          BR      IDONE
1469
1470
1471  006244  000000                  IPAT10: 0
1472  006246  170360                  IPAT11: 170360
1473  006250  016161                  IPAT12: 016161
1474  006252  052525                  IPAT13: 052525
1475
1476  006254  177777                  IPAT20: -1
1477  006256  177777                  IPAT21: -1
1478  006260  177777                  IPAT22: -1
1479  006262  177777                  IPAT23: -1
1480
1481  006264  000000                  IDAT00: 0
1482  006266  000000                  IDAT01: 0
1483  006270  000000                  IDAT02: 0
1484  006272  000000                  IDAT03: 0
1485
1486  006274  000000                  IDATI0: 0
1487  006276  000000                  IDATI1: 0
1488  006300  000000                  IDATI2: 0
1489  006302  000000                  IDATI3: 0
1490
1491  006304                          IDONE:
1492  006304  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
1493                                                                  ;SEE IF THE USER HAS EXPRESSED
1494                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
1495                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
1496                                                                  ;THE USER TYPED CONTROL G?).
1497
1498
1499                                  ;;*****************************************************************
1500                                  ;*TEST 10       FDST MODE 0 TEST
1501                                  ;*
1502                                  ;*THIS IS A TEST OF THE STORE INSTRUCTIONS, STD AND STF,  WITH FDST MODE 0.
1503                                  ;*
1504                                  ;;*****************************************************************
1505  006306  000004                  TST10:  SCOPE
1506  006310                          T1:
1507  006310  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1508  006312  170011                          SETD                            ;SET FD
1509  006314  012700  007036                  MOV     #TPAT10,R0
1510  006320  012701  007066                  MOV     #TDATI0,R1
1511  006324  012702  000004                  MOV     #4,R2
1512  006330  012021                  T2:     MOV     (R0)+,(R1)+             ;SET UP THE INPUT DATA BUFFER.
1513  006332  077202                          SOB     R2,T2
1514
1515  006334  012700  007066                  MOV     #TDATI0,R0              ;LOAD AC0
```

```
1516  006340  172410                        LDD     (R0),AC0
1517
1518  006342  012700  007046                MOV     #TPAT20,R0              ;LOAD AC1
1519  006346  172510                        LDD     (R0),AC1
1520
1521  006350  012701  000001                MOV     #1,R1                   ;IF THE (BUT FDST) FORK FAILS
1522  006354  012737  006644  000004        MOV     #TERR0,@#ERRVECT        ;AN ODD ADDRESS TRAP COULD RESULT.
1523  006362  012737  006370  001236        MOV     #T3,@#$TMP2
1524  006370  174001                T3:     STD     AC0,AC1
1525  006372  000240                T4:     NOP
1526  006374  000240                T5:     NOP
1527
1528  006376  012700  007056                MOV     #TDAT00,R0
1529  006402  174110                        STD     AC1,(R0)                ;GET THE DATA.
1530
1531  006404  012703  007056                MOV     #TDAT00,R3              ;SEE IF THE DATA IS CORRECT.
1532  006410  012704  007066                MOV     #TDAT10,R4
1533  006414  012705  000004                MOV     #4,R5
1534  006420  022324                T6:     CMP     (R3)+,(R4)+
1535  006422  001413                        BEQ     T105
1536
1537  006424  012703  007062                MOV     #TDAT02,R3              ;DID (BUT FD) FAIL?
1538  006430  012705  000002                MOV     #2,R5
1539  006434  005723                T7:     TST     (R3)+
1540  006436  001402                        BEQ     T10
1541  006440  000137  006726                JMP     @#TERR1
1542  006444  077505                T10:    SOB     R5,T7
1543  006446  000137  006746                JMP     @#TERR2
1544
1545  006452  077516                T105:   SOB     R5,T6
1546
1547                                ;NOW TEST THE STF AC0,AC1 INSTRUCTION.
1548
1549  006454                        T11:
1550  006454  104414                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
1551
1552  006456  012700  007036        T12:    MOV     #TPAT10,R0              ;SET UP THE INPUT DATA BUFFER.
1553  006462  012701  007066                MOV     #TDAT10,R1
1554  006466  012702  000004                MOV     #4,R2
1555  006472  012021                T13:    MOV     (R0)+,(R1)+
1556  006474  077202                        SOB     R2,T13
1557
1558  006476  012700  007066                MOV     #TDAT10,R0              ;SET UP AC0
1559  006502  172410                        LDD     (R0),AC0
1560
1561  006504  012700  007046                MOV     #TPAT20,R0              ;SET UP AC1
1562  006510  172510                        LDD     (R0),AC1
1563
1564  006512  012701  000001                MOV     #1,R1
1565  006516  012737  006526  001236        MOV     #T14,@#$TMP2
1566  006524  170001                        SETF                            ;CLEAR FD
1567  006526  174001                T14:    STF     AC0,AC1
1568  006530  000240                T15:    NOP
1569  006532  000240                T16:    NOP
1570
1571  006534  005000                        CLR     R0
```

```
1572  006536  170200                        T10     FDST MODE 0 TEST        STFPS   R0                      ;SEE IF FPS IS CLEAR.
1573  006540  022700  000010                        CMP     #10,R0
1574  006544  001401                              BEQ     T17
1575  006546  000521                              BR      TERR3
1576
1577  006550                              T17:
1578  006550  170011                              SETD                            ;SET FD.
1579
1580  006552  012700  007056                      MOV     #TDAT00,R0
1581  006556  174110                              STD     AC1,(R0)                ;PICK UP AC1.
1582
1583
1584  006560  012737  177777  007072              MOV     #-1,@#TDATI2
1585  006566  012737  177777  007074              MOV     #-1,@#TDATI3
1586  006574  012703  007056                      MOV     #TDAT00,R3
1587  006600  012704  007066                      MOV     #TDATI0,R4
1588  006604  012705  000004                      MOV     #4,R5
1589  006610  022324                      T20:    CMP     (R3)+,(R4)+             ;WAS THE DATA TRANSFERRED CORRECTLY?
1590  006612  001412                              BEQ     T23
1591
1592  006614  023737  007062  007042              CMP     @#TDAT02,@#TPAT12       ;DID (BUT FD) FAIL.
1593  006622  001401                              BEQ     T22
1594  006624  000440                      T21:    BR      TERR1
1595  006626  023737  007064  007044      T22:    CMP     @#TDAT03,@#TPAT13
1596  006634  001373                              BNE     T21
1597  006636  000456                              BR      TERR4
1598
1599  006640  077515                      T23:    SOB     R5,T20
1600  006642  000515                              BR      TDONE
1601
1602
1603                                      ;TRAP HERE THROUGH VECTOR 4 IF AN ODD ADDRESS OCCURS.
1604  006644  022716  006372              TERR0:  CMP     #T4,(SP)                ;MAKE SURE THE TRAP WAS ON
1605  006650  001413                              BEQ     1$                      ;AN INSTRUCTION BEING TESTED.
1606  006652  022716  006374                      CMP     #T5,(SP)
1607  006656  001410                              BEQ     1$
1608  006660  022716  006530                      CMP     #T15,(SP)
1609  006664  001405                              BEQ     1$
1610  006666  022716  006532                      CMP     #T16,(SP)
1611  006672  001402                              BEQ     1$
1612  006674  000137  062556                      JMP     @#CPSPUR
1613
1614  006700  011637  001236              1$:     MOV     (SP),@#$TMP2
1615  006704  022626                              CMP     (SP)+,(SP)+
1616  006706  012737  000527  001240              MOV     #527,@#$TMP3
1617  006714  012737  000640  001242              MOV     #640,@#$TMP4
1618  006722  104001                      2$:     ERROR   1
1619  006724  000464                              BR      TDONE
1620
1621                                      ;REPORT DATA FAILURE.
1622  006726                              TERR1:
1623  006726  012737  007066  001242              MOV     #TDATI0,@#$TMP4
1624  006734  012737  007056  001244              MOV     #TDAT00,@#$TMP5
1625  006742  104001                      1$:     ERROR   1
1626  006744  000454                              BR      TDONE
1627
```

```
1628                                    ;REPORT FAILURE OF (BUT FD).
1629  006746  012737  000160  001246    TERR2:  MOV     #160,@#$TMP6
1630  006754  012737  000161  001250            MOV     #161,@#$TMP7
1631  006762  012737  000640  001244    TERR25: MOV     #640,@#$TMP5
1632  006770  104012                    1$:     ERROR   12
1633  006772  000441                            BR      TDONE
1634  006774  012737  000161  001246    TERR4:  MOV     #161,@#$TMP6
1635  007002  012737  000160  001250            MOV     #160,@#$TMP7
1636  007010  000764                            BR      TERR25
1637
1638                                    ;REPORT INCORRECT FPS AFTER STORE INSTRUCTION.
1639  007012                            TERR3:
1640  007012  012737  006530  001236            MOV     #T15,@#$TMP2
1641  007020  010037  001240                    MOV     R0,@#$TMP3
1642  007024  012737  000010  001242            MOV     #10,@#$TMP4
1643  007032  104001                    1$:     ERROR   1
1644  007034  000420                            BR      TDONE
1645
1646  007036  000000                    TPAT10: 0
1647  007040  170360                    TPAT11: 170360
1648  007042  016161                    TPAT12: 016161
1649  007044  052525                    TPAT13: 052525
1650
1651  007046  177777                    TPAT20: -1
1652  007050  177777                    TPAT21: -1
1653  007052  177777                    TPAT22: -1
1654  007054  177777                    TPAT23: -1
1655
1656  007056  000000                    TDAT00: 0
1657  007060  000000                    TDAT01: 0
1658  007062  000000                    TDAT02: 0
1659  007064  000000                    TDAT03: 0
1660
1661  007066  000000                    TDATI0: 0
1662  007070  000000                    TDATI1: 0
1663  007072  000000                    TDATI2: 0
1664  007074  000000                    TDATI3: 0
1665
1666  007076                            TDONE:
1667  007076  104413                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
1668                                                                    ;SEE IF THE USER HAS EXPRESSED
1669                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
1670                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
1671                                                                    ;THE USER TYPED CONTROL G?).
1672
1673
1674
1675                                    ;;*******************************************************************
1676                                    ;*TEST 11        ACCUMULATORS DATA PATTERNS TEST
1677                                    ;*
1678                                    ;*THIS IS A TEST OF THE FLOATING POINT PROCESSOR ACCUMULATORS.
1679                                    ;*EACH ACCUMULATOR IS TESTED IN TWO WAYS:
1680                                    ;*      1 TEST PATTERN GENERATED BY FLOATING A ONE ACROSS
1681                                    ;*        A FIELD OF ZEROES.
1682                                    ;*      2 TEST PATTERN GENERATED BY FLOATING A ZERO ACROSS
1683                                    ;*        A FIELD OF ONES.
```

```
1684                                    ;*EACH OF ACCUMULATORS AC0 THROUGH AC5 IS TESTED.
1685                                    ;;**************************************************************
1686  007100  000004                    TST11:  SCOPE
1687  007102  170011                            SETD                            ;SET FD.
1688                                    ;TEST ACCUMULATOR 0 WITH FLOATING ONE
1689  007104  012737  007134  001236            MOV     #G1,@#$TMP2
1690  007112  012700  011002                    MOV     #GPAT00,R0
1691  007116  012701  011042                    MOV     #GDAT00,R1
1692  007122  104414                            LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
1693  007124  004737  010650                    JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1694  007130  012703  000102                    MOV     #102,R3
1695  007134                            G1:
1696  007134  172410                            LDD     (R0),AC0
1697  007136  174000                            STD     AC0,AC0
1698  007140  172400                            LDD     AC0,AC0                 ;STORE THE TEST PATTERN.
1699  007142  174011                            STD     AC0,(R1)
1700  007144  004737  010746                    JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1701                                                                            ;THAT WHICH WAS WRITTEN.
1702  007150  005737  010776                    TST     @#GFLAG1
1703  007154  001004                            BNE     G2
1704  007156  005137  010776                    COM     @#GFLAG1
1705  007162  000261                            SEC
1706  007164  000401                            BR      G3
1707  007166  000241                    G2:     CLC
1708  007170  006160  000006            G3:     ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1709  007174  006160  000004                    ROL     4(R0)
1710  007200  006160  000002                    ROL     2(R0)
1711  007204  006110                            ROL     (R0)
1712  007206  004737  010726                    JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1713                                                                            ;BUFFER.
1714  007212  077330                            SOB     R3,G1
1715
1716                                    ;TEST ACCUMULATOR 0 WITH FLOATING ZERO
1717  007214  012737  007244  001236            MOV     #G4,@#$TMP2
1718  007222  012700  011012                    MOV     #GPAT10,R0
1719  007226  012701  011042                    MOV     #GDAT00,R1
1720  007232  104414                            LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
1721  007234  004737  010650                    JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1722  007240  012703  000102                    MOV     #102,R3
1723  007244                            G4:
1724  007244  172410                            LDD     (R0),AC0
1725  007246  174000                            STD     AC0,AC0
1726  007250  172400                            LDD     AC0,AC0                 ;STORE THE TEST PATTERN.
1727  007252  174011                            STD     AC0,(R1)
1728  007254  004737  010746                    JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1729                                                                            ;THAT WHICH WAS WRITTEN.
1730  007260  005737  010776                    TST     @#GFLAG1
1731  007264  001004                            BNE     G5
1732  007266  005137  010776                    COM     @#GFLAG1
1733  007272  000241                            CLC
1734  007274  000401                            BR      G6
1735  007276  000261                    G5:     SEC
1736  007300  006160  000006            G6:     ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1737  007304  006160  000004                    ROL     4(R0)
1738  007310  006160  000002                    ROL     2(R0)
1739  007314  006110                            ROL     (R0)
```

```
1740  007316  004737  010726          JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1741                                                                  ;BUFFER.
1742  007322  077330                  SOB     R3,G4
1743
1744                          ;TEST ACCUMULATOR 1 WITH FLOATING ONE
1745  007324  012737  007354  001236  MOV     #G7,@#$TMP2
1746  007332  012700  011002          MOV     #GPAT00,R0
1747  007336  012701  011042          MOV     #GDAT00,R1
1748  007342  104414                  LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
1749  007344  004737  010650          JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1750  007350  012703  000102          MOV     #102,R3
1751  007354                  G7:
1752  007354  172410                  LDD     (R0),AC0
1753  007356  174001                  STD     AC0,AC1
1754  007360  172401                  LDD     AC1,AC0                 ;STORE THE TEST PATTERN.
1755  007362  174011                  STD     AC0,(R1)
1756  007364  004737  010746          JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1757                                                                  ;THAT WHICH WAS WRITTEN.
1758  007370  005737  010776          TST     @#GFLAG1
1759  007374  001004                  BNE     G10
1760  007376  005137  010776          COM     @#GFLAG1
1761  007402  000261                  SEC
1762  007404  000401                  BR      G11
1763  007406  000241          G10:    CLC
1764  007410  006160  000006  G11:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1765  007414  006160  000004          ROL     4(R0)
1766  007420  006160  000002          ROL     2(R0)
1767  007424  006110                  ROL     (R0)
1768  007426  004737  010726          JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1769                                                                  ;BUFFER.
1770  007432  077330                  SOB     R3,G7
1771
1772
1773                          ;TEST ACCUMULATOR 1 WITH FLOATING ZERO
1774  007434  012737  007464  001236  MOV     #G12,@#$TMP2
1775  007442  012700  011012          MOV     #GPAT10,R0
1776  007446  012701  011042          MOV     #GDAT00,R1
1777  007452  104414                  LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
1778  007454  004737  010650          JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1779  007460  012703  000102          MOV     #102,R3
1780  007464                  G12:
1781  007464  172410                  LDD     (R0),AC0
1782  007466  174001                  STD     AC0,AC1
1783  007470  172401                  LDD     AC1,AC0                 ;STORE THE TEST PATTERN.
1784  007472  174011                  STD     AC0,(R1)
1785  007474  004737  010746          JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1786                                                                  ;THAT WHICH WAS WRITTEN.
1787  007500  005737  010776          TST     @#GFLAG1
1788  007504  001004                  BNE     G13
1789  007506  005137  010776          COM     @#GFLAG1
1790  007512  000241                  CLC
1791  007514  000401                  BR      G14
1792  007516  000261          G13:    SEC
1793  007520  006160  000006  G14:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1794  007524  006160  000004          ROL     4(R0)
1795  007530  006160  000002          ROL     2(R0)
```

```
1796  007534  006110                          ROL     (R0)
1797  007536  004737  010726                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1798                                                                          ;BUFFER.
1799  007542  077330                          SOB     R3,G12
1800
1801                                  ;TEST ACCUMULATOR 2 WITH FLOATING ONE
1802  007544  012737  007574  001236          MOV     #G15,@#$TMP2
1803  007552  012700  011002                  MOV     #GPAT00,R0
1804  007556  012701  011042                  MOV     #GDAT00,R1
1805  007562  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1806  007564  004737  010650                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1807  007570  012703  000102                  MOV     #102,R3
1808  007574                          G15:
1809  007574  172410                          LDD     (R0),AC0
1810  007576  174002                          STD     AC0,AC2
1811  007600  172402                          LDD     AC2,AC0         ;STORE THE TEST PATTERN.
1812  007602  174011                          STD     AC0,(R1)
1813  007604  004737  010746                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1814                                                                          ;THAT WHICH WAS WRITTEN.
1815  007610  005737  010776                  TST     @#GFLAG1
1816  007614  001004                          BNE     G16
1817  007616  005137  010776                  COM     @#GFLAG1
1818  007622  000261                          SEC
1819  007624  000401                          BR      G17
1820  007626  000241                  G16:    CLC
1821  007630  006160  000006          G17:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1822  007634  006160  000004                  ROL     4(R0)
1823  007640  006160  000002                  ROL     2(R0)
1824  007644  006110                          ROL     (R0)
1825  007646  004737  010726                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1826                                                                          ;BUFFER.
1827  007652  077330                          SOB     R3,G15
1828
1829                                  ;TEST ACCUMULATOR 2 WITH FLOATING ZERO
1830  007654  012737  007704  001236          MOV     #G20,@#$TMP2
1831  007662  012700  011012                  MOV     #GPAT10,R0
1832  007666  012701  011042                  MOV     #GDAT00,R1
1833  007672  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1834  007674  004737  010650                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1835  007700  012703  000102                  MOV     #102,R3
1836  007704                          G20:
1837  007704  172410                          LDD     (R0),AC0
1838  007706  174002                          STD     AC0,AC2
1839  007710  172402                          LDD     AC2,AC0         ;STORE THE TEST PATTERN.
1840  007712  174011                          STD     AC0,(R1)
1841  007714  004737  010746                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1842                                                                          ;THAT WHICH WAS WRITTEN.
1843  007720  005737  010776                  TST     @#GFLAG1
1844  007724  001004                          BNE     G21
1845  007726  005137  010776                  COM     @#GFLAG1
1846  007732  000241                          CLC
1847  007734  000401                          BR      G22
1848  007736  000261                  G21:    SEC
1849  007740  006160  000006          G22:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1850  007744  006160  000004                  ROL     4(R0)
1851  007750  006160  000002                  ROL     2(R0)
```

```
1852  007754  006110                          ROL     (R0)
1853  007756  004737  010726                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1854                                                                          ;BUFFER.
1855  007762  077330                          SOB     R3,G20
1856
1857                                  ;TEST ACCUMULATOR 3 WITH FLOATING ONE
1858  007764  012737  010014  001236          MOV     #G23,@#$TMP2
1859  007772  012700  011002                  MOV     #GPAT00,R0
1860  007776  012701  011042                  MOV     #GDAT00,R1
1861  010002  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1862  010004  004737  010650                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1863  010010  012703  000102                  MOV     #102,R3
1864  010014                          G23:
1865  010014  172410                          LDD     (R0),AC0
1866  010016  174003                          STD     AC0,AC3
1867  010020  172403                          LDD     AC3,AC0         ;STORE THE TEST PATTERN.
1868  010022  174011                          STD     AC0,(R1)
1869  010024  004737  010746                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1870                                                                          ;THAT WHICH WAS WRITTEN.
1871  010030  005737  010776                  TST     @#GFLAG1
1872  010034  001004                          BNE     G24
1873  010036  005137  010776                  COM     @#GFLAG1
1874  010042  000261                          SEC
1875  010044  000401                          BR      G25
1876  010046  000241                  G24:    CLC
1877  010050  006160  000006          G25:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1878  010054  006160  000004                  ROL     4(R0)
1879  010060  006160  000002                  ROL     2(R0)
1880  010064  006110                          ROL     (R0)
1881  010066  004737  010726                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1882                                                                          ;BUFFER.
1883  010072  077330                          SOB     R3,G23
1884
1885                                  ;TEST ACCUMULATOR 3 WITH FLOATING ZERO
1886  010074  012737  010124  001236          MOV     #G26,@#$TMP2
1887  010102  012700  011012                  MOV     #GPAT10,R0
1888  010106  012701  011042                  MOV     #GDAT00,R1
1889  010112  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1890  010114  004737  010650                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1891  010120  012703  000102                  MOV     #102,R3
1892  010124                          G26:
1893  010124  172410                          LDD     (R0),AC0
1894  010126  174003                          STD     AC0,AC3
1895  010130  172403                          LDD     AC3,AC0         ;STORE THE TEST PATTERN.
1896  010132  174011                          STD     AC0,(R1)
1897  010134  004737  010746                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1898                                                                          ;THAT WHICH WAS WRITTEN.
1899  010140  005737  010776                  TST     @#GFLAG1
1900  010144  001004                          BNE     G27
1901  010146  005137  010776                  COM     @#GFLAG1
1902  010152  000241                          CLC
1903  010154  000401                          BR      G30
1904  010156  000261                  G27:    SEC
1905  010160  006160  000006          G30:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1906  010164  006160  000004                  ROL     4(R0)
1907  010170  006160  000002                  ROL     2(R0)
```

```
1908  010174  006110                          ROL     (R0)
1909  010176  004737  010726                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1910                                                                          ;BUFFER.
1911  010202  077330                          SOB     R3,G26
1912
1913                                  ;TEST ACCUMULATOR 4 WITH FLOATING ONE
1914  010204  012737  010234  001236          MOV     #G31,@#$TMP2
1915  010212  012700  011002                  MOV     #GPAT00,R0
1916  010216  012701  011042                  MOV     #GDAT00,R1
1917  010222  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1918  010224  004737  010650                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1919  010230  012703  000102                  MOV     #102,R3
1920  010234                          G31:
1921  010234  172410                          LDD     (R0),AC0
1922  010236  174004                          STD     AC0,AC4
1923  010240  172404                          LDD     AC4,AC0         ;STORE THE TEST PATTERN.
1924  010242  174011                          STD     AC0,(R1)
1925  010244  004737  010746                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1926                                                                          ;THAT WHICH WAS WRITTEN.
1927  010250  005737  010776                  TST     @#GFLAG1
1928  010254  001004                          BNE     G32
1929  010256  005137  010776                  COM     @#GFLAG1
1930  010262  000261                          SEC
1931  010264  000401                          BR      G33
1932  010266  000241                  G32:    CLC
1933  010270  006160  000006          G33:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1934  010274  006160  000004                  ROL     4(R0)
1935  010300  006160  000002                  ROL     2(R0)
1936  010304  006110                          ROL     (R0)
1937  010306  004737  010726                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1938                                                                          ;BUFFER.
1939  010312  077330                          SOB     R3,G31
1940
1941                                  ;TEST ACCUMULATOR 4 WITH FLOATING ZERO
1942  010314  012737  010344  001236          MOV     #G34,@#$TMP2
1943  010322  012700  011012                  MOV     #GPAT10,R0
1944  010326  012701  011042                  MOV     #GDAT00,R1
1945  010332  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
1946  010334  004737  010650                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1947  010340  012703  000102                  MOV     #102,R3
1948  010344                          G34:
1949  010344  172410                          LDD     (R0),AC0
1950  010346  174004                          STD     AC0,AC4
1951  010350  172404                          LDD     AC4,AC0         ;STORE THE TEST PATTERN.
1952  010352  174011                          STD     AC0,(R1)
1953  010354  004737  010746                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1954                                                                          ;THAT WHICH WAS WRITTEN.
1955  010360  005737  010776                  TST     @#GFLAG1
1956  010364  001004                          BNE     G35
1957  010366  005137  010776                  COM     @#GFLAG1
1958  010372  000241                          CLC
1959  010374  000401                          BR      G36
1960  010376  000261                  G35:    SEC
1961  010400  006160  000006          G36:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1962  010404  006160  000004                  ROL     4(R0)
1963  010410  006160  000002                  ROL     2(R0)
```

```
1964  010414  006110                          ROL     (R0)
1965  010416  004737  010726                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1966                                                                          ;BUFFER.
1967  010422  077330                          SOB     R3,G34
1968
1969                                  ;TEST ACCUMULATOR 5 WITH FLOATING ONE
1970  010424  012737  010454  001236          MOV     #G37,@#$TMP2
1971  010432  012700  011002                  MOV     #GPAT00,R0
1972  010436  012701  011042                  MOV     #GDAT00,R1
1973  010442  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
1974  010444  004737  010650                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
1975  010450  012703  000102                  MOV     #102,R3
1976  010454                          G37:
1977  010454  172410                          LDD     (R0),AC0
1978  010456  174005                          STD     AC0,AC5
1979  010460  172405                          LDD     AC5,AC0                 ;STORF THE TEST PATTERN.
1980  010462  174011                          STD     AC0,(R1)
1981  010464  004737  010746                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
1982                                                                          ;THAT WHICH WAS WRITTEN.
1983  010470  005737  010776                  TST     @#GFLAG1
1984  010474  001004                          BNE     G40
1985  010476  005137  010776                  COM     @#GFLAG1
1986  010502  000261                          SEC
1987  010504  000401                          BR      G41
1988  010506  000241                  G40:    CLC
1989  010510  006160  000006          G41:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
1990  010514  006160  000004                  ROL     4(R0)
1991  010520  006160  000002                  ROL     2(R0)
1992  010524  006110                          ROL     (R0)
1993  010526  004737  010726                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
1994                                                                          ;BUFFER.
1995  010532  077330                          SOB     R3,G37
1996
1997                                  ;TEST ACCUMULATOR 5 WITH FLOATING ZERO
1998  010534  012737  010564  001236          MOV     #G42,@#$TMP2
1999  010542  012700  011012                  MOV     #GPAT10,R0
2000  010546  012701  011042                  MOV     #GDAT00,R1
2001  010552  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
2002  010554  004737  010650                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
2003  010560  012703  000102                  MOV     #102,R3
2004  010564                          G42:
2005  010564  172410                          LDD     (R0),AC0
2006  010566  174005                          STD     AC0,AC5
2007  010570  172405                          LDD     AC5,AC0                 ;STORE THE TEST PATTERN.
2008  010572  174011                          STD     AC0,(R1)
2009  010574  004737  010746                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
2010                                                                          ;THAT WHICH WAS WRITTEN.
2011  010600  005737  010776                  TST     @#GFLAG1
2012  010604  001004                          BNE     G43
2013  010606  005137  010776                  COM     @#GFLAG1
2014  010612  000241                          CLC
2015  010614  000401                          BR      G44
2016  010616  000261                  G43:    SEC
2017  010620  006160  000006          G44:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
2018  010624  006160  000004                  ROL     4(R0)
2019  010630  006160  000002                  ROL     2(R0)
```

```
2020  010634  006110                        ROL     (R0)
2021  010636  004737  010726                JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
2022                                                                          ;BUFFER.
2023  010642  077330                        SOB     R3,G42
2024
2025
2026  010644  000137  011052                JMP     @#GDONE
2027
2028                                  ;USE THIS ROUTINE TO INITIALIZE ALL THE DATA BUFFERS.
2029  010650  012705  010776          GSETUP: MOV     #GFLAG1,R5
2030  010654  012704  000026                  MOV     #26,R4
2031  010660  005025                  1$:     CLR     (R5)+
2032  010662  077402                          SOB     R4,1$
2033
2034  010664  012705  011012                MOV     #GPAT10,R5
2035  010670  012704  000010                MOV     #10,R4
2036  010674  005125                  2$:     COM     (R5)+
2037  010676  077402                          SOB     R4,2$
2038
2039  010700  020067  000076          GS1:    CMP     R0,GPAT00
2040  010704  001401                          BEQ     3$
2041  010706  000207                          RTS     PC
2042
2043  010710  012705  011042          3$:     MOV     #GDAT00,R5
2044  010714  012704  000004                MOV     #4,R4
2045  010720  005125                  4$:     COM     (R5)+
2046  010722  077402                          SOB     R4,4$
2047  010724  000207                          RTS     PC
2048
2049  010726  012705  011042          GRESET: MOV     #GDAT00,R5
2050  010732  012704  000004                MOV     #4,R4
2051  010736  005025                  1$:     CLR     (R5)+
2052  010740  077402                          SOB     R4,1$
2053  010742  000137  010700                JMP     @#GS1
2054
2055                                  ;SEE IF THE DATA WRITTEN MATCHES THE DATA READ.
2056  010746  011637  001244          GCMP:   MOV     (SP),@#$TMP5    ;STORE ERROR PC
2057  010752  012705  011042                MOV     #GDAT00,R5
2058  010756  012704  000004                MOV     #4,R4
2059  010762  010002                        MOV     R0,R2
2060  010764  022225                  1$:     CMP     (R2)+,(R5)+
2061  010766  001401                          BEQ     2$
2062  010770  104003                          ERROR   3               ;PROBABLY BAD MMU, OTHERWISE IT IS THE FP CHIP
2063  010772  077404                  2$:     SOB     R4,1$
2064  010774  000207                          RTS     PC
2065
2066
2067
2068
2069  010776  000000          GFLAG1: 0
2070  011000  000000          GFLAG2: 0
2071
2072  011002  000000          GPAT00: 0
2073  011004  000000          GPAT01: 0
2074  011006  000000          GPAT02: 0
2075  011010  000000          GPAT03: 0
```

```
2076
2077  011012  177777                GPAT10: -1
2078  011014  177777                GPAT11: -1
2079  011016  177777                GPAT12: -1
2080  011020  177777                GPAT13: -1
2081
2082  011022  177777                GAND0:  -1
2083  011024  177777                GAND1:  -1
2084  011026  177777                GAND2:  -1
2085  011030  177777                GAND3:  -1
2086
2087  011032  000000                GOR0:   0
2088  011034  000000                GOR1:   0
2089  011036  000000                GOR2:   0
2090  011040  000000                GOR3:   0
2091
2092  011042  000000                GDAT00: 0
2093  011044  000000                GDAT01: 0
2094  011046  000000                GDAT02: 0
2095  011050  000000                GDAT03: 0
2096
2097
2098  011052                        GDONE:
2099  011052  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
2100                                                                ;SEE IF THE USER HAS EXPRESSED
2101                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
2102                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2103                                                                ;THE USER TYPED CONTROL G?).
2104
2105
2106                                ;;******************************************************************
2107                                ;*TEST 12       FPP ACCUMULATORS DUAL ADDRESS TEST
2108                                ;*
2109                                ;*THIS TEST PERFORMS A DUAL ADDRESSING TEST ON THE FLOATING ACCUMULATORS.
2110                                ;*NOTE THAT ACCUMULATOR ZERO IS USED TO ACCESS ALL THE OTHERS.
2111                                ;*
2112                                ;;******************************************************************
2113  011054  000004                TST12:  SCOPE
2114  011056  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2115
2116  011060  005037  011604        H1:     CLR     @#HFLAG
2117  011064  012700  011606                MOV     #HA1W,R0                ;INITIALIZE THE LOAD BUFFER DATA.
2118  011070  012701  011726                MOV     #HDAT1,R1
2119  011074  012703  000024                MOV     #24,R3
2120  011100  012120                H2:     MOV     (R1)+,(R0)+
2121  011102  077302                        SOB     R3,H2
2122
2123  011104  004767  000422                JSR     PC,HCLR         ;CLEAR THE OUTPUT DATA BUFFER.
2124
2125  011110  170011                H3:     SETD
2126                                ;LOAD ACCUMULATOR 1
2127  011112  012700  011606                MOV     #HA1W,R0
2128  011116  172410                        LDD     (R0),AC0
2129  011120  174001                        STD     AC0,AC1
2130                                ;LOAD ACCUMULATOR 2
2131  011122  012700  011616                MOV     #HA2W,R0
```

```
2132  011126  172410                  LDD     (R0),AC0
2133  011130  174002                  STD     AC0,AC2
2134                                  ;LOAD ACCUMULATOR 3
2135  011132  012700  011626          MOV     #HA3W,R0
2136  011136  172410                  LDD     (R0),AC0
2137  011140  174003                  STD     AC0,AC3
2138                                  ;LOAD ACCUMULATOR 4
2139  011142  012700  011636          MOV     #HA4W,R0
2140  011146  172410                  LDD     (R0),AC0
2141  011150  174004                  STD     AC0,AC4
2142                                  ;LOAD ACCUMULATOR 5
2143  011152  012700  011646          MOV     #HA5W,R0
2144  011156  172410                  LDD     (R0),AC0
2145  011160  174005                  STD     AC0,AC5
2146
2147  011162  004737  011416  H4:     JSR     PC,@#HSTD               ;GO READ ALL ACCUMULATORS BACK.
2148
2149  011166  004737  011474          JSR     PC,@#HCMP               ;SEE IF DATA IS CORRECT.
2150
2151                                  ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 1,
2152                                  ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
2153                                  ;THE DATA.
2154  011172  012700  011606          MOV     #HA1W,R0
2155  011176  012702  000004          MOV     #4,R2
2156  011202  010001                  MOV     R0,R1
2157  011204  005121          H5:     COM     (R1)+
2158  011206  172410                  LDD     (R0),AC0
2159  011210  174001                  STD     AC0,AC1
2160  011212  004737  011416          JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
2161  011216  004737  011474          JSR     PC,@#HCMP               ;CHECK THE DATA.
2162  011222  077210                  SOB     R2,H5
2163
2164                                  ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 2,
2165                                  ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
2166                                  ;THE DATA.
2167  011224  012700  011616          MOV     #HA2W,R0
2168  011230  012702  000004          MOV     #4,R2
2169  011234  010001                  MOV     R0,R1
2170  011236  005121          H6:     COM     (R1)+
2171  011240  172410                  LDD     (R0),AC0
2172  011242  174002                  STD     AC0,AC2
2173  011244  004737  011416          JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
2174  011250  004737  011474          JSR     PC,@#HCMP               ;CHECK THE DATA.
2175  011254  077210                  SOB     R2,H6
2176
2177                                  ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 3,
2178                                  ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
2179                                  ;THE DATA.
2180  011256  012700  011626          MOV     #HA3W,R0
2181  011262  012702  000004          MOV     #4,R2
2182  011266  010001                  MOV     R0,R1
2183  011270  005121          H7:     COM     (R1)+
2184  011272  172410                  LDD     (R0),AC0
2185  011274  174003                  STD     AC0,AC3
2186  011276  004737  011416          JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
2187  011302  004737  011474          JSR     PC,@#HCMP               ;CHECK THE DATA.
```

```
 2188  011306  077210                    SOB     R2,H7
 2189
 2190                                    ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 4,
 2191                                    ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
 2192                                    ;THE DATA.
 2193  011310  012700  011636            MOV     #HA4W,R0
 2194  011314  012702  000004            MOV     #4,R2
 2195  011320  010001                    MOV     R0,R1
 2196  011322  005121            H10:    COM     (R1)+
 2197  011324  172410                    LDD     (R0),AC0
 2198  011326  174004                    STD     AC0,AC4
 2199  011330  004737  011416            JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
 2200  011334  004737  011474            JSR     PC,@#HCMP               ;CHECK THE DATA.
 2201  011340  077210                    SOB     R2,H10
 2202
 2203                                    ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 5,
 2204                                    ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
 2205                                    ;THE DATA.
 2206  011342  012700  011646            MOV     #HA5W,R0
 2207  011346  012702  000004            MOV     #4,R2
 2208  011352  010001                    MOV     R0,R1
 2209  011354  005121            H11:    COM     (R1)+
 2210  011356  172410                    LDD     (R0),AC0
 2211  011360  174005                    STD     AC0,AC5
 2212  011362  004737  011416            JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
 2213  011366  004737  011474            JSR     PC,@#HCMP               ;CHECK THE DATA.
 2214  011372  077210                    SOB     R2,H11
 2215
 2216
 2217  011374  005737  011604            TST     @#HFLAG
 2218  011400  001402                    BEQ     H12
 2219  011402  000137  011776            JMP     @#HDONE
 2220
 2221  011406  005137  011604    H12:    COM     @#HFLAG
 2222  011412  000137  011110            JMP     @#H3
 2223
 2224                                    ;STORE ALL ACCUMULATORS IN THE OUTPUT BUFFERS.
 2225  011416  004737  011532    HSTD:   JSR     PC,@#HCLR               ;CLEAR ALL OUTPUT BUFFERS.
 2226                                    ;STORE ACCUMULATOR 1
 2227  011422  012704  011656            MOV     #HA1R,R4
 2228  011426  172401                    LDD     AC1,AC0
 2229  011430  174014                    STD     AC0,(R4)
 2230                                    ;STORE ACCUMULATOR 2
 2231  011432  012704  011666            MOV     #HA2R,R4
 2232  011436  172402                    LDD     AC2,AC0
 2233  011440  174014                    STD     AC0,(R4)
 2234                                    ;STORE ACCUMULATOR 3
 2235  011442  012704  011676            MOV     #HA3R,R4
 2236  011446  172403                    LDD     AC3,AC0
 2237  011450  174014                    STD     AC0,(R4)
 2238                                    ;STORE ACCUMULATOR 4
 2239  011452  012704  011706            MOV     #HA4R,R4
 2240  011456  172404                    LDD     AC4,AC0
 2241  011460  174014                    STD     AC0,(R4)
 2242                                    ;STORE ACCUMULATOR 5
 2243  011462  012704  011716            MOV     #HA5R,R4
```

```
2244  011466  172405                        LDD     AC5,AC0
2245  011470  174014                        STD     AC0,(R4)
2246  011472  000207                        RTS     PC
2247
2248                                  ;COMPARE DATA LOADED WITH DATA READ.
2249  011474  012637  011602          HCMP:   MOV     (SP)+,@#HADR            ;SAVE RETURN ADDRESS.
2250  011500  012703  011606                  MOV     #HA1W,R3
2251  011504  012704  011656                  MOV     #HA1R,R4
2252  011510  012705  000024                  MOV     #24,R5
2253  011514  022324                  HCMP1:  CMP     (R3)+,(R4)+
2254  011516  001402                          BEQ     HCMP2
2255  011520  000137  011550                  JMP     @#HERROR
2256  011524  077505                  HCMP2:  SOB     R5,HCMP1
2257  011526  000177  000050                  JMP     @HADR
2258
2259                                  ;CLEAR THE DATA OUTPUT BUFFER.
2260  011532  012704  011656          HCLR:   MOV     #HA1R,R4
2261  011536  012705  000024                  MOV     #24,R5
2262  011542  005024                  HCLR1:  CLR     (R4)+
2263  011544  077502                          SOB     R5,HCLR1
2264  011546  000207                          RTS     PC
2265
2266                                  ;REPORT ERROR.
2267  011550                          HERROR:
2268  011550  012703  011606                  MOV     #HA1W,R3
2269  011554  012704  001236                  MOV     #$TMP2,R4
2270  011560  012705  000012                  MOV     #12,R5
2271  011564  010324                  1$:     MOV     R3,(R4)+
2272  011566  062703  000010                  ADD     #10,R3
2273  011572  077504                          SOB     R5,1$
2274  011574  104001                  2$:     ERROR   1
2275  011576  000137  011776                  JMP     @#HDONE
2276
2277
2278  011602  000000                  HADR:   0
2279  011604  000000                  HFLAG:  0
2280
2281  011606  000000  000000  000000  HA1W:   .WORD   0,0,0,0
2282  011614  000000
2283  011616  000000  000000  000000  HA2W:   .WORD   0,0,0,0
2284  011624  000000
2285  011626  000000  000000  000000  HA3W:   .WORD   0,0,0,0
2286  011634  000000
2287  011636  000000  000000  000000  HA4W:   .WORD   0,0,0,0
2288  011644  000000
2289  011646  000000  000000  000000  HA5W:   .WORD   0,0,0,0
2290  011654  000000
2291
2292  011656  000000  000000  000000  HA1R:   .WORD   0,0,0,0
2293  011664  000000
2294  011666  000000  000000  000000  HA2R:   .WORD   0,0,0,0
2295  011674  000000
2296  011676  000000  000000  000000  HA3R:   .WORD   0,0,0,0
2297  011704  000000
2298  011706  000000  000000  000000  HA4R:   .WORD   0,0,0,0
2299  011714  000000
```

```
2300  011716  000000  000000  000000  HA5R:   .WORD   0,0,0,0
2301  011724  000000
2302
2303  011726  073567  073567  073567  HDAT1:  .WORD   73567,73567,73567,73567
2304  011734  073567
2305  011736  063146  063146  063146  HDAT2:  .WORD   63146,63146,63146,63146
2306  011744  063146
2307  011746  010421  010421  010421  HDAT3:  .WORD   10421,10421,10421,10421
2308  011754  010421
2309  011756  031463  031463  031463  HDAT4:  .WORD   31463,31463,31463,31463
2310  011764  031463
2311  011766  042104  042104  042104  HDAT5:  .WORD   42104,42104,42104,42104
2312  011774  042104
2313
2314  011776                          HDONE:
2315  011776  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
2316                                                                  ;SEE IF THE USER HAS EXPRESSED
2317                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
2318                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2319                                                                  ;THE USER TYPED CONTROL G?).
2320
2321
2322                                  ;;*****************************************************************
2323                                  ;*TEST 13       FSRC MODE 0 WITH ILLEGAL ACCUMULATOR TEST
2324                                  ;*
2325                                  ;*THIS IS A TEST OF FSRC MODE 0 WITH ACCUMULATORS 6 AND 7. USE OF
2326                                  ;*EITHER OF THESE NON-EXISTENT ACCUMULATORS SHOULD RESULT IN A TRAP TO 244
2327                                  ;*WITH FEC=2 (ILLEGAL OPCODE TRAP)
2328                                  ;*
2329                                  ;;*****************************************************************
2330  012000  000004                  TST13:  SCOPE
2331  012002                          S1:
2332  012002  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2333  012004  170011                          SETD                            ;SET FD
2334  012006  012700  012512                  MOV     #SPAT10,R0              ;LOAD AC0
2335  012012  172410                          LDD     (R0),AC0
2336
2337  012014  012737  012212  000244          MOV     #SERR0,@#FPVECT         ;USE OF THE NON-EXISTENT AC-
2338                                                                  ;CUMULATOR SHOULD RESULT IN
2339                                                                  ;A TRAP TO 244.
2340  012022  012700  000001                  MOV     #1,R0                   ;A FAILURE IN THE FSRC FLOWS
2341                                                                  ;WILL RESULT IN AN ODD ADDRESS
2342  012026  012737  012422  000004          MOV     #SERR1,@#ERRVECT        ;TRAP TO 4.
2343  012034  005003                          CLR     R3
2344
2345  012036  172407                  S2:     LDD     AC7,AC0
2346  012040  170000                  S3:     CFCC
2347  012042  005203                          INC     R3
2348  012044  005203                  S4:     INC     R3
2349
2350  012046  012701  012522                  MOV     #SDAT00,R1              ;NO TRAP OCCURRED!!
2351  012052  174011                          STD     AC0,(R1)                ;SEE IF AC0 WAS MODIFIED.
2352
2353  012054  012701  012522                  MOV     #SDAT00,R1
2354  012060  012702  012512                  MOV     #SPAT10,R2
2355  012064  012703  000004                  MOV     #4,R3
```

```
2356  012070  022122                  S5:     CMP     (R1)+,(R2)+
2357  012072  001402                          BEQ     S6
2358  012074  000137  012352                  JMP     @#SERR2                 ;BAD DATA
2359  012100  077305                  S6:     SOB     R3,S5
2360
2361  012102  000137  012376                  JMP     @#SERR3                 ;MISSING TRAP
2362
2363                                  ;NOW TEST AC6.
2364  012106                          S7:
2365  012106  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2366  012110  170011                          SETD
2367
2368  012112  012700  012512                  MOV     #SPAT10,R0              ;LOAD AC0
2369  012116  172410                          LDD     (R0),AC0
2370
2371  012120  012737  012270  000244          MOV     #SERR4,@#FPVECT
2372  012126  012700  000001                  MOV     #1,R0
2373  012132  012737  012454  000004          MOV     #SERR5,@#ERRVECT
2374  012140  005003                          CLR     R3
2375
2376  012142  172406                  S8:     LDD     AC6,AC0
2377  012144  170000                  S9:     CFCC
2378  012146  005203                          INC     R3
2379  012150  005203                  S10:    INC     R3
2380
2381  012152  012701  012522                  MOV     #SDAT00,R1
2382  012156  174011                          STD     AC0,(R1)                ;NO TRAP! GET AC0.
2383
2384  012160  012701  012522                  MOV     #SDAT00,R1              ;WAS AC0 MODIFIED.
2385  012164  012702  012512                  MOV     #SPAT10,R2
2386  012170  012703  000004                  MOV     #4,R3
2387  012174  022122                  S11:    CMP     (R1)+,(R2)+
2388  012176  001402                          BEQ     S12
2389  012200  000137  012364                  JMP     @#SERR6
2390  012204  077305                  S12:    SOB     R3,S11
2391  012206  000137  012410                  JMP     @#SERR7
2392
2393                                  ;TRAPPED TO 244.
2394  012212  021627  012040          SERR0:  CMP     (SP),#S3                ;PC OF TRAP CORRECT?
2395  012216  001402                          BEQ     1$
2396  012220  000137  062524                  JMP     @#FPSPUR
2397
2398  012224  012737  012106  012506  1$:     MOV     #S7,@#SADR
2399
2400  012232  011637  001236          SERR10: MOV     (SP),@#$TMP2
2401  012236  022626                          CMP     (SP)+,(SP)+
2402  012240  005004                          CLR     R4
2403  012242  170204                          STFPS   R4                      ;IS FPS CORRECT?
2404  012244  022704  100200                  CMP     #100200,R4
2405  012250  001020                          BNE     SERR15
2406
2407  012252  005004                          CLR     R4
2408  012254  170304                          STST    R4                      ;IS FEC CORRECT?
2409  012256  022704  000002                  CMP     #2,R4
2410  012262  001023                          BNE     SERR20
2411  012264  000177  000216                  JMP     @SADR
```

```
2412
2413  012270  021627  012144          SERR4:  CMP     (SP),#S9
2414  012274  001402                          BEQ     1$
2415  012276  000137  062524                  JMP     @#FPSPUR
2416  012302  012737  012532  012506  1$:     MOV     #SDONE,@#SADR
2417  012310  000750                          BR      SERR10
2418
2419                                  ;REPORT FPS FAILURE:
2420  012312  012737  100200  001242  SERR15: MOV     #100200,@#$TMP4
2421  012320  010437  001240                  MOV     R4,@#$TMP3
2422  012324  104001                  1$:     ERROR   1
2423  012326  000177  000154                  JMP     @SADR
2424
2425                                  ;REPORT FEC BAD:
2426  012332  012737  000002  001242  SERR20: MOV     #2,@#$TMP4
2427  012340  010437  001240                  MOV     R4,@#$TMP3
2428  012344  104001                  1$:     ERROR   1
2429  012346  000177  000134                  JMP     @SADR
2430
2431
2432                                  ;AC0 WAS MODIFIED. (BUT FSRC) FORK FAILED.
2433  012352  012737  012036  001236  SERR2:  MOV     #S2,@#$TMP2
2434  012360  104001                  1$:     ERROR   1
2435  012362  000463                          BR      SDONE
2436  012364  012737  012142  001236  SERR6:  MOV     #S8,@#$TMP2
2437  012372  104001                  1$:     ERROR   1
2438  012374  000456                          BR      SDONE
2439
2440  012376  012737  012036  001236  SERR3:  MOV     #S2,@#$TMP2
2441  012404  104001                  1$:     ERROR   1
2442  012406  000451                          BR      SDONE
2443  012410  012737  012142  001236  SERR7:  MOV     #S8,@#$TMP2
2444  012416  104001                  1$:     ERROR   1
2445  012420  000444                          BR      SDONE
2446
2447                                  ;FAILURE OF (BUT FSRC) CAUSED AN ODD ADDRESS TRAP TO 4.
2448  012422  021627  012040          SERR1:  CMP     (SP),#S3                ;DID TRAP OCCUR ON TESTED INSTRUCTION?
2449  012426  001405                          BEQ     1$
2450  012430  021627  012044                  CMP     (SP),#S4
2451  012434  001402                          BEQ     1$
2452  012436  000137  062556                  JMP     @#CPSPUR
2453
2454  012442  011637  001236          1$:     MOV     (SP),@#$TMP2
2455  012446  022626                          CMP     (SP)+,(SP)+
2456  012450  104001                  2$:     ERROR   1
2457  012452  000427                          BR      SDONE
2458
2459  012454  021627  012142          SERR5:  CMP     (SP),#S8                ;DID TRAP OCCUR ON TEST INSTRUCTION?
2460  012460  001405                          BEQ     1$
2461  012462  021627  012144                  CMP     (SP),#S9
2462  012466  001402                          BEQ     1$
2463  012470  000137  062556                  JMP     @#CPSPUR
2464
2465  012474  011637  001236          1$:     MOV     (SP),@#$TMP2
2466  012500  022626                          CMP     (SP)+,(SP)+
2467  012502  104001                  2$:     ERROR   1
```

```
2468  012504  000412                          BR      SDONE
2469
2470  012506  000000                  SADR:   0
2471  012510  177777                          -1
2472  012512  010421                  SPAT10: 10421
2473  012514  021042                  SPAT11: 21042
2474  012516  031463                  SPAT12: 31463
2475  012520  042104                  SPAT13: 42104
2476
2477  012522  000000                  SDAT00: 0
2478  012524  000000                  SDAT01: 0
2479  012526  000000                  SDAT02: 0
2480  012530  000000                  SDAT03: 0
2481
2482  012532                          SDONE:
2483  012532  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
2484                                                                  ;SEE IF THE USER HAS EXPRESSED
2485                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
2486                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2487                                                                  ;THE USER TYPED CONTROL G?).
2488
2489                                  ;;*****************************************************************
2490                                  ;*TEST 14       FSRC MODE 2 TEST
2491                                  ;*
2492                                  ;* THIS IS A TEST OF FSRC MODE 2, AUTO
2493                                  ;* INCREMENT MODE.
2494                                  ;*
2495                                  ;;*****************************************************************
2496  012534  000004                  TST14:  SCOPE
2497  012536  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
2498
2499  012540                          J1:
2500  012540  170011                          SETD                    ;SET DOUBLE MODE
2501
2502  012542  012700  013016                  MOV     #JDAT0,R0
2503  012546  172410                          LDD     (R0),AC0        ;LOAD AC0=ALL 1
2504
2505  012550  012700  012776                  MOV     #JDAT1,R0
2506  012554  005003                          CLR     R3
2507  012556  012737  012646  000004         MOV     #JERR0,@#ERRVECT
2508
2509  012564  172420                  J2:     LDD     (R0)+,AC0       ;TEST INSTRUCTION
2510  012566  005203                  J3:     INC     R3
2511  012570  005203                  J4:     INC     R3
2512
2513  012572  012701  013006                  MOV     #JDAT00,R1
2514  012576  174011                          STD     AC0,(R1)        ;PICK UP RESULTS
2515
2516  012600  020027  012766                  CMP     R0,#JBUF0       ;WAS AN AUTO
2517  012604  001001                          BNE     1$              ;DECREMENT EXECUTED?
2518  012606  000442                          BR      JERR1
2519
2520  012610  012702  012776          1$:     MOV     #JDAT1,R2       ;IS DATA CORRECT?
2521  012614  012703  013006                  MOV     #JDAT00,R3
2522  012620  012704  000004                  MOV     #4,R4
2523  012624  022223                  J5:     CMP     (R2)+,(R3)+
```

```
 2524  012626  001401                        BEQ     J6
 2525  012630  000443                        BR      JERR2
 2526  012632  077404                J6:     SOB     R4,J5
 2527
 2528  012634  022700  013006                CMP     #JDATI0+10,R0   ;WAS R0 INCREM.
 2529  012640  001401                        BEQ     J7              ;BY 10 (OCTAL)
 2530  012642  000424                        BR      JERR1
 2531
 2532  012644  000470                J7:     BR      JDONE
 2533
 2534                                  ;IF A TRAP THROUGH 4 OCCURS COME HERE
 2535
 2536  012646  021627  012566        JERR0:  CMP     (SP),#J3        ;SEE IF THE TRAP
 2537  012652  001405                        BEQ     J10             ;OCCURRED ON THE
 2538  012654  021627  012570                CMP     (SP),#J4        ;TESTED INSTRUCTION
 2539  012660  001402                        BEQ     J10
 2540  012662  000137  062556                JMP     @#CPSPUR
 2541
 2542  012666  012737  000762  001240  J10:  MOV     #762,@#$TMP3    ;REPORT FSRC FLOW
 2543  012674  012737  000322  001242        MOV     #322,@#$TMP4    ;FAILURE
 2544  012702  011637  001236                MOV     (SP),@#$TMP2
 2545  012706  022626                        CMP     (SP)+,(SP)+
 2546  012710  104001                1$:     ERROR   1
 2547  012712  000445                        BR      JDONE
 2548
 2549  012714                        JERR1:                          ;REPORT, R0 NOT
 2550  012714  012737  012564  001236        MOV     #J2,@#$TMP2     ;CORRECTLY AFFECTED
 2551  012722  010037  001240                MOV     R0,@#$TMP3
 2552  012726  012737  013006  001242        MOV     #JDATI0+10,@#$TMP4
 2553  012734  104001                1$:     ERROR   1
 2554  012736  000433                        BR      JDONE
 2555
 2556                                  ;REPORT DATA FAILURE
 2557
 2558  012740                        JERR2:
 2559  012740  012737  012564  001236        MOV     #J2,@#$TMP2
 2560  012746  012737  012776  001240        MOV     #JDATI0,@#$TMP3
 2561  012754  012737  013006  001242        MOV     #JDAT00,@#$TMP4
 2562  012762  104001                1$:     ERROR   1
 2563  012764  000420                        BR      JDONE
 2564
 2565  012766  010421                JBUF0:  .WORD   010421
 2566  012770  021042                JBUF1:          021042
 2567  012772  042104                JBUF2:          042104
 2568  012774  031463                JBUF3:          031463
 2569
 2570  012776  052525                JDATI0:         052525
 2571  013000  114631                JDATI1:         114631
 2572  013002  063146                JDATI2:         063146
 2573  013004  073567                JDATI3:         073567
 2574
 2575  013006  000000                JDAT00:         0
 2576  013010  000000                JDAT01:         0
 2577  013012  000000                JDAT02:         0
 2578  013014  000000                JDAT03:         0
 2579
```

```
 2580  013016  177777                JDAT0:          -1
 2581  013020  177777                JDAT1:          -1
 2582  013022  177777                JDAT2:          -1
 2583  013024  177777                JDAT3:          -1
 2584
 2585
 2586  013026                        JDONE:
 2587  013026  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 2588                                                                ;SEE IF THE USER HAS EXPRESSED
 2589                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
 2590                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 2591                                                                ;THE USER TYPED CONTROL G?).
 2592
 2593
 2594                                ;;******************************************************************
 2595                                ;*TEST 15       FSRC MODE 4 TEST
 2596                                ;*
 2597                                ;* THIS IS A TEST OF FSRC MODE 4, AUTO
 2598                                ;* DECREMENT MODE.
 2599                                ;*
 2600                                ;;******************************************************************
 2601  013030  000004                TST15:  SCOPE
 2602  013032  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 2603
 2604  013034                        K1:
 2605  013034  170011                        SETD                    ;SET DOUBLE MODE
 2606
 2607  013036  012700  013310                MOV     #KPAT0,R0
 2608  013042  172410                        LDD     (R0),AC0        ;LOAD A DEFAULT
 2609                                                                ;PATTERN INTO AC0
 2610  013044  012700  013270                MOV     #KBUF0,R0
 2611  013050  005003                        CLR     R3
 2612  013052  012737  013142  000004       MOV     #KERR0,@#ERRVECT
 2613
 2614  013060  172440                K2:     LDD     -(R0),AC0       ;TEST INSTRUCTION
 2615  013062  005203                K3:     INC     R3
 2616  013064  005203                K4:     INC     R3
 2617
 2618  013066  012701  013300                MOV     #KDAT00,R1
 2619  013072  174011                        STD     AC0,(R1)        ;PICK UP THE RESULT
 2620
 2621  013074  020027  013300                CMP     R0,#KBUF0+10    ;WAS AN AUTO
 2622  013100  001001                        BNE     1$              ;INCREMENT EXECUTED
 2623  013102  000441                        BR      KERR1
 2624
 2625  013104  012702  013260        1$:     MOV     #KDAT10,R2      ;IS DATA CORRECT?
 2626  013110  012703  013300                MOV     #KDAT00,R3
 2627  013114  012704  000004                MOV     #4,R4
 2628  013120  022223                K5:     CMP     (R2)+,(R3)+
 2629  013122  001401                        BEQ     K6
 2630  013124  000442                        BR      KERR2
 2631  013126  077404                K6:     SOB     R4,K5
 2632
 2633  013130  022700  013260                CMP     #KBUF0-10,R0    ;WAS R0 DECREMENTED
 2634  013134  001401                        BEQ     K7              ;PROPERLY?
 2635  013136  000423                        BR      KERR1
```

```
2636
2637  013140  000467                    K7:     BR      KDONE
2638
2639                                    ;TRAP TO HERE ON AN ODD ADDRESS ERROR
2640
2641  013142  021627  013062            KERR0:  CMP     (SP),#K3        ;SEE IF THE ERROR
2642  013146  001405                            BEQ     K10             ;OCCURRED AT THE
2643  013150  021627  013064                    CMP     (SP),#K4        ;INSTRUCTION TESTED.
2644  013154  001402                            BEQ     K10
2645  013156  000137  062556                    JMP     @#CPSPUR
2646
2647  013162  012737  000762  001240  K10:     MOV     #762,@#$TMP3    ;REPORT FAILURE IN
2648  013170  012737  000324  001242            MOV     #324,@#$TMP4    ;FSRC FLOWS
2649  013176  011637  001236                    MOV     (SP),@#$TMP2
2650  013202  104001                    1$:     ERROR   1
2651  013204  000445                            BR      KDONE
2652
2653  013206                            KERR1:                          ;REPORT, R0
2654  013206  012737  013060  001236            MOV     #K2,@#$TMP2     ;INCORRECTLY AFFECTED.
2655  013214  010037  001240                    MOV     R0,@#$TMP3
2656  013220  012737  013260  001242            MOV     #KDATI0,@#$TMP4
2657  013226  104001                    1$:     ERROR   1
2658  013230  000433                            BR      KDONE
2659
2660                                    ;REPORT DATA FAILURE
2661
2662  013232                            KERR2:
2663  013232  012737  013060  001236            MOV     #K2,@#$TMP2
2664  013240  012737  013260  001240            MOV     #KDATI0,@#$TMP3
2665  013246  012737  013300  001242            MOV     #KDAT00,@#$TMP4
2666  013254  104001                    1$:     ERROR   1
2667  013256  000420                            BR      KDONE
2668
2669  013260  052525                    KDATI0: .WORD   052525
2670  013262  114631                    KDATI1:         114631
2671  013264  063140                    KDATI2:         063140
2672  013266  073567                    KDATI3:         073567
2673
2674  013270  010421                    KBUF0:          010421
2675  013272  031463                    KBUF1:          031463
2676  013274  042104                    KBUF2:          042104
2677  013276  021042                    KBUF3:          021042
2678
2679  013300  000000                    KDAT00:         0
2680  013302  000000                    KDAT01:         0
2681  013304  000000                    KDAT02:         0
2682  013306  000000                    KDAT03:         0
2683
2684  013310  177777                    KPAT0:          -1
2685  013312  177777                    KPAT1:          -1
2686  013314  177777                    KPAT2:          -1
2687  013316  177777                    DPAT3:          -1
2688
2689  013320                            KDONE:
2690  013320  104413                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
2691                                                                    ;SEE IF THE USER HAS EXPRESSED
```

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


                        ;;*****************************************************************
                        ;*TEST 16       FSRC MODE 2, WITH FD=0, TEST
                        ;*
                        ;* THIS IS A TEST OF FSRC MODE 2 WITH
                        ;* FD=0.  (AUTO INCREMENT)
                        ;*
                        ;;*****************************************************************
013322  000004          TST16:  SCOPE
013324  104414                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.

013326                  L1:
013326  170011                  SETD                    ;SET DOUBLE MODE

013330  012700  013576          MOV     #LPAT10,R0
013334  172410                  LDD     (R0),AC0        ;LOAD AC0

013336  012700  013620          MOV     #LDAT10,R0      ;SET UP THE INPUT
013342  012701  013606          MOV     #LPAT20,R1      ;DATA
013346  012702  000004          MOV     #4,R2

013352  012120          1$:     MOV     (R1)+,(R0)+
013354  077202                  SOB     R2,1$

013356  012700  013620          MOV     #LDAT10,R0
013362  005003                  CLR     R3
013364  170001                  SETF                    ;CLEAR FD.

013366  172420          L2:     LDF     (R0)+,AC0
013370  005203          L3:     INC     R3

013372                  L4:
013372  170011                  SETD                    ;SET FD

013374  012701  013632          MOV     #LDAT00,R1
013400  174011                  STD     AC0,(R1)        ;PICK UP RESULTS

013402  020027  013624          CMP     R0,#LDAT12      ;WAS R0 INCREMENTED
013406  001401                  BEQ     1$              ;CORRECTLY BY 4
013410  000421                  BR      LERR1

013412  012737  177777  013624  1$:     MOV     #-1,@#LDAT12
013420  012737  177777  013626          MOV     #-1,@#LDAT13
013426  012702  013620                  MOV     #LDAT10,R2      ;IS DATA CORRECT
013432  012703  013632                  MOV     #LDAT00,R3
013436  012704  000004                  MOV     #4,R4

013442  022223          L5:     CMP     (R2)+,(R3)+
013444  001401                  BEQ     L6
013446  000427                  BR      LERR2
013450  077404          L6:     SOB     R4,L5
```

```
 2748  013452  000473                          BR      LDONE
 2749
 2750  013454                          LERR1:                          ;REPORT FAILURE
 2751  013454  012737  013366  001236          MOV     #L2,@#$TMP2     ;R0 NOT INCREMENTED
 2752  013462  010037  001240                  MOV     R0,@#$TMP3      ;BY 4
 2753  013466  012737  013624  001242          MOV     #LDATI2,@#$TMP4
 2754  013474  104001                  1$:     ERROR   1
 2755  013476  000461                          BR      LDONE
 2756
 2757  013500                          LERR3:                          ;REPORT DATA FAILURE.
 2758  013500  012737  013366  001236          MOV     #L2,@#$TMP2
 2759  013506  012737  013620  001240          MOV     #LDATI0,@#$TMP3
 2760  013514  012737  013632  001242          MOV     #LDAT00,@#$TMP4
 2761  013522  104001                  1$:     ERROR   1
 2762  013524  000446                          BR      LDONE
 2763
 2764  013526  012702  013606          LERP2:  MOV     #LPAT20,R2      ;DID (BUT FD)
 2765  013532  012703  013632                  MOV     #LDAT00,R3      ;FAIL.
 2766  013536  012704  000004                  MOV     #4,R4
 2767  013542  022223                  1$:     CMP     (R2)+,(R3)+
 2768  013544  001355                          BNE     LERR3
 2769  013546  077403                          SOB     R4,1$
 2770  013550  012737  013366  001236          MOV     #L2,@#$TMP2
 2771  013556  012737  013620  001240          MOV     #LDATI0,@#$TMP3
 2772  013564  012737  013634  001242          MOV     #LDAT01,@#$TMP4
```

```
 2773  013572  104001                  2$:     ERROR   1
 2774  013574  000422                          BR      LDONE
 2775
 2776  013576  177777                  LPAT10: .WORD   -1
 2777  013600  177777                  LPAT11:         -1
 2778  013602  177777                  LPAT12:         -1
 2779  013604  177777                  LPAT13:         -1
 2780
 2781  013606  052525                  LPAT20:         052525
 2782  013610  114631                  LPAT21:         114631
 2783  013612  063142                  LPAT22:         063142
 2784  013614  073567                  LPAT23:         073567
 2785  013616  000001                          .WORD   000001
 2786  013620  000000                  LDATI0:         0
 2787  013622  000000                  LDATI1:         0
 2788  013624  000000                  LDATI2:         0
 2789  013626  000000                  LDATI3:         0
 2790  013630  000001                          .WORD   00001
 2791  013632  000000                  LDAT00:         0
 2792  013634  000000                  LDAT01:         0
 2793  013636  000000                  LDAT02:         0
 2794  013640  000000                  LDAT03:         0
 2795
 2796  013642                          LDONE:
 2797  013642  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 2798                                                                  ;SEE IF THE USER HAS EXPRESSED
 2799                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 2800                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 2801                                                                  ;THE USER TYPED CONTROL G?).
 2802
 2803
 2804                                  ;;*****************************************************************
 2805                                  ;*TEST 17       FSRC MODE 2 WITH GR7, IMMEDIATE MODE, TEST
 2806                                  ;*
 2807                                  ;* THIS IS A TEST OF FSRC MODE 2
 2808                                  ;* USING GR7 (THE PC).  THIS IS IMMEDIATE
 2809                                  ;* MODE.
 2810                                  ;*
 2811                                  ;;*****************************************************************
 2812  013644  000004                  TST17:  SCOPE
 2813
 2814  013646                          M1:
 2815  013646  170011                          SETD
 2816
 2817  013650  012700  014142                  MOV     #MPAT10,R0
 2818  013654  172410                          LDD     (R0),AC0        ;LOAD BACKROUND
 2819                                                                  ;PATTERN INTO AC0.
 2820  013656  005004                          CLR     R4
 2821  013660  012737  014102  000004         MOV     #MERR3,@#ERRVECT
 2822
 2823  013666  172427  000000          M15:    LDD     #0,AC0          ;TEST INSTRUCTION
 2824          013670                          .=.-2           ;EFFECTIVELY: 05204 IS PUT IN THE FIRST
 2825  013670  005204                          .WORD   5204    ;16 BIT WORD, OR THE 'EXP-FRACTION' WORD.
 2826  013672  005204                  M2:     INC     R4              ;NOTE THAT
 2827  013674  005204                  M3:     INC     R4              ;005204=INC R4
 2828  013676  005204                  M4:     INC     R4
```

```
2829
2830  013700  020427  000003          CMP     R4,#3           ;SEE IF THE PC
2831  013704  001401                  BEQ     1$              ;WAS INCREMENTED
2832  013706  000443                  BR      MERR0           ;BY 2 DURING THE
2833                                                          ;INSTRUCTION.  IF
2834                                                          ;NOT THEN A BAD
2835                                                          ;CONSTANT WAS GENERATED
2836  013710  012700  014162      1$: MOV     #MDAT00,R0
2837  013714  174010                  STD     AC0,(R0)        ;GET THE DATA
2838
2839  013716  012700  014162          MOV     #MDAT00,R0
2840  013722  022720  005204          CMP     #5204,(R0)+     ;IS THE DATA CORRECT?
2841  013726  001401                  BEQ     M5
2842  013730  000451                  BR      MERR1
2843  013732  012701  000003      M5: MOV     #3,R1
2844  013736  005720              M6: TST     (R0)+
2845  013740  001002                  BNE     M7
2846  013742  077103                  SOB     R1,M6
2847  013744  000512                  BR      MDONE
2848
2849  013746  012700  014162      M7: MOV     #MDAT00,R0      ;DID (BUT GRM) FAIL?
2850  013752  012701  000004          MOV     #4,R1
2851  013756  022720  005204      M8: CMP     #5204,(R0)+
2852  013762  001401                  BEQ     M9
2853  013764  000433                  BR      MERR1
2854  013766  077105              M9: SOB     R1,M8
2855
2856  013770                      MERR2:                      ;REPORT FAILURE
2857  013770  012737  013666  001236        MOV     #M15,@#$TMP2    ;OF (BUT GR7)
2858  013776  012737  014152  001240        MOV     #MPAT20,@#$TMP3
2859  014004  012737  014162  001242        MOV     #MDAT00,@#$TMP4
2860  014012  104001              1$: ERROR   1
2861  014014  000466                  BR      MDONE
2862
2863  014016  012705  013672      MERR0: MOV     #M2,R5          ;REPORT FAILURE
2864  014022  010537  001242          MOV     R5,@#$TMP4      ;PC INCREMENTED
2865  014026  162704  000003          SUB     #3,R4
2866  014032  006304                  ASL     R4
2867  014034  160405                  SUB     R4,R5
2868  014036  010537  001240          MOV     R5,@#$TMP3
2869  014042  012737  013666  001236        MOV     #M15,@#$TMP2
2870  014050  104001              1$: ERROR   1
2871  014052  000447                  BR      MDONE
2872
2873  014054                      MERR1:                      ;REPORT DATA
2874  014054  012737  013666  001236        MOV     #M15,@#$TMP2    ;FAILURE
2875  014062  012737  014162  001240        MOV     #MDAT00,@#$TMP3
2876  014070  012737  014152  001242        MOV     #MPAT20,@#$TMP4
2877  014076  104001              1$: ERROR   1
2878  014100  000434                  BR      MDONE
2879                              ;TRAP TO HERE THROUGH 4.
2880  014102  032716  000001      MERR3: BIT     #1,(SP)         ;SEE IF THE
2881  014106  001002                  BNE     1$              ;TRAP TO 4 OCCURRED
2882  014110  000137  062556          JMP     @#CPSPUR        ;BECAUSE OF AN
2883                                                          ;ODD ADDRESS
2884  014114  011637  001240      1$: MOV     (SP),@#$TMP3    ;IF YES REPORT
```

```
2885  014120  012737  013672  001242          MOV     #M2,@#$TMP4     ;BAD CONSTANT
2886  014126  012737  013666  001236          MOV     #M15,@#$TMP2    ;GENERATED
2887  014134  022626                          CMP     (SP)+,(SP)+
2888  014136  104001                  2$:     ERROR   1
2889  014140  000414                          BR      MDONE
2890
2891  014142  177777                  MPAT10: -1
2892  014144  177777                  MPAT11: -1
2893  014146  177777                  MPAT12: -1
2894  014150  177777                  MPAT13: -1
2895
2896  014152  005204                  MPAT20: 5204
2897  014154  005204                  MPAT21: 5204
2898  014156  005204                  MPAT22: 5204
2899  014160  005204                  MPAT23: 5204
2900
2901  014162  000000                  MDAT00: 0
2902  014164  000000                  MDAT01: 0
2903  014166  000000                  MDAT02: 0
2904  014170  000000                  MDAT03: 0
2905
2906  014172                          MDONE:
2907  014172  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
2908                                                                  ;SEE IF THE USER HAS EXPRESSED
2909                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
2910                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2911                                                                  ;THE USER TYPED CONTROL G?).
2912
2913
2914                                  ;;*****************************************************************
2915                                  ;*TEST 20       FSRC MODE 3 TEST
2916                                  ;*
2917                                  ;* THIS IS A TEST OF FSRC MODE 3, AUTO INCREMENT
2918                                  ;* DEFERRED
2919                                  ;*
2920                                  ;;*****************************************************************
2921  014174  000004                  TST20:  SCOPE
2922
2923  014176                          N1:
2924  014176  170011                          SETD                    ;SET FD MODE
2925
2926  014200  012700  014660                  MOV     #NPAT10,R0
2927  014204  172410                          LDD     (R0),AC0        ;LOAD AC0 WITH A DEFAULT
2928                                                                  ;PATTERN
2929  014206  012700  014646                  MOV     #NPAT20,R0
2930  014212  005003                          CLR     R3
2931  014214  012737  014370  000004          MOV     #NERR0,@#ERRVECT        ;IF A FAILURE OCCURS
2932                                                                  ;IN THE FSRC FLOWS AN
2933                                                                  ;ODD TRAP TO 4 COULD OCCUR
2934  014222  172430                  N2:     LDD     @(R0)+,AC0      ;TEST INSTRUCTION.
2935  014224  005203                  N3:     INC     R3
2936  014226  005203                  N4:     INC     R3
2937
2938  014230  012701  014626                  MOV     #NDAT00,R1
2939  014234  174011                          STD     AC0,(R1)        ;GET THE DATA
2940
```

```
2941  014236  020027  014650          CMP     R0,#NPAT20+2    ;WAS R0 INCREMENTED
2942  014242  001437                  BEQ     N12             ;BY 2?
2943
2944  014244  020027  014656  N5:     CMP     R0,#NPAT20+10   ;FSRC MODE 2?
2945  014250  001001                  BNE     N6
2946  014252  000506                  BR      NERR1
2947
2948  014254  020027  014636  N6:     CMP     R0,#NPAT20-10   ;FSRC MODE 4?
2949  014260  001001                  BNE     N7
2950  014262  000520                  BR      NERR2
2951
2952  014264  020027  014646  N7:     CMP     R0,#NPAT20
2953  014270  001023                  BNE     N11
2954
2955  014272  012702  014626          MOV     #NDAT00,R2      ;FSRC MODE 0?
2956  014276  012703  000004          MOV     #4,R3
2957  014302  022227  177777  N8:     CMP     (R2)+,#-1
2958  014306  001002                  BNE     N9
2959  014310  077304                  SOB     R3,N8
2960  014312  000510                  BR      NERR3
2961
2962  014314  012702  014626  N9:     MOV     #NDAT00,R2      ;FSRC MODE 1
2963  014320  012703  014646          MOV     #NPAT20,R3
2964  014324  012704  000004          MOV     #4,R4
2965  014330  022223          N10:    CMP     (R2)+,(R3)+
2966  014332  001002                  BNE     N11
2967  014334  077403                  SOB     R4,N10
2968  014336  000502                  BR      NERR4
2969
2970  014340  000505          N11:    BR      NERR5
2971
2972  014342  012702  014626  N12:    MOV     #NDAT00,R2      ;DATA CORRECT?
2973  014346  012703  014670          MOV     #NDAT10,R3
2974  014352  012704  000004          MOV     #4,R4
2975  014356  022223          N13:    CMP     (R2)+,(R3)+
2976  014360  001002                  BNE     N14
2977  014362  077403                  SOB     R4,N13
2978  014364  000545                  BR      NDONE
2979
2980  014366  000504          N14:    BR      NERR6
2981
2982                          ;IF AN ODD ADDRESS TRAP OCCURS COME HERE
2983                          ;TO SEE IF THE FAILURE WAS IN THE FSRC
2984                          ;FLOWS
2985
2986  014370  022716  014226  NERR0:  CMP     #N4,(SP)        ;FSRC MODE 6 OR 7?
2987  014374  001412                  BEQ     NERR10
2988  014376  022716  014224          CMP     #N3,(SP)
2989  014402  001402                  BEQ     1$
2990  014404  000137  062556          JMP     @#CPSPUR
2991  014410  020027  014644  1$:     CMP     R0,#NPAT20-2    ;FSRC MODE 5?
2992  014414  001407                  BEQ     NERR11
2993  014416  000137  062556          JMP     @#CPSPUR
2994
2995  014422          NERR10:                                 ;WENT TO FSRC
2996  014422  011637  001236          MOV     (SP),@#$TMP2    ;MODE 6 OR 7.
```

```
2997  014426  022626                          CMP     (SP)+,(SP)+
2998  014430  104001                  1$:     ERROR   1
2999  014432  000522                          BR      NDONE
3000
3001  014434  011637  001236          NERR11: MOV     (SP),@#$TMP2    ;WENT TO FSRC
3002  014440  022626                          CMP     (SP)+,(SP)+     ;MODE 5.
3003  014442  012737  000627  001244          MOV     #627,@#$TMP5
3004  014450  012737  000323  001250          MOV     #323,@#$TMP7
3005  014456  012737  000325  001246          MOV     #325,@#$TMP6
3006  014464  104001                  1$:     ERROR   1
3007  014466  000504                          BR      NDONE
3008  014470  012737  000322  001246  NERR1:  MOV     #322,@#$TMP6    ;FSRC MODE 2.
3009  014476  012737  000627  001244  NERR20: MOV     #627,@#$TMP5
3010  014504  012737  000323  001250          MOV     #323,@#$TMP7
3011  014512  012737  014222  001236          MOV     #N2,@#$TMP2
3012  014520  104001                  1$:     ERROR   1
3013  014522  000466                          BR      NDONE
3014  014524  012737  000324  001246  NERR2:  MOV     #324,@#$TMP6    ;FSRC MODE 4
3015  014532  000761                          BR      NERR20
3016  014534  012737  000320  001246  NERR3:  MOV     #320,@#$TMP6    ;FSRC MODE 0
3017  014542  000755                          BR      NERR20
3018  014544  012737  000321  001246  NERR4:  MOV     #321,@#$TMP6    ;FSRC MODE 1
3019  014552  000751                          BR      NERR20
3020
3021  014554  010037  001240          NERR5:  MOV     R0,@#$TMP3      ;R0 NOT
3022  014560  012737  014650  001242          MOV     #NPAT20+2,@#$TMP4       ;INCREMENTED
3023  014566  012737  014222  001236          MOV     #N2,@#$TMP2     ;PROPERLY.
3024  014574  104001                  1$:     ERROR   1
3025  014576  000440                          BR      NDONE
3026
3027  014600                          NERR6:                          ;DATA FAILURE.
3028  014600  012737  014222  001236          MOV     #N2,@#$TMP2
3029  014606  012737  014626  001240          MOV     #NDAT00,@#$TMP3
3030  014614  012737  014670  001242          MOV     #NDATI0,@#$TMP4
3031  014622  104001                  1$:     ERROR   1
3032  014624  000425                          BR      NDONE
3033
3034  014626  000000                  NDAT00: .WORD   0
3035  014630  000000                  NDAT01:         0
3036  014632  000000                  NDAT02:         0
3037  014634  000000                  NDAT03:         0
3038
3039  014636  052525  052525  052525          .WORD   52525,52525,52525,52525
3040  014644  052525
3041  014646  014670                  NPAT20: .WORD   NDATI0
3042  014650  070707                  NPAT21:         070707
3043  014652  070707                  NPAT22:         070707
3044  014654  070707                  NPAT23:         070707
3045  014656  000001                          .WORD   1
3046  014660  177777                  NPAT10: .WORD   -1
3047  014662  177777                  NPAT11:         -1
3048  014664  177777                  NPAT12:         -1
3049  014666  177777                  NPAT13:         -1
3050
3051  014670  010421                  NDATI0: .WORD   010421
3052  014672  021042                  NDATI1:         021042
```

```
3053  014674  031463                  NDAT12:         031463
3054  014676  042104                  NDAT13:         042104
3055
3056  014700                          NDONE:
3057  014700  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
3058                                                                  ;SEE IF THE USER HAS EXPRESSED
3059                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
3060                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3061                                                                  ;THE USER TYPED CONTROL G?).
3062
3063
3064                                  ;;*****************************************************************
3065                                  ;*TEST 21        FSRC MODE 5 TEST
3066                                  ;*
3067                                  ;* THIS IS A TEST OF FSRC MODE 5, AUTO DECREMENT
3068                                  ;* DEFERRED.
3069                                  ;*
3070
3071                                  ;;*****************************************************************
3072  014702  000004                  TST21:  SCOPE
3073
3074  014704                          01:
3075  014704  170011                          SETD                    ;SET FD MODE
3076
3077  014706  012700  015364                  MOV     #OPAT10,R0
3078  014712  172410                          LDD     (R0),AC0        ;LOAD AC0 WITH A
3079                                                                  ;DEFAULT PATTERN.
3080  014714  012700  015352                  MOV     #OPAT21,R0
3081  014720  005003                          CLR     R3
3082  014722  012737  015074  000004         MOV     #OERR0,@#ERRVEC ;IF A FAILURE
3083                                                                  ;OCCURS IN THE FSRC
3084                                                                  ;FLOWS AN ODD ADDR.
3085                                                                  ;TRAP TO 4 MAY OCCUR.
3086  014730  172450                  02:     LDD     @-(R0),AC0      ;TEST INSTRUCTION
3087  014732  005203                  03:     INC     R3
3088  014734  005203                  04:     INC     R3
3089
3090  014736  012701  015332                  MOV     #ODAT00,R1
3091  014742  174011                          STD     AC0,(R1)        ;GET THE DATA
3092
3093  014744  020027  015350                  CMP     R0,#OPAT20      ;WAS R0 DECREMENTED
3094  014750  001436                          BEQ     012             ;BY 2?
3095
3096  014752  020027  015362          05:     CMP     R0,#OPAT21+10   ;FSRC MODE 2
3097  014756  001001                          BNE     06
3098  014760  000505                          BR      OERR1
3099
3100  014762  020027  015342          06:     CMP     R0,#OPAT21-10   ;FSRC MODE 4?
3101  014766  001001                          BNE     07
3102  014770  000517                          BR      OERR2
3103
3104  014772  020027  015352          07:     CMP     R0,#OPAT21
3105
3106  014776  012702  015334                  MOV     #ODAT01,R2      ;FSRC MODE 0?
3107  015002  012703  000004                  MOV     #4,R3
3108  015006  022227  177777          08:     CMP     (R2)+,#-1
```

```
3109  015012  001002                        BNE     09
3110  015014  077304                        SOB     R3,08
3111  015016  000510                        BR      OERR3
3112
3113  015020  012702  015332        09:     MOV     #ODAT00,R2      ;FSRC MODE 1?
3114  015024  012703  015352                MOV     #OPAT21,R3
3115  015030  012704  000004                MOV     #4,R4
3116  015034  022223                010:    CMP     (R2)+,(R3)+
3117  015036  001002                        BNE     011
3118  015040  077403                        SOB     R4,010
3119  015042  000502                        BR      OERR4
3120
3121  015044  000505                011:    BR      OERR5
3122
3123  015046  012702  015332        012:    MOV     #ODAT00,R2      ;DATA CORRECT?
3124  015052  012703  015374                MOV     #ODATI0,R3
3125  015056  012704  000004                MOV     #4,R4
3126  015062  022223                013:    CMP     (R2)+,(R3)+
3127  015064  001002                        BNE     014
3128  015066  077403                        SOB     R4,013
3129  015070  000545                        BR      ODONE
3130
3131  015072  000504                014:    BR      OERR6
3132
3133                                ;IF AN ODD ADDRESS TRAP OCCURS COME
3134                                ;HERE TO SEE IF THE FAILURE WAS IN THE
3135                                ;FSRC FLOWS:
3136
3137  015074  022716  014734        OERR0:  CMP     #04,(SP)        ;FSRC MODE 6 OR 7?
3138  015100  001412                        BEQ     OERR10
3139  015102  022716  014732                CMP     #03,(SP)
3140  015106  001402                        BEQ     1$
3141  015110  000137  062556                JMP     @#CPSPUR
3142  015114  020027  015354        1$:     CMP     R0,#OPAT21+2    ;FSRC MODE 3?
3143  015120  001425                        BEQ     OERR1
3144  015122  000137  062556                JMP     @#CPSPUR
3145
3146  015126                        OERR10:                         ;WENT TO FSRC
3147  015126  011637  001236                MOV     (SP),@#$TMP2    ;MODE 6 OR 7
3148  015132  022626                        CMP     (SP)+,(SP)+
3149  015134  104001                1$:     ERROR   1
3150  015136  000522                        BR      ODONE
3151
3152  015140  011637  001240        OERR11: MOV     (SP),@#$TMP3    ;WENT TO FSRC MODE
3153  015144  022626                        CMP     (SP)+,(SP)+     ;3
3154  015146  012737  000627  001244        MOV     #627,@#$TMP5
3155  015154  012737  000325  001250        MOV     #325,@#$TMP7
3156  015162  012737  000323  001246        MOV     #323,@#$TMP6
3157  015170  104001                1$:     ERROR   1
3158  015172  000504                        BR      ODONE
3159
3160  015174  012737  000322  001246OERR1:  MOV     #322,@#$TMP6    ;FSRC MODE2
3161  015202  012737  000627  001242OERR20: MOV     #627,@#$TMP4
3162  015210  012737  000325  001250        MOV     #325,@#$TMP7
3163  015216  012737  014730  001236        MOV     #02,@#$TMP2
3164  015224  104001                1$:     ERROR   1
```

```
3165  015226  000466                          BR      ODONE
3166  015230  012737  000324  001246  OERR2:  MOV     #324,@#$TMP6    ;FSRC MODE 4
3167  015236  000761                          BR      OERR20
3168  015240  012737  000320  001246  OERR3:  MOV     #320,@#$TMP6    ;FSRC MODE 0
3169  015246  000755                          BR      OERR20
3170  015250  012737  000321  001246  OERR4:  MOV     #321,@#$TMP6    ;FSRC MODE 1
3171  015256  000751                          BR      OERR20
3172
3173
3174  015260  010037  001240          OERR5:  MOV     R0,@#$TMP3      ;R0 NOT DECREMENTED
3175  015264  012737  015350  001242          MOV     #OPAT20,@#$TMP4 ;PROPER_Y
3176  015272  012737  014734  001236          MOV     #04,@#$TMP2
3177  015300  104001                  1$:     ERROR   1
3178  015302  000440                          BR      ODONE
3179
3180  015304                          OERR6:                          ;DATA FAILURE
3181  015304  012737  014730  001236          MOV     #02,@#$TMP2
3182  015312  012737  015332  001240          MOV     #ODAT00,@#$TMP3
3183  015320  012737  015374  001242          MOV     #ODAT10,@#$TMP4
3184  015326  104001                  1$:     ERROR   1
3185  015330  000425                          BR      ODONE
3186
3187  015332  000000                  ODAT00: .WORD   0
3188  015334  000000                  ODAT01:         0
3189  015336  000000                  ODAT02:         0
3190  015340  000000                  ODAT03:         0
3191
3192  015342  052525  052525  052525          .WORD   52525,52525,52525
3193  015350  015374                  OPAT20: .WORD   ODAT10
3194  015352  070707                  OPAT21: 070707
3195  015354  070707                  OPAT22: 070707
3196  015356  070707                  OPAT23: 070707
3197  015360  070707                  OPAT24: 070707
3198  015362  000001                          .WORD   1
3199  015364  177777                  OPAT10: .WORD   -1
3200  015366  177777                  OPAT11:         -1
3201  015370  177777                  OPAT12:         -1
3202  015372  177777                  OPAT13:         -1
3203
3204  015374  073567                  ODAT10: .WORD   73567
3205  015376  004210                  ODAT11:         004210
3206  015400  114631                  ODAT12:         114631
3207  015402  125252                  ODAT13:         125252
3208
3209  015404                          ODONE:
3210  015404  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
3211                                                                  ;SEE IF THE USER HAS EXPRESSED
3212                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
3213                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3214                                                                  ;THE USER TYPED CONTROL G?).
3215
3216
3217                                  ;;*****************************************************************
3218                                  ;*TEST 22        FSRC MODE 6 TEST
3219                                  ;*
3220                                  ;* THIS IS A TEST OF FSRC MODE 6, INDEX MODE
```

```
3221                                    ;*
3222                                    ;;*****************************************************************
3223  015406  000004                    TST22:  SCOPE
3224
3225  015410                            P1:
3226  015410  170011                            SETD                    ;SET FD MODE
3227
3228  015412  012700  016030                    MOV     #PPAT10,R0
3229  015416  172410                            LDD     (R0),AC0        ;LOAD A DEFAJLT PATTERN
3230                                                                    ;INTO AC0
3231  015420  012737  015526  000004            MOV     #PERR0,@#ERRVECT        ;IF THE (BUT FSRC) FORQ
3232                                                                    ;FAILS AN ODD ADDRESS TRAP
3233  015426  012700  015577                    MOV     #PDAT10-241,R0  ;COULD OCCUR.
3234
3235  015432  172460  000241           P2:     LDD     241(R0),AC0
3236          015434                    P3=P2+2
3237
3238  015436  012701  016050           P4:     MOV     #PDAT00,R1
3239  015442  174011                            STD     AC0,(R1)        ;GET THE DATA
3240  015444  012703  000004                    MOV     #4,R3
3241  015450  012702  016040                    MOV     #PDAT10,R2
3242  015454  012701  016050                    MOV     #PDAT00,R1
3243  015460  022221                    P5:     CMP     (R2)+,(R1)+     ;CHECK THE DATA
3244  015462  001007                            BNE     P6
3245  015464  077303                            SOB     R3,P5
3246  015466  022700  015577                    CMP     #PDAT10-241,R0  ;R0 CORRECT?
3247  015472  001401                            BEQ     1$
3248  015474  000512                            BR      PERR21
3249  015476  000137  016060           1$:     JMP     @#PDONE
3250
3251  015502  012701  016050           P6:     MOV     #PDAT00,R1
3252  015506  012703  000004                    MOV     #4,R3
3253  015512  022721  177777           P7:     CMP     #-1,(R1)+       ;WAS IT FSRC MODE 0?
3254  015516  001401                            BEQ     P8
3255  015520  000512                            BR      PERR1
3256  015522  077305                    P8:     SOB     R3,P7
3257  015524  000523                            BR      PERR2
3258                                    ;TRAP TO HERE ON AN ODD ADDRESS
3259  015526  021627  015434           PERR0:  CMP     (SP),#P3
3260  015532  001411                            BEQ     PERR11
3261  015534  021627  015436                    CMP     (SP),#P4        ;WAS IT FSRC MODE 7?
3262  015540  001402                            BEQ     PERR10
3263  015542  000137  062556                    JMP     @#CPSPUR
3264
3265  015546  012737  000327  001246  PERR10: MOV     #327,@#$TMP6
3266  015554  000443                            BR      PERR17
3267  015556  022700  015577           PERR11: CMP     #PDAT10-241,R0  ;WAS IT FSRC MODE 1
3268  015562  001004                            BNE     PERR12
3269  015564  012737  000321  001246            MOV     #321,@#$TMP6
3270  015572  000434                            BR      PERR17
3271  015574  022700  015607           PERR12: CMP     #PDAT10-241+10,R0       ;WAS IT FSRC MODE 2
3272  015600  001004                            BNE     PERR13
3273  015602  012737  000322  001246            MOV     #322,@#$TMP6
3274  015610  000425                            BR      PERR17
3275  015612  022700  015601           PERR13: CMP     #PDAT10-241+2,R0        ;WAS IT FSRC MODE 3
3276  015616  001004                            BNE     PERR14
```

```
3277  015620  012737  000323  001246          MOV     #323,a#$TMP6
3278  015626  000416                          BR      PERR17
3279  015630  022700  015567          PERR14: CMP     #PDATI0-241-10,R0       ;WAS IT FSRC MODE 4
3280  015634  001004                          BNE     PERR15
3281  015636  012737  000324  001246          MOV     #324,a#$TMP6
3282  015644  000407                          BR      PERR17
3283  015646  022700  015575          PERR15: CMP     #PDATI0-241-2,R0        ;WAS IT FSRC MODE 5
3284  015652  001401                          BEQ     PERR16
3285  015654  000416                          BR      PERR20
3286  015656  012737  000325  001246  PERR16: MOV     #325,a#$TMP6
3287
3288  015664  012737  000627  001244  PERR17: MOV     #627,a#$TMP5    ;REPORT FSRC
3289  015672  012737  000326  001250          MOV     #326,a#$TMP7    ;FLOWS FAILURE.
3290  015700  011637  001236                  MOV     (SP),a#$TMP2
3291  015704  022626                          CMP     (SP)+,(SP)+
3292  015706  104001                  1$:     ERROR   1
3293  015710  000463                          BR      PDONE
3294
3295  015712  011637  001236          PERR20: MOV     (SP),a#$TMP2    ;REPORT R0 AFFECTED
3296  015716  022626                          CMP     (SP)+,(SP)+
3297  015720  000403                          BR      PERR22
3298  015722  012737  015432  001236  PERR21: MOV     #P2,a#$TMP2
3299  015730                          PERR22:
3300  015730  010037  001240                  MOV     R0,a#$TMP3
3301  015734  012737  015577  001242          MOV     #PDATI0-241,a#$TMP4
3302  015742  104001                  1$:     ERROR   1
3303  015744  000445                          BR      PDONE
3304
3305  015746                          PERR1:                          ;DATA FAILURE.
3306  015746  012737  015432  001236          MOV     #P2,a#$TMP2
3307  015754  012737  016040  001240          MOV     #PDATI0,a#$TMP3
3308  015762  012737  016050  001242          MOV     #PDAT00,a#$TMP4
3309  015770  104001                  1$:     ERROR   1
3310  015772  000432                          BR      PDONE
3311
3312  015774                          PERR2:                          ;FSRC FAILURE TO
3313  015774  012737  015432  001236          MOV     #P2,a#$TMP2     ;MODE 0
3314  016002  012737  000627  001244          MOV     #627,a#$TMP5
3315  016010  012737  000326  001250          MOV     #326,a#$TMP7
3316  016016  012737  000320  001246          MOV     #320,a#$TMP6
3317  016024  104001                  1$:     ERROR   1
3318  016026  000414                          BR      PDONE
3319
3320  016030  177777                  PPAT10: .WORD   -1
3321  016032  177777                  PPAT11:         -1
3322  016034  177777                  PPAT12:         -1
3323  016036  177777                  PPAT13:         -1
3324
3325  016040  010421                  PDATI0: .WORD   010421
3326  016042  031463                  PDATI1:         031463
3327  016044  052525                  PDATI2:         052525
3328  016046  073567                  PDATI3:         073567
3329
3330  016050  000000                  PDAT00: .WORD   0
3331  016052  000000                  PDAT01:         0
3332  016054  000000                  PDAT02:         0
```

```
3333  016056  000000                    PDAT03:         0
3334
3335  016060                            PDONE:
3336  016060  104413                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
3337                                                                    ;SEE IF THE USER HAS EXPRESSED
3338                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
3339                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3340                                                                    ;THE USER TYPED CONTROL G?).
3341
3342
3343                                    ;;*****************************************************************
3344                                    ;*TEST 23       FSRC MODE 7 TEST
3345                                    ;*
3346                                    ;* THIS IS A TEST OF FSRC MODE 7, INDEX
3347                                    ;* DEFERRED MODE.
3348                                    ;*
3349                                    ;;*****************************************************************
3350  016062  000004                    TST23:  SCOPE
3351
3352  016064                            Q1:
3353  016064  170011                            SETD
3354
3355  016066  012700  016520                    MOV     #QPAT10,R0
3356  016072  172410                            LDD     (R0),AC0        ;LOAD A DEFAULT
3357                                                                    ;PATTERN INTO AC0
3358  016074  012737  016226  000004            MOV     #QERR0,@#ERRVECT        ;IF THE (BUT FSRC)
3359                                                                    ;FORK FAILS AN
3360                                                                    ;ODD ADR TRAP COULD
3361                                                                    ;OCCUR
3362  016102  012700  016267                    MOV     #QPAT20-241,R0
3363
3364  016106  172470  000241            Q2:     LDD     @241(R0),AC0
3365          016110                    Q3=Q2+2
3366
3367  016112  012701  016540            Q4:     MOV     #QDAT00,R1
3368  016116  174011                            STD     AC0,(R1)        ;GET THE DATA
3369
3370  016120  012703  000004                    MOV     #4,R3
3371  016124  012704  016540                    MOV     #QDAT00,R4
3372  016130  012705  016550                    MOV     #QDAT10,R5
3373  016134  022425                    Q5:     CMP     (R4)+,(R5)+     ;CHECK THE DATA
3374  016136  001007                            BNE     Q6
3375  016140  077303                            SOB     R3,Q5
3376
3377  016142  022700  016267                    CMP     #QPAT20-241,R0  ;CHECK R0.
3378  016146  001401                            BEQ     1$
3379  016150  000514                            BR      QERR21
3380  016152  000137  016560            1$:     JMP     @#QDONE
3381
3382  016156  012701  016540            Q6:     MOV     #QDAT00,R1
3383  016162  012703  000004                    MOV     #4,R3
3384  016166  022721  177777            Q7:     CMP     #-1,(R1)+       ;WAS IT FSRC MODE 0?
3385  016172  001002                            BNE     Q8
3386  016174  077304                            SOB     R3,Q7
3387  016176  000513                            BR      QERR2
3388
```

```
3389  016200  012701  016530          Q8:     MOV     #QPAT20,R1
3390  016204  012702  016540                  MOV     #QDAT00,R2
3391  016210  012703  000004                  MOV     #4,R3
3392  016214  022122                  Q9:     CMP     (R1)+,(R2)+     ;WAS IT FSRC 6
3393  016216  001401                          BEQ     Q10             ;OR DATA FAILURE
3394  016220  000524                          BR      QERR1
3395  016222  077304                  Q10:    SOB     R3,Q9
3396  016224  000504                          BR      QERR3
3397
3398                                  ;TRAP TO HERE ON AN ODD ADR FAILURE
3399
3400  016226  021627  015434          QERR0:  CMP     (SP),#P3
3401  016232  000137  062556                  JMP     @#CPSPUR
3402
3403  016236  022700  016267          QERR11: CMP     #QPAT20-241,R0  ;WAS IT FSRC
3404  016242  001004                          BNE     QERR12          ;MODE 1?
3405  016244  012737  000321  001246          MOV     #321,@#$TMP6
3406  016252  000434                          BR      QERR17
3407  016254  022700  016277          QERR12: CMP     #QPAT20-241+10,R0       ;WAS IT FSRC
3408  016260  001004                          BNE     QERR13          ;MODE 2?
3409  016262  012737  000322  001246          MOV     #322,@#$TMP6
3410  016270  000425                          BR      QERR17
3411  016272  022700  016271          QERR13: CMP     #QPAT20-241+2,R0        ;WAS I' FSRC
3412  016276  001004                          BNE     QERR14          ;MODE 3?
3413  016300  012737  000323  001246          MOV     #323,@#$TMP6
3414  016306  000416                          BR      QERR17
3415  016310  022700  016257          QERR14  CMP     #QPAT20-241-10,R0       ;WAS IT FSRC
3416  016314  001004                          BNE     QERR15          ;MODE 4
3417  016316  012737  000324  001246          MOV     #324,@#$TMP6
3418  016324  000407                          BR      QERR17
3419
3420  016326  022700  016265          QERR15: CMP     #QPAT20-241-2,R0        ;WAS IT FSRC
3421  016332  001401                          BEQ     QERR16          ;MODE 5
3422  016334  000416                          BR      QERR20
3423
3424  016336  012737  000325  001246  QERR16: MOV     #325,@#$TMP6
3425
3426  016344  012737  000627  001244  QERR17  MOV     #627,@#$TMP5    ;REPORT FSRC FAILURE
3427  016352  012737  000327  001250          MOV     #327,@#$TMP7
3428  016360  011637  001236                  MOV     (SP),@#$TMP2
3429  016364  022626                          CMP     (SP)+,(SP)+
3430  016366  104001                  1$:     ERROR   1
3431  016370  000473                          BR      QDONE
3432
3433  016372  011637  001236          QERR20: MOV     (SP),@#$TMP2    ;REPORT R0 AFFECTED.
3434  016376  022626                          CMP     (SP)+,(SP)+
3435  016400  000403                          BR      QERR22
3436  016402  012737  016106  001236  QERR21: MOV     #Q2,@#$TMP2
3437  016410                          QERR22:
3438  016410  010037  001240                  MOV     R0,@#$TMP3
3439  016414  012737  016267  001242          MOV     #QPAT20-241,@#$TMP4
3440  016422  104001                  1$:     ERROR   1
3441  016424  000455                          BR      QDONE
3442
3443  016426  012737  000320  001246  QERR2:  MOV     #320,@#$TMP6    ;WENT TO FSRC
3444  016434  000403                          BR      QERR4           ;MODE 0
```

```
3445  016436  012737  000326  001246  QERR3:  MOV     #326,@#$TMP6    ;WENT TO FSRC
3446                                                                  ;MODE 6
3447  016444  012737  000627  001244  QERR4:  MOV     #627,@#$TMP5
3448  016452  012737  000327  001250          MOV     #327,@#$TMP7
3449  016460  012737  016106  001236          MOV     #Q2,@#$TMP2
3450  016466  104001                  1$:     ERROR   1
3451  016470  000433                          BR      QDONE
3452
3453  016472                          QERR1:                          ;DATA FAILURE
3454  016472  012737  016106  001236          MOV     #Q2,@#$TMP2
3455  016500  012737  016550  001240          MOV     #QDATI0,@#$TMP3
3456  016506  012737  016540  001242          MOV     #QDAT00,@#$TMP4
3457  016514  104001                  1$:     ERROR   1
3458  016516  000420                          BR      QDONE
3459
3460  016520  177777                  QPAT10: .WORD   -1
3461  016522  177777                  QPAT11:         -1
3462  016524  177777                  QPAT12:         -1
3463  016526  177777                  QPAT13:         -1
3464
3465  016530  016550                  QPAT20: .WORD   QDATI0
3466  016532  052525                  QPAT21:         52525
3467  016534  052525                  QPAT22:         52525
3468  016536  052525                  QPAT23:         52525
3469
3470  016540  000000                  QDAT00: .WORD   0
3471  016542  000000                  QDAT01:         0
3472  016544  000000                  QDAT02:         0
3473  016546  000000                  QDAT03:         0
3474
3475  016550  073567                  QDATI0: .WORD   073567
3476  016552  052525                  QDATI1: .WORD   052525
3477  016554  031463                  QDATI2: .WORD   031463
3478  016556  010421                  QDATI3: .WORD   010421
3479
3480  016560                          QDONE:
3481  016560  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
3482                                                                  ;SEE IF THE USER HAS EXPRESSED
3483                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
3484                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3485                                                                  ;THE USER TYPED CONTROL G?).
3486
3487                                  ;;*****************************************************************
3488                                  ;*TEST 24       (BUT EZBT Y8),(BUT ENBT) AND (BUT FIUV) TEST
3489                                  ;*
3490                                  ;* LOAD INSTRUCTION FLOWS.
3491                                  ;* EACH OF THE PATTERNS:
3492                                  ;*                 0
3493                                  ;*                 +NUM
3494                                  ;*                 -NUM
3495                                  ;*                 -0
3496                                  ;* IS LOADED TWICE, ONCE WITH AC>0 THEN
3497                                  ;* WITH AC=0.  AFTER EACH LOAD THE FPS IS
3498                                  ;* CHECK TO INSURE THAT CONTROL WAS PASSED
3499                                  ;* THROUGH WITH THE FORKS PROPERLY.
3500                                  ;*
```

```
3501                                    ;;*****************************************************************
3502  016562  000004                    TST24:  SCOPE
3503  016564  005037  017662                    CLR     @#UFLAG
3504  016570  012700  017612                    MOV     #UPAT00,R0      ;SET UP AC#0 DATA.
3505  016574  012701  000004                    MOV     #4,R1
3506  016600  012720  177777            U0:     MOV     #-1,(R0)+
3507  016604  077103                            SOB     R1,U0
3508
3509  016606  012737  000033  017664            MOV     #033,@#UTMP1
3510  016614  012737  000023  017666            MOV     #023,@#UTMP2
3511  016622  012737  017342  000244            MOV     #UERR0,@#FPVECT ;IN CASE (BUT FIUV FAILS)
3512  016630                            U1:
3513  016630  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3514  016632  012700  000200                    MOV     #200,R0
3515  016636  170100                            LDFPS   R0
3516  016640  012700  017612                    MOV     #UPAT00,R0      ;LOAD AC0
3517  016644  172410                            LDD     (R0),AC0
3518  016646  013737  017664  017670            MOV     @#UTMP1,@#UROM1
3519  016654  012737  000001  017672            MOV     #001,@#UROM2
3520  016662  012737  000254  017674            MOV     #254,@#UROM3
3521
3522  016670  012700  017622                    MOV     #UPAT10,R0      ;LOAD 0 INTO AC0
3523  016674  172410                    U2:     LDD     (R0),AC0
3524  016676  010037  001252                    MOV     R0,@#$TMP10
3525  016702  012737  016674  001236            MOV     #U2,@#$TMP2
3526
3527  016710  012704  000204                    MOV     #204,R4         ;SEE IF FPS IS CORRECT
3528  016714  170205                            STFPS   R5
3529
3530  016716  020405                            CMP     R4,R5
3531  016720  001402                            BEQ     U3
3532  016722  000137  017366                    JMP     @#UERR1
3533
3534  016726                            U3:
3535  016726  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3536  016730  012700  000200                    MOV     #200,R0
3537  016734  170100                            LDFPS   R0
3538
3539  016736  012700  017612                    MOV     #UPAT00,R0      ;LOAD AC0
3540  016742  172410                            LDD     (R0),AC0
3541  016744  013737  017666  017670            MOV     @#UTMP2,@#UROM1
3542  016752  012737  000003  017672            MOV     #003,@#UROM2
3543  016760  012737  000054  017674            MOV     #054,@#UROM3
3544
3545  016766  012700  017632                    MOV     #UPAT20,R0      ;LOAD A POSITIVE NUMBER
3546                                                                    ;INTO AC0
3547  016772  172410                    U4:     LDD     (R0),AC0
3548  016774  010037  001252                    MOV     R0,@#$TMP10
3549  017000  012737  016772  001236            MOV     #U4,@#$TMP2
3550  017006  012704  000200                    MOV     #200,R4         ;FPS CORRECT?
3551  017012  170205                            STFPS   R5
3552
3553  017014  020405                            CMP     R4,R5
3554  017016  001402                            BEQ     U5
3555  017020  000137  017452                    JMP     @#UERR2
3556  017024                            U5:
```

```
3557  017024  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3558  017026  012700  000200                  MOV     #200,R0
3559  017032  170100                          LDFPS   R0
3560  017034  012700  017612                  MOV     #UPAT00,R0      ;LOAD AC0
3561  017040  172410                          LDD     (R0),AC0
3562  017042  013737  017666  017670          MOV     @#UTMP2,@#UROM1
3563  017050  012737  000403  017672          MOV     #403,@#UROM2
3564  017056  012737  000056  017674          MOV     #056,@#UROM3
3565  017064  012700  017642                  MOV     #UPAT30,R0      ;LOAD A NEGATIUE
3566                                                                  ;NUMBER INTO AL0
3567  017070  172410                  U6:     LDD     (R0),AC0
3568  017072  010037  001252                  MOV     R0,@#$TMP10
3569  017076  012737  017070  001236          MOV     #U6,@#$TMP2
3570  017104  012704  000210                  MOV     #210,R4         ;FPS CORRECT
3571  017110  170205                          STFPS   R5
3572  017112  020405                          CMP     R4,R5
3573  017114  001402                          BEQ     U7
3574  017116  000137  017452                  JMP     @#UERR2
3575  017122                          U7:
3576  017122  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3577  017124  012700  000200                  MOV     #200,R0
3578  017130  170100                          LDFPS   R0
3579  017132  012700  017612                  MOV     #UPAT00,R0      ;LOAD AC0
3580  017136  172410                          LDD     (R0),AC0
3581  017140  013737  017664  017670          MOV     @#UTMP1,@#UROM1
3582  017146  012737  000401  017672          MOV     #401,@#UROM2
3583  017154  012737  000256  017674          MOV     #256,@#UROM3
3584  017162  012700  017652                  MOV     #UPAT40,R0      ;LOAD -0 INTO AC0
3585  017166  172410                  U10:    LDD     (R0),AC0
3586  017170  000240                  U11:    NOP                     ;TRAP FROM HERE IF
3587  017172  010037  001252                  MOV     R0,@#$TMP10
3588  017176  012737  017166  001236          MOV     #U10,@#$TMP2    ;(BUT FIUV) FAULS!
3589  017204  012704  000214                  MOV     #214,R4         ;SEE IF FPS IS CORRECT.
3590  017210  170205                          STFPS   R5
3591  017212  020405                          CMP     R4,R5
3592  017214  001402                          BEQ     U12
3593  017216  000137  017366                  JMP     @#UERR1
3594  017222  005737  017662          U12:    TST     @#UFLAG ;SEE IF ALL THE PATTERNS
3595  017226  001021                          BNE     U14             ;HAVE BEEN TEST WITH
3596                                                                  ;BOTH AC NOT EQUAL TO 0 AND AC=0
3597  017230  012700  017612                  MOV     #UPAT00,R0      ;IF NOT GO BACK AND
3598  017234  012701  000004                  MOV     #4,R1           ;CHECK THEM WITH AC=0
3599  017240  005020                  U13:    CLR     (R0)+
3600  017242  077102                          SOB     R1,U13
3601  017244  012737  177777  017662          MOV     #-1,@#UFLAG
3602  017252  012737  000233  017664          MOV     #233,@#UTMP1
3603  017260  012737  000223  017666          MOV     #223,@#UTMP2
3604  017266  000137  016630                  JMP     @#U1
3605  017272                          U14:
3606  017272  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3607                                  ;NOW SEE IF A TRAP CAN BE FORCED BY SETTING FIUV AND LOADING -0
3608  017274  012737  017536  000244          MOV     #UERR3,@#FPVECT
3609  017302  012700  004200                  MOV     #4200,R0        ;SET FD AND FIUV
3610  017306  170100                          LDFPS   R0
3611  017310  012700  017612                  MOV     #UPAT00,R0      ;SET UP AC0
3612  017314  172410                          LDD     (R0),AC0
```

```
3613  017316  012700  017652            MOV     #UPAT40,R0      ;LOAD -0
3614  017322  172410            U15:    LDD     (R0),AC0        ;SHOULD TRAP TO 244
3615  017324  170000            U16:    CFCC
3616  017326  000240                    NOP
3617  017330  012737  017322  001236    MOV     #U15,@#$TMP2    ;REPORT ERROR.
3618                                                            ;DIDN'T TRAP
3619  017336  104001            1$:     ERROR   1               ;(BUT FIUV) FAILED.
3620  017340  000556                    BR      UDONE
3621
3622                            ;TRAPPED TO 244.  DID (BUT FIUV) FAIL?
3623  017342  021627  017170    UERR0:  CMP     (SP),#U11
3624  017346  001402                    BEQ     1$
3625  017350  000137  062524            JMP     @#FPSPUR
3626  017354  011637  001236    1$:     MOV     (SP),@#$TMP2
3627  017360  022626                    CMP     (SP)+,(SP)+
3628  017362  104001            2$:     ERROR   1
3629  017364  000544                    BR      UDONE
3630
3631                            ;COME HERE TO ANALYZE FPS ERRORS
3632
3633  017366  032705  000004    UERR1:  BIT     #4,R5
3634  017372  001432                    BEQ     UERR20
3635  017374  012737  000443  001244  UERR10: MOV     #443,@#$TMP5
3636  017402  013703  017674            MOV     @#UROM3,R3
3637  017406  010337  001250            MOV     R3,@#$TMP7
3638  017412  032703  000200            BIT     #200,R3
3639  017416  001403                    BEQ     1$
3640  017420  042703  000200            BIC     #200,R3
3641  017424  000402                    BR      2$
3642  017426  052703  000200    1$:     BIS     #200,R3
3643  017432  010337  001246    2$:     MOV     R3,@#$TMP6
3644  017436  010537  001240    UERR11: MOV     R5,@#$TMP3
3645  017442  010437  001242            MOV     R4,@#$TMP4
3646  017446  104001            1$:     ERROR   1
3647  017450  000512                    BR      UDONE
3648  017452  032705  000004    UERR2:  BIT     #4,R5
3649  017456  001746                    BEQ     UERR10
3650  017460  013737  017670  001244  UERR20: MOV     @#UROM1,@#$TMP5
3651  017466  013703  017672            MOV     @#UROM2,R3
3652  017472  010337  001250            MOV     R3,@#$TMP7
3653  017476  032703  000400            BIT     #400,R3
3654  017502  001403                    BEQ     1$
3655  017504  042703  000400            BIC     #400,R3
3656  017510  000402                    BR      2$
3657  017512  052703  000400    1$:     BIS     #400,R3
3658  017516  010337  001246    2$:     MOV     R3,@#$TMP6
3659  017522  010537  001240    UERR21: MOV     R5,@#$TMP3
3660  017526  010437  001242            MOV     R4,@#$TMP4
3661  017532  104001            1$:     ERROR   1
3662  017534  000460                    BR      UDONE
3663
3664                            ;INTERRUPT HERE WHEN FIUV SET AND ATTEMPTED TO LOAD-0
3665  017536  021627  017324    UERR3:  CMP     (SP),#U16
3666  017542  001402                    BEQ     1$
3667  017544  000137  062524            JMP     @#FPSPUR
3668  017550  022626            1$:     CMP     (SP)+,(SP)+
```

```
3669  017552  005000                          CLR     R0
3670  017554  170300                          STST    R0              ;GET FEC.
3671  017556  022700  000014                  CMP     #14,R0          ;CORRECT
3672  017562  001001                          BNE     UERR4
3673  017564  000444                          BR      UDONE
3674  017566  012737  017322  001236  UERR4:  MOV     #U15,@#$TMP2
3675  017574  012737  000012  001242          MOV     #12,@#$TMP4
3676  017602  010037  001240                  MOV     R0,@#$TMP3
3677  017606  104001                  1$:     ERROR   1
3678  017610  000432                          BR      UDONE
3679  017612  000000                  UPAT00: .WORD   0
3680  017614  000000                  UPAT01:         0
3681  017616  000000                  UPAT02:         0
3682  017620  000000                  UPAT03:         0
3683
3684  017622  000000                  UPAT10: .WORD   0               ;0
3685  017624  000000                  UPAT11:         0
3686  017626  000000                  UPAT12:         0
3687  017630  000000                  UPAT13:         0
3688
3689  017632  010421                  UPAT20: .WORD   010421          ;POS NUM
3690  017634  114631                  UPAT21:         114631
3691  017636  125252                  UPAT22:         125252
3692  017640  177777                  UPAT23:         177777
3693
3694  017642  114631                  UPAT30:         114631          ;NEG NUM
3695  017644  135673                  UPAT31:         135673
3696  017646  146314                  UPAT32:         146314
3697  017650  167356                  UPAT33:         167356
3698
3699  017652  100000                  UPAT40:         100000          ;NEG ZERO
3700  017654  000000                  UPAT41:         0
3701  017656  000000                  UPAT42:         0
3702  017660  000000                  UPAT43:         0
3703
3704  017662  000000                  UFLAG:  .WORD   0
3705  017664  000000                  UTMP1:          0
3706  017666  000000                  UTMP2:          0
3707  017670  000000                  UROM1:          0
3708  017672  000000                  UROM2:          0
3709  017674  000000                  UROM3:          0
3710  017676                          UDONE:
3711
3712
3713                                  ;;*****************************************************************
3714                                  ;*TEST 25       ADDF,ADDD,SUBF AND SUBD WITH FSRC=AC-0 TEST
3715
3716                                  ;*
3717                                  ;* THIS IS A TEST OF ADD AND SUB WITH FSRC-AC=0
3718                                  ;*
3719
3720                                  ;;*****************************************************************
3721  017676  000004                  TST25:  SCOPE
3722  017700                          W1:
3723  017700  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3724  017702  012700  000200                  MOV     #200,R0
```

```
3725  017706  170100                        LDFPS   R0              ;SET DOUBLE MODE
3726  017710  012700  020364                MOV     #WPAT00,R0      ;LOAD AC0=:
3727  017714  172410                        LDD     (R0),AC0
3728  017716  012737  017730  001236        MOV     #W2,@#$TMP2     ;SAVE FP INSTRUCTION
3729  017724  012700  020364                MOV     #WPAT00,R0
3730  017730  172010                W2:     ADDD    (R0),AC0        ;TEST INSTRUCTION. ADD ITSELF
3731  017732  170205                        STFPS   R5              ;GET FPS
3732  017734  170011                        SETD                    ;SET DOUBLE MODE
3733  017736  012700  020364                MOV     #WPAT00,R0
3734  017742  174010                        STD     AC0,(R0)        ;GET THE RESULT
3735  017744  012701  020364                MOV     #WPAT00,R1
3736  017750  012702  000004                MOV     #4,R2
3737  017754  022021                W3:     CMP     (R0)+,(R1)+     ;IS RESULT CORRECT
3738  017756  001403                        BEQ     W4
3739                                                                ;NO
3740  017760  104002                1$:     ERROR   2
3741  017762  000137  020404                JMP     @#WDONE
3742  017766  077206                W4:     SOB     R2,W3
3743  017770  022705  000204                CMP     #204,R5         ;IS FPS CORRECT
3744  017774  001410                        BEQ     W5
3745                                                                ;NO
3746  017776  012737  000204  001242        MOV     #204,@#$TMP4
3747  020004  010537  001240                MOV     R5,@#$TMP3
3748  020010  104001                1$:     ERROR   1
3749  020012  000137  020404                JMP     @#WDONE
3750  020016                        W5:
3751  020016  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3752  020020  012700  000200                MOV     #200,R0
3753  020024  170100                        LDFPS   R0              ;SET DOUBLE MODE
3754  020026  012700  020364                MOV     #WPAT00,R0      ;LOAD AC0=0
3755  020032  172410                        LDD     (R0),AC0
3756  020034  012737  020052  001236        MOV     #W6,@#$TMP2
3757  020042  005000                        CLR     R0
3758  020044  170100                        LDFPS   R0              ;GO TO FLOATING MODE
3759  020046  012700  020364                MOV     #WPAT00,R0
3760  020052  172010                W6:     ADDF    (R0),AC0        ;TEST INSTRUCTION
3761  020054  170205                        STFPS   R5              ;GET FPS
3762  020056  170011                        SETD                    ;RESET TO DOUBLE MODE
3763  020060  012700  020364                MOV     #WPAT00,R0
3764  020064  174010                        STD     AC0,(R0)        ;GET THE RESULT
3765  020066  012701  020364                MOV     #WPAT00,R1
3766  020072  012702  000004                MOV     #4,R2
3767  020076  022021                W7:     CMP     (R0)+,(R1)+     ;WAS THE RESULT
3768  020100  001402                        BEQ     W10             ;NO. REPORT FAILURE.
3769  020102  104002                1$:     ERROR   2
3770  020104  000537                        BR      WDONE
3771  020106  077205                W10:    SOB     R2,W7
3772  020110  022705  000004                CMP     #4,R5           ;WAS FPS CORRECT
3773  020114  001407                        BEQ     W11
3774                                                                ;INCORRECT FPS.
3775  020116  012737  000004  001242        MOV     #4,@#$TMP4
3776  020124  010537  001240                MOV     R5,@#$TMP3
3777  020130  104002                1$:     ERROR   2
3778  020132  000524                        BR      WDONE
3779  020134                        W11:
3780  020134  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
3781  020136  012700  000200                MOV     #200,R0
3782  020142  170100                        LDFPS   R0              ;SET DOUBLE MODE
3783  020144  012700  020364                MOV     #WPAT00,R0      ;LOAD AC0=0
3784  020150  172410                        LDD     (R0),AC0
3785  020152  012737  020164  001236        MOV     #W12,@#$TMP2
3786  020160  012700  020364                MOV     #WPAT00,R0
3787  020164  173010            W12:        SUBD    (R0),AC0        ;TEST INSTRUCTION
3788  020166  170205                        STFPS   R5              ;GET FPS
3789  020170  170011                        SETD                    ;SET DOUBLE MODE
3790  020172  012700  020364                MOV     #WPAT00,R0
3791  020176  174010                        STD     AC0,(R0)        ;GET THE RESULT
3792  020200  012701  020364                MOV     #WPAT00,R1
3793  020204  012702  000004                MOV     #4,R2
3794  020210  022021            W13:        CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
3795  020212  001402                        BEQ     W14
3796                                                                ;NO.
3797  020214  104002            1$:         ERROR   2
3798  020216  000472                        BR      WDONE
3799  020220  077205            W14:        SOB     R2,W13
3800  020222  022705  000204                CMP     #204,R5         ;IS FPS CORRECT?
3801  020226  001407                        BEQ     W15
3802                                                                ;NO.
3803  020230  012737  000204  001242        MOV     #204,@#$TMP4
3804  020236  010537  001240                MOV     R5,@#$TMP3
3805  020242  104001            1$:         ERROR   1
3806  020244  000457                        BR      WDONE
3807  020246                    W15:
3808  020246  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3809  020250  012700  000200                MOV     #200,R0
3810  020254  170100                        LDFPS   R0              ;SET DOUBLE MODE
3811  020256  012700  020364                MOV     #WPAT00,R0      ;LOAD AC0=0
3812  020262  172410                        LDD     (R0),AC0
3813  020264  012737  020302  001236        MOV     #W16,@#$TMP2
3814  020272  005000                        CLR     R0
3815  020274  170100                        LDFPS   R0              ;ENTER FLOATING MODE.
3816  020276  012700  020364                MOV     #WPAT00,R0
3817  020302  173010            W16:        SUBF    (R0),AC0        ;TEST INSTRUCTION.
3818  020304  170205                        STFPS   R5              ;GET FPS
```

```
3819  020306  170011                          SETD                    ;RESET TO DOUBLE MODE
3820  020310  012700  020364                   MOV     #WPAT00,R0      ;GET THE RESULT.
3821  020314  174010                          STD     AC0,(R0)
3822  020316  012701  020364                   MOV     #WPAT00,R1
3823  020322  012702  000004                   MOV     #4,R2
3824  020326  022021                  W17:     CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
3825  020330  001402                          BEQ     W20
3826                                                                   ;NO.
3827  020332  104002                  1$:      ERROR   2
3828  020334  000423                          BR      WDONE
3829  020336  077205                  W20:     SOB     R2,W17
3830  020340  022705  000004                   CMP     #4,R5           ;IS FPS CORRECT?
3831  020344  001417                          BEQ     WDONE
3832                                                                   ;NO
3833  020346  012737  000004  001242           MOV     #4,@#$TMP4
3834  020354  010537  001240                   MOV     R5,@#$TMP3
3835  020360  104001                  1$:      ERROR   1
3836  020362  000410                          BR      WDONE
3837
3838  020364  000000                  WPAT00: .WORD   0
3839  020366  000000                  WPAT01:         0
3840  020370  000000                  WPAT02:         0
3841  020372  000000                  WPAT03:         0
3842
3843  020374  000000                  WDAP00: .WORD   0
3844  020376  000000                  WDAT01:         0
3845  020400  000000                  WDAT02:         0
3846  020402  000000                  WDAT03:         0
3847
3848  020404                          WDONE:
3849  020404  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
3850                                                                  ;SEE IF THE USER HAS EXPRESSED
3851                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
3852                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3853                                                                  ;THE USER TYPED CONTROL G?).
3854
3855
3856                                  ;;*****************************************************************
3857                                  ;*TEST 26        ADDD AND SUB WITH FSRC=0
3858                                  ;*
3859                                  ;* THIS IS A TEST OF ADD AND SUB WITH FSRC=0.
3860                                  ;*
3861
3862                                  ;;*****************************************. ********************
3863  020406  000004                  TST26:  SCOPE
3864  020410                          X1:
3865  020410  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3866  020412  012700  000200                   MOV     #200,R0
3867  020416  170100                          LDFPS   R0              ;SET DOUBLE MODE
3868  020420  012700  021150                   MOV     #XPAT00,R0      ;SET AC0 TO POSITIVE
3869  020424  010037  021136                   MOV     R0,@#XTMP       ;NUMBER #0
3870  020430  172410                          LDD     (R0),AC0
3871  020432  012737  020444  001236           MOV     #X2,@#$TMP2
3872  020440  012700  021160                   MOV     #XPAT10,R0      ;FSRC=0
3873  020444  172010                  X2:     ADDD    (R0),AC0        ;TEST INSTRUCTION
3874  020446  170205                          STFPS   R5
```

```
3875  020450  170011                          SETD
3876  020452  012700  021140                  MOV     #XDAT00,R0      ;GET RESULT.
3877  020456  174010                          STD     AC0,(R0)
3878  020460  012701  021150                  MOV     #XPAT00,R1
3879  020464  012702  000004                  MOV     #4,R2
3880  020470  022021                  X3:     CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
3881  020472  001401                          BEQ     X4
3882  020474  000553                          BR      XERR1
3883  020476  077204                  X4:     SOB     R2,X3
3884  020500  012704  000200                  MOV     #200,R4
3885  020504  020405                          CMP     R4,R5           ;IS FPS CORRECT?
3886  020506  001402                          BEQ     X5
3887  020510  000137  021106                  JMP     @#XERR2
3888  020514                           X5:
3889  020514  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3890  020516  012700  000200                  MOV     #200,R0
3891  020522  170100                          LDFPS   R0              ;SET DOUBLE MODE
3892  020524  012700  021170                  MOV     #XPAT20,R0      ;SET AC0 TO
3893  020530  010037  021136                  MOV     R0,@#XTMP       ;NEGATIVE NUMBER
3894  020534  172410                          LDD     (R0),AC0
3895  020536  012737  020550  001236          MOV     #X6,@#STMP2
3896  020544  012700  021160                  MOV     #XPAT10,R0      ;FSRC=0
3897  020550  172010                  X6:     ADDD    (R0),AC0        ;TEST INSTRUCTION
3898  020552  170205                          STFPS   R5
3899  020554  170011                          SETD
3900  020556  012700  021140                  MOV     #XDAT00,R0      ;GET RESULT
3901  020562  174010                          STD     AC0,(R0)
3902  020564  012701  021170                  MOV     #XPAT20,R1
3903  020570  012702  000004                  MOV     #4,R2
3904  020574  022021                  X7:     CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
3905  020576  001401                          BEQ     X10
3906  020600  000511                          BR      XERR1
3907  020602  077204                  X10:    SOB     R2,X7
3908  020604  012704  000210                  MOV     #210,R4
3909  020610  020405                          CMP     R4,R5           ;IS FPS CORRECT?
3910  020612  001401                          BEQ     X11
3911  020614  000534                          BR      XERR2
3912  020616                           X11:
3913  020616  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3914  020620  012700  000200                  MOV     #200,R0
3915  020624  170100                          LDFPS   R0              ;SET DOUBLE MODE
3916  020626  012700  021150                  MOV     #XPAT00,R0      ;SET AC0 TO NON-ZERO
3917  020632  010037  021136                  MOV     R0,@#XTMP       ;POSITIVE NUMBER
3918  020636  172410                          LDD     (R0),AC0
3919  020640  012737  020652  001236          MOV     #X12,@#STMP2
3920  020646  012700  021160                  MOV     #XPAT10,R0      ;FSRC=0
3921  020652  173010                  X12:    SUBD    (R0),AC0        ;TEST INSTRUCTION
3922  020654  170205                          STFPS   R5
3923  020656  170011                          SETD
3924  020660  012700  021140                  MOV     #XDAT00,R0      ;GET RESULT
3925  020664  174010                          STD     AC0,(R0)
3926  020666  012701  021150                  MOV     #XPAT00,R1
3927  020672  012702  000004                  MOV     #4,R2
3928  020676  022021                  X13:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
3929  020700  001401                          BEQ     X14
3930  020702  000463                          BR      XERR3
```

```
3931  020704  077204                    X14:    SOB     R2,X13
3932  020706  012704  000200                    MOV     #200,R4         ;IS FPS CORRECT?
3933  020712  020405                            CMP     R4,R5
3934  020714  001401                            BEQ     X15
3935  020716  000501                            BR      XERR4
3936  020720                            X15:
3937  020720  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
3938  020722  012700  000200                    MOV     #200,R0
3939  020726  170100                            LDFPS   R0              ;SET DOUBLE MODE
3940  020730  012700  021170                    MOV     #XPAT20,R0      ;SET AC0=A NEGATIVE
3941  020734  010037  021136                    MOV     R0,@#XTMP       ;NUMBER
3942  020740  172410                            LDD     (R0),AC0
3943  020742  012737  020754  001236            MOV     #X16,@#$TMP2
3944  020750  012700  021160                    MOV     #XPAT10,R0      ;FSRC=0
3945  020754  173010                    X16:    SUBD    (R0),AC0        ;TEST INSTRUCTION.
3946  020756  170205                            STFPS   R5
3947  020760  170011                            SETD
3948  020762  012700  021140                    MOV     #XDAT00,R0      ;GET RESULT
3949  020766  174010                            STD     AC0,(R0)
3950  020770  012701  021170                    MOV     #XPAT20,R1
3951  020774  012702  000004                    MOV     #4,R2
3952  021000  022021                    X17:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
3953  021002  001401                            BEQ     X20
3954  021004  000422                            BR      XERR3
3955  021006  077204                    X20:    SOB     R2,X17
3956  021010  012704  000210                    MOV     #210,R4         ;IS FPS CORRECT?
3957  021014  020405                            CMP     R4,R5
3958  021016  001401                            BEQ     X21
3959  021020  000440                            BR      XERR4
3960  021022  000466                    X21:    BR      XDONE
3961
3962                                    ;REPORT DATA ERRORS
3963
3964  021024  012737  021160  001240    XERR1:  MOV     #XPAT10,@#$TMP3
3965  021032  013737  021136  001242            MOV     @#XTMP,@#$TMP4
3966  021040  012737  021140  001244            MOV     #XDAT00,@#$TMP5
3967  021046  104001                    1$:     ERROR   1
3968  021050  000453                            BR      XDONE
3969  021052  012737  021160  001240    XERR3:  MOV     #XPAT10,@#$TMP3
3970  021060  013737  021136  001242            MOV     @#XTMP,@#$TMP4
3971  021066  012737  021140  001244            MOV     #XDAT00,@#$TMP5
3972  021074  013737  021136  001246            MOV     @#XTMP,@#$TMP6
3973  021102  104002                    1$:     ERROR   2
3974  021104  000435                            BR      XDONE
3975
3976                                    ;REPORT FPS ERRORS
3977
3978  021106                            XERR2:
3979  021106  010537  001240                    MOV     R5,@#$TMP3
3980  021112  010437  001242                    MOV     R4,@#$TMP4
3981  021116  104001                    1$:     ERROR   1
3982  021120  000427                            BR      XDONE
3983  021122                            XERR4:
3984  021122  010537  001240                    MOV     R5,@#$TMP3
3985  021126  010437  001242                    MOV     R4,@#$TMP4
3986  021132  104001                    1$:     ERROR   1
```

```
 3987  021134  000421                          BR      XDONE
 3988  021136  000000                  XTMP:   .WORD   0
 3989  021140  000000                  XDAT00: .WORD   0
 3990  021142  000000                  XDAT01:         0
 3991  021144  000000                  XDAT02:         0
 3992  021146  000000                  XDAT03:         0
 3993
 3994  021150  010421                  XPAT00: .WORD   010421
 3995  021152  021042                  XPAT01:         021042
 3996  021154  031463                  XPAT02:         031463
 3997  021156  042104                  XPAT03:         042104
 3998
 3999  021160  000000                  XPAT10: .WORD   0
 4000  021162  000000                  XAPT11:         0
 4001  021164  000000                  XPAT12:         0
 4002  021166  000000                  XPAT13:         0
 4003  021170  104210                  XPAT20: .WORD   104210
 4004  021172  114631                  XPAT21:         114631
 4005  021174  125252                  XPAT22:         125252
 4006  021176  135673                  XPAT23:         135673
 4007
 4008  021200                          XDONE:
 4009  021200  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 4010                                                                  ;SEE IF THE USER HAS EXPRESSED
 4011                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 4012                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 4013                                                                  ;THE USER TYPED CONTROL G?).
 4014
 4015                                  ;;**************************************************************
 4016                                  ;*TEST 27        SUBD WITH AC=0 TEST
 4017                                  ;*
 4018                                  ;* THIS IS A TEST OF SUBD WITH AC=0.  BOTH POSITIVE
 4019                                  ;* AND NEGATIVE FSRC'S ARE TRIED.
 4020                                  ;*
 4021
 4022                                  ;;**************************************************************
 4023  021202  000004                  TST27:  SCOPE
 4024  021204  005037  021534                  CLR     @#YFLAG
 4025  021210  012737  021554  021536          MOV     #YPAT00,@#YTMP1 ;P
 4026  021216  012737  021564  021540          MOV     #YPAT10,@#YTMP2 ;N
 4027  021224  012737  000210  021542          MOV     #210,@#YTMP3
 4028  021232                          Y1:
 4029  021232  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4030  021234  012700  000200                  MOV     #200,R0
 4031  021240  170100                          LDFPS   R0              ;SET DOUBLE MODE
 4032  021242  012700  021574                  MOV     #YPAT20,R0      ;SET AC0=0
 4033  021246  172410                          LDD     (R0),AC0
 4034  021250  013700  021536                  MOV     @#YTMP1,R0
 4035  021254  173010                  Y2:     SUBD    (R0),AC0        ;TEST INSTRUCTION
 4036  021256  170205                          STFPS   R5
 4037  021260  170011                          SETD
 4038  021262  012700  021544                  MOV     #YDAT00,R0      ;GET RESULT
 4039  021266  174010                          STD     AC0,(R0)
 4040  021270  012702  000004                  MOV     #4,R2
 4041  021274  013701  021540                  MOV     @#YTMP2,R1      ;CHECK RESULT.
 4042  021300  022021                  Y3:     CMP     (R0)+,(R1)+
```

```
4043  021302  001026                        BNE     Y6
4044  021304  077203                        SOB     R2,Y3
4045  021306  023705  021542                CMP     @#YTMP3,R5      ;FPS CORRECT?
4046  021312  001401                        BEQ     Y4
4047  021314  000475                        BR      YERR3
4048  021316  005737  021534        Y4:     TST     @#YFLAG         ;FINISHED TEST?
4049  021322  001015                        BNE     Y5
4050  021324  012737  177777  021534        MOV     #-1,@#YFLAG
4051  021332  012737  021564  021536        MOV     #YPAT10,@#YTMP1
4052  021340  012737  021554  021540        MOV     #YPAT00,@#YTMP2
4053  021346  012737  000200  021542        MOV     #200,@#YTMP3
4054  021354  000726                        BR      Y1
4055  021356  000512                Y5:     BR      YDONE
4056  021360  012702  000004        Y6:     MOV     #4,R2
4057  021364  012700  021536                MOV     #YTMP1,R0       ;DID XOR OF SIGN BIT
4058  021370  012701  021544                MOV     #YDAT00,R1      ;FAIL?
4059  021374  022021                Y7:     CMP     (R0)+,(R1)+
4060  021376  001002                        BNE     YERR1
4061  021400  077203                        SOB     R2,Y7
4062  021402  000421                        BR      YERR2
4063  021404                        YERR1:                          ;DATA FAILURE
4064  021404  012737  021254  001236        MOV     #Y2,@#$TMP2
4065  021412  013737  021536  001240        MOV     @#YTMP1,@#$TMP3
4066  021420  012737  021574  001242        MOV     #YPAT20,@#$TMP4
4067  021426  012737  021544  001244        MOV     #YDAT00,@#$TMP5
4068  021434  013737  021540  001246        MOV     @#YTMP2,@#$TMP6
4069  021442  104001                1$:     ERROR   1
4070  021444  000457                        BR      YDONE
4071  021446                        YERR2:                          ;XOR OF SIGN BIT
4072  021446  012737  021254  001236        MOV     #Y2,@#$TMP2     ;FAILED
4073  021454  013737  021536  001240        MOV     @#YTMP1,@#$TMP3
4074  021462  012737  021574  001242        MOV     #YPAT20,@#$TMP4
4075  021470  012737  021544  001244        MOV     #YDAT00,@#$TMP5
4076  021476  013737  021540  001246        MOV     @#YTMP2,@#$TMP6
4077  021504  104002                1$:     ERROR   2
4078  021506  000436                        BR      YDONE
4079  021510                        YERR3:                          ;FPS WRONG.
4080  021510  012737  021254  001236        MOV     #Y2,@#$TMP2
4081  021516  010537  001240                MOV     R5,@#$TMP3
4082  021522  013737  021542  001242        MOV     @#YTMP3,@#$TMP4
4083  021530  104002                1$:     ERROR   2
4084  021532  000424                        BR      YDONE
4085
4086  021534  000000                YFLAG:  .WORD   0
4087  021536  000000                YTMP1:          0
4088  021540  000000                YTMP2:          0
4089  021542  000000                YTMP3:          0
4090
4091  021544  000000                YDAT00: .WORD   0
4092  021546  000000                YDAT01:         0
4093  021550  000000                YDAT02:         0
4094  021552  000000                YDAT03:         0
4095
4096  021554  063146                YPAT00:         063146
4097  021556  052525                YPAT01:         052525
4098  021560  042104                YPAT02:         042104
```

```
 4099  021562  167356                  YPAT03:         167356
 4100
 4101  021564  163146                  YPAT10:         163146
 4102  021566  052525                  YPAT11:         052525
 4103  021570  042104                  YPAT12:         042104
 4104  021572  167356                  YPAT13:         167356
 4105
 4106  021574  000000                  YPAT20:         0
 4107  021576  000000                  YPAT21:         0
 4108  021600  000000                  YPAT22:         0
 4109  021602  000000                  YPAT23:         0
 4110
 4111  021604                          YDONE:
 4112  021604  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 4113                                                                  ;SEE IF THE USER HAS EXPRESSED
 4114                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 4115                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 4116                                                                  ;THE USER TYPED CONTROL G?).
 4117                                  ;;*****************************************************************
 4118                                  ;*TEST 30        ADDD WITH AC=0 TEST
 4119
 4120
 4121                                  ;@
 4122                                  ;@ THIS IS A TEST OF ADDD WITH AC=0.  BOTH
 4123                                  ;* POSITIVE AND NEGATIVE FSRC'S ARE TRIED.
 4124                                  ;*
 4125
 4126                                  ;;*****************************************************************
 4127  021606  000004                  TST30:  SCOPE
 4128  021610  005067  000224                  CLR     ZFLAG
 4129  021614  012737  022056  022042          MOV     #ZPAT00,@#ZTMP1 ;P
 4130  021622  012737  000200  022044          MOV     #200,@#ZTMP2
 4131  021630                          Z1:
 4132  021630  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4133  021632  012700  000200                  MOV     #200,R0
 4134  021636  170100                          LDFPS   R0              ;SET DOUBLE MODE
 4135  021640  012700  022076                  MOV     #ZPAT20,R0      ;SET AC0=0
 4136  021644  172410                          LDD     (R0),AC0
 4137  021646  013700  022042                  MOV     @#ZTMP1,R0
 4138  021652  172010                  Z2:     ADDD    (R0),AC0        ;TEST INSTRUCTION
 4139  021654  170205                          STFPS   R5
 4140  021656  170011                          SETD
 4141  021660  012700  022046                  MOV     #ZDAT00,R0      ;GET RESULT
 4142  021664  174010                          STD     AC0,(R0)
 4143  021666  012702  000004                  MOV     #4,R2
 4144  021672  013701  022042                  MOV     @#ZTMP1,R1      ;RESULT CORRECT?
 4145  021676  022021                  Z3:     CMP     (R0)+,(R1)+
 4146  021700  001401                          BEQ     Z4
 4147  021702  000423                          BR      ZERR1
 4148  021704  077204                  Z4:     SOB     R2,Z3
 4149  021706  023705  022044                  CMP     @#ZTMP2,R5      ;FPS CORRECT?
 4150  021712  001401                          BEQ     Z5
 4151  021714  000437                          BR      ZERR2
 4152  021716  005737  022040          Z5:     TST     @#ZFLAG         ;FINISHED TEST?
 4153  021722  001012                          BNE     Z6
 4154  021724  012737  177777  022040          MOV     #-1,@#ZFLAG
```

```
 4155  021732  012737  022066  022042          MOV     #ZPAT10,a#ZTMP1
 4156  021740  012737  000210  022044          MOV     #210,a#ZTMP2
 4157  021746  000730                          BR      Z1
 4158  021750  000456                  Z6:     BR      ZDONE
 4159  021752                          ZERR1:                          ;DATA FAILURE
 4160  021752  012737  021652  001236          MOV     #Z2,a#$TMP2
 4161  021760  013737  022042  001240          MOV     a#ZTMP1,a#$TMP3
 4162  021766  012737  022076  001242          MOV     #ZPAT20,a#$TMP4
 4163  021774  012737  022046  001244          MOV     #ZDAT00,a#$TMP5
 4164  022002  013737  022042  001246          MOV     a#ZTMP1,a#$TMP6
 4165  022010  104002                  1$:     ERROR   2
 4166  022012  000435                          BR      ZDONE
 4167  022014                          ZERR2:
 4168  022014  012737  021652  001236          MOV     #Z2,a#$TMP2
 4169  022022  010537  001240                  MOV     R5,a#$TMP3
 4170  022026  013737  022044  001242          MOV     a#ZTMP2,a#$TMP4
 4171  022034  104002                  1$:     ERROR   2
 4172  022036  000423                          BR      ZDONE
 4173
 4174  022040  000000                  ZFLAG:  .WORD   0
 4175  022042  000000                  ZTMP1:          0
 4176  022044  000000                  ZTMP2:          0
 4177
 4178  022046  000000                  ZDAT00: .WORD   0
 4179  022050  000000                  ZDAT01:         0
 4180  022052  000000                  ZDAT02:         0
 4181  022054  000000                  ZDAT03:         0
 4182
 4183  022056  031463                  ZPAT00:         031463
 4184  022060  010421                  ZPAT01:         010421
 4185  022062  146314                  ZPAT02:         146314
 4186  022064  156735                  ZPAT03:         156735
 4187
 4188  022066  156735                  ZPAT10:         156735
 4189  022070  167356                  ZPAT11:         167356
 4190  022072  135673                  ZPAT12:         135673
 4191  022074  146314                  ZPAT13:         146314
 4192
 4193  022076  000000                  ZPAT20:         0
 4194  022100  000000                  ZPAT21:         0
 4195  022102  000000                  ZPAT22:         0
 4196  022104  000000                  ZPAT23:         0
 4197
 4198  022106                          ZDONE:
 4199  022106  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 4200                                                                  ;SEE IF *THE USER HAS EXPRESSED
 4201                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 4202                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 4203                                                                  ;THE USER TYPED CONTROL G?).
 4204
 4205
 4206                                  ;;******************************************************************
 4207                                  ;*TEST 31       ADDF AND ADDD WITH E(AC)=E(FSRC) TEST AND (BUT FT) TEST
 4208                                  ;*
 4209                                  ;* THIS IS A TEST OF THE ADD INSTRUCTION WITH THE
 4210                                  ;* OPERANDS HAVING EQUAL EXPONENTS.
```

```
4211                                    ;* THE ROUND/TRUNK FLOWS IS ALSO TESTED.
4212                                    ;*
4213                                    ;;*****************************************************************
4214  022110  000004                    TST31:  SCOPE
4215  022112                            AA1:
4216  022112  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4217  022114  012700  003240                    MOV     #3240,R0
4218  022120  170100                            LDFPS   R0              ;SET FIU FIV FD AND FT
4219  022122  012737  022472  000244            MOV     #AAERR0,@#FPVECT  ;IN CASE THE OVER/UNDER
4220  022130  012700  023050                    MOV     #AAPAT0,R0      ;FLOWS IN TRAP WILL
4221                                                                    ;OCCUR
4222  022134  172410                            LDD     (R0),AC0        ;SET UP AC0
4223  022136  012737  022150  001236            MOV     #AA2,@#$TMP2    ;OPERAND
4224  022144  012700  023060                    MOV     #AAPAT1,R0
4225  022150  172010                    AA2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
4226                                                                    ;SHOULD TRUNCATE
4227  022152  012700  023040            AA3:    MOV     #AADAT0,R0
4228  022156  174010                            STD     AC0,(R0)        ;GET THE RESULT
4229  022160  012701  023070                    MOV     #AAPAT2,R1
4230  022164  012702  000004                    MOV     #4,R2
4231  022170  022021                    AA4:    CMP     (R0)+,(R1)+     ;CORRECT?
4232
4233  022172  001414                            BEQ     AA7
4234  022174  012700  023100                    MOV     #AAPAT3,R0      ;DID (BUT FT) FAIL
4235  022200  012701  023040                    MOV     #AADAT0,R1
4236  022204  012702  000004                    MOV     #4,R2
4237  022210  022021                    AA5:    CMP     (R0)+,(R1)+
4238  022212  001401                            BEQ     AA6
4239  022214  000561                            BR      AAERR1  ;DATA ERROR
4240  022216  077204                    AA6:    SOB     R2,AA5
4241  022220  000137  022614                    JMP     @#AAERR2        ;(BUT FT) ERROR
4242  022224  077217                    AA7:    SOB     R2,AA4
4243
4244                                    ;NOW TEST DOUBLE FLOATING ROUND MODE.
4245                                    ;A 1 SHOULD BE ADDED TO THE LSB ON ROUNG MODE.
4246
4247  022226                            AA10:
4248  022226  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4249  022230  012700  003200                    MOV     #3200,R0        ;SET FD FIV FIV. FT-0
4250  022234  170100                            LDFPS   R0
4251  022236  012700  023050                    MOV     #AAPAT0,R0
4252  022242  172410                            LDD     (R0),AC0        ;SET UP AC0 OPERAND
4253  022244  012737  022256  001236            MOV     #AA11,@#$TMP2
4254  022252  012700  023060                    MOV     #AAPAT1,R0
4255  022256  172010                    AA11:   ADDD    (R0),AC0        ;TEST INSTRUCTION
4256                                                                    ;SHOULD ROUND
4257  022260  012700  023040            AA12:   MOV     #AADAT0,R0
4258  022264  174010                            STD     AC0,(R0)        ;GET THE RESULT
4259  022266  012701  023100                    MOV     #AAPAT3,R1
4260  022272  012702  000004                    MOV     #4,R2
4261  022276  022021                    AA13:   CMP     (R0)+,(R1)+     ;CORRECT?
4262  022300  001425                            BEQ     AA20
4263  022302  012700  023070                    MOV     #AAPAT2,R0      ;DID (BUT FT) FAIL?
4264  022306  012701  023040                    MOV     #AADAT0,R1
4265  022312  012702  000004                    MOV     #4,R2
4266  022316  022021                    AA14:   CMP     (R0)+,(R1)+
```

```
 4267  022320  001413                        BEQ     AA17
 4268  022322  012700  023110                MOV     #AAPAT4,R0      ;WAS THE FLOATING
 4269  022326  012701  023040                MOV     #AADATO,R1      ;CONSTANT USED
 4270  022332  012702  000004                MOV     #4,R2           ;INSTEAD OF THE
 4271  022336  022021                AA15:   CMP     (R0)+,(R1)+     ;DOUBLE CONSTANT
 4272  022340  001401                        BEQ     AA16            ;IN THE ROUND
 4273  022342  000542                        BR      AAERR3  ;FLOWS?
 4274  022344  077204                AA16:   SOB     R2,AA15         ;DATA ERROR
 4275  022346  000544                        BR      AAERR4  ;CONSTANT ERROR
 4276  022350  077216                AA17:   SOB     R2,AA14
 4277  022352  000560                        BR      AAERR5  ;(BUT FT) ERROR
 4278  022354  077230                AA20:   SOB     R2,AA13
 4279
 4280                                ;NOW TEST ADDF WITH FT=0, ROUND MODE
 4281
 4282  022356                        AA21:
 4283  022356  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4284  022360  012700  003200                MOV     #3200,R0                ;FIV=1, FIV 1, FT=0
 4285  022364  170100                        LDFPS   R0
 4286  022366  012700  023050                MOV     #AAPATO,R0      ;LOAD AC0 OPERAND
 4287  022372  172410                        LDD     (R0),AC0
 4288  022374  170001                        SETF                    ;ENTER FLOATING MODE
 4289  022376  012737  022410  001236        MOV     #AA22,@#$TMP2
 4290  022404  012700  023120                MOV     #AAPAT5,R0
 4291  022410  172010                AA22:   ADDF    (R0),AC0        ;TEST INSTRUCTION
 4292                                                                ;SHOULD ROUND
 4293  022412                        AA23:
 4294  022412  170011                        SETD                    ;RESET TO DOUBLE
 4295                                                                ;MODE
 4296  022414  012700  023040                MOV     #AADATO,R0      ;GET THE RESULT
 4297  022420  174010                        STD     AC0,(R0)
 4298  022422  012701  023130                MOV     #AAPAT6,R1      ;CORRECT?
 4299  022426  012702  000002                MOV     #2,R2
 4300  022432  022021                AA24:   CMP     (R0)+,(R1)+
 4301  022434  001413                        BEQ     AA27
 4302  022436  012700  023070                MOV     #AAPAT2,R0      ;WAS THE DOUBLE
 4303  022442  012701  023040                MOV     #AADATO,R1      ;CONSTANT USED INSTEAD
 4304  022446  012702  000002                MOV     #2,R2           ;OF THE FLOATING
 4305  022452  022011                AA25:   CMP     (R0)+,(R1)      ;CONSTANT IN THE
 4306  022454  001401                        BEQ     AA26            ;ROUND FLOWS?
 4307  022456  000534                        BR      AAERR6  ;DATA ERROR
 4308  022460  077204                AA26:   SOB     R2,AA25
 4309  022462  000550                        BR      AAERR7  ;CONSTANT ERROR
 4310  022464  077216                AA27:   SOB     R2,AA24
 4311  022466  000137  023140                JMP     @#AADONE
 4312
 4313                                ;COME HERE IF A TRAP OCCURS TO 244.
 4314
 4315  022472  013700  001236        AAERR0: MOV     @#$TMP2,R0      ;SEE IF THE TRAP WAS
 4316  022476  005720                        TST     (R0)+   ;AT A TEST INSTRUCTION
 4317  022500  020016                        CMP     R0,(SP)
 4318  022502  001402                        BEQ     1$
 4319  022504  000137  062524        10$:    JMP     @#FPSPUR
 4320  022510                        1$:
 4321  022510  170300                        STST    R0              ;GET FEC
 4322  022512  020027  000010                CMP     R0,#10
```

```
 4323  022516  001405                          BEQ     20$             ;OVERFLOW
 4324  022520  020027  000012                  CMP     R0,#12
 4325  022524  001410                          BEQ     30$             ;UNDERFLOW
 4326  022526  000766                          BR      10$
 4327  022530  022532                  20$
 4328  022532  011637  001236          20$:    MOV     (SP),@#$TMP2    ;REPORT OVERFLOW ERROR
 4329  022536  022626                          CMP     (SP)+,(SP)+
 4330  022540  104002                  21$:    ERROR   2
 4331  022542  000137  023140          25$:    JMP     @#AADONE
 4332  022546  011637  001236          30$:    MOV     (SP),@#$TMP2    ;REPORT UNDERFLOW
 4333  022552  022626                          CMP     (SP)+,(SP)+     ;ERROR
 4334  022554  104002                  31$:    ERROR   2
 4335  022556  000771                          BR      25$
 4336
 4337                                  ;ADDD RESULT INCORRECT
 4338  022560  012737  023070  001246  AAERR1: MOV     #AAPAT2,@#$TMP6
 4339  022566  012737  023050  001242  AAER10: MOV     #AAPAT0,@#$TMP4
 4340  022574  012737  023060  001240          MOV     #AAPAT1,@#$TMP3
 4341  022602  012737  023040  001244          MOV     #AADAT0,@#$TMP5
 4342  022610  104002                  1$:     ERROR   2
 4343  022612  000552                          BR      AADONE
 4344  022614  012737  023070  001246  AAERR2: MOV     #AAPAT2,@#$TMP6 ;(BUT FT) FAILED.
 4345  022622  012737  023050  001242          MOV     #AAPAT0,@#$TMP4
 4346  022630  012737  023060  001240          MOV     #AAPAT1,@#$TMP3
 4347  022636  012737  023040  001244          MOV     #AADAT0,@#$TMP5
 4348  022644  104002                  1$:     ERROR   2
 4349  022646  000534                          BR      AADONE
 4350  022650  012737  023100  001246  AAERR3: MOV     #AAPAT3,@#$TMP6 ;DATA ERROR.
 4351  022656  000743                          BR      AAER10
 4352  022660  012737  023100  001246  AAERR4: MOV     #AAPAT3,@#$TMP6 ;BAD CONSTANT
 4353  022666  012737  023050  001242          MOV     #AAPAT0,@#$TMP4
 4354  022674  012737  023060  001240          MOV     #AAPAT1,@#$TMP3
 4355  022702  012737  023040  001244          MOV     #AADAT0,@#$TMP5
 4356  022710  104002                  1$:     ERROR   2
 4357  022712  000512                          BR      AADONE
 4358  022714  012737  023100  001246  AAERR5: MOV     #AAPAT3,@#$TMP6 ;(BUT FT) FAILED.
 4359  022722  012737  023050  001242          MOV     #AAPAT0,@#$TMP4
 4360  022730  012737  023060  001240          MOV     #AAPAT1,@#$TMP3
 4361  022736  012737  023040  001244          MOV     #AADAT0,@#$TMP5
 4362  022744  104002                  1$:     ERROR   2
 4363  022746  000474                          BR      AADONE
 4364  022750  012737  023120  001240  AAERR6: MOV     #AAPAT5,@#$TMP3 ;FD=0 AND
 4365  022756  012737  023050  001242          MOV     #AAPAT0,@#$TMP4 ;DATA ERROR
 4366  022764  012737  023040  001244          MOV     #AADAT0,@#$TMP5
 4367  022772  012737  023130  001246          MOV     #AAPAT6,@#$TMP6
 4368  023000  104002                  1$:     ERROR   2
 4369  023002  000456                          BR      AADONE
 4370  023004  012737  023120  001240  AAERR7: MOV     #AAPAT5,@#$TMP3 ;CONSTANT ERROR
 4371  023012  012737  023050  001242          MOV     #AAPAT0,@#$TMP4
 4372  023020  012737  023040  001244          MOV     #AADAT0,@#$TMP5
 4373  023026  012737  023130  001246          MOV     #AAPAT6,@#$TMP6
 4374  023034  104002                  1$:     ERROR   2
 4375  023036  000440                          BR      AADONE
 4376  023040  000000                  AADAT0:         0
 4377  023042  000000                                  0
 4378  023044  000000                                  0
```

```
 4379  023046  000000                          0
 4380  023050  000200                AAPAT0:   200
 4381  023052  000000                          0
 4382  023054  000000                          0
 4383  023056  000000                          0
 4384  023060  000200                AAPAT1:   200
 4385  023062  000000                          0
 4386  023064  000000                          0
 4387  023066  000001                          1
 4388  023070  000400                AAPAT2:   400
 4389  023072  000000                          0
 4390  023074  000000                          0
 4391  023076  000000                          0
 4392  023100  000400                AAPAT3:   400
 4393  023102  000000                          0
 4394  023104  000000                          0
 4395  023106  000001                          1
 4396  023110  000400                AAPAT4:   400
 4397  023112  000000                          0
 4398  023114  100000                          100000
 4399  023116  000000                          0
 4400  023120  000200                AAPAT5:   200
 4401  023122  000001                          1
 4402  023124  000000                          0
 4403  023126  000000                          0
 4404  023130  000400                AAPAT6:   400
 4405  023132  000001                          1
 4406  023134  000000                          0
 4407  023136  000000                          0
 4408  023140                        AADONE:
 4409  023140  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 4410                                                                ;SEE IF THE USER HAS EXPRESSED
 4411                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
 4412                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 4413                                                                ;THE USER TYPED CONTROL G?).
 4414                                ;;******************************************************************
 4415                                ;*TEST 32       ADDF AND ADDD WITH E(AC) LESS THAN E(FSRC) TEST
 4416                                ;*
 4417                                ;*THIS IS ATEST OF THE ADDD AND ADDF
 4418                                ;*INSTRUCTIONS AND THE ALIGN AC ALGORITHM
 4419                                ;*FLOWS.  THE CONSTANT (25 FOR FLOATING, 57 FOR
 4420                                ;*DOUBLE) USED IS CHECKED.  THEN SIMPLE
 4421                                ;*AND WORST CASE ALIGNMENT SITUATIONS ARE
 4422                                ;*TRIED.  NOTE E(AC) IS LESS THEN E(FSRC)
 4423                                ;*
 4424                                ;;******************************************************************
 4425  023142  000004                TST32:  SCOPE
 4426                                ;EXPONENT DIFFERENCE=57=71 (OCT) FD=1
 4427  023144                        CC1:
 4428  023144  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4429  023146  012704  003200                MOV     #3200,R4        ;SET FIV,FIV, AND FD
 4430  023152  170104                        LDFPS   R4
 4431  023154  012737  023174  001236        MOV     #CC2,@#$TMP2
 4432  023162  012700  024576                MOV     #CCP0,R0        ;SET AC0 OPERAND
 4433  023166  172410                        LDD     (R0),AC0                ;AC0
 4434  023170  012700  024616                MOV     #CCP2,R0
```

```
4435  023174  172010                        CC2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
4436  023176  170205                                STFPS   R5              ;GET FPS
4437  023200  012700  024566                        MOV     #CCDATO,R0      ;GET THE RESULT
4438  023204  174010                                STD     AC0,(R0)
4439  023206  012701  024616                        MOV     #CCP2,R1        ;IS IT CORRECT
4440  023212  012702  000004                        MOV     #4,R2
4441  023216  022021                        CC3:    CMP     (R0)+,(R1)+
4442  023220  001415                                BEQ     CC6
4443  023222  012700  024566                        MOV     #CCDATO,R0      ;DID A BAD
4444  023226  012701  024576                        MOV     #CCP0,R1        ;CONSTANT (NOT 57)
4445  023232  012702  000004                        MOV     #4,R2           ;GET GENERATED
4446  023236  022021                        CC4:    CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
4447  023240  001402                                BEQ     CC5             ;FLOWS?
4448  023242  000137  024164                        JMP     @#CCER1         ;DATA ERROR.D
4449  023246  077205                        CC5:    SOB     R2,CC4
4450  023250  000137  024222                        JMP     @#CCER2         ;BAD CONSTANT.D
4451  023254  077220                        CC6:    SOB     R2,CC3
4452  023256  020405                                CMP     R4,R5           ;FPS CORRECT?
4453  023260  001402                                BEQ     CC7
4454  023262  000137  024130                        JMP     @#CCER0         ;BAD FPS.
4455                                        ;EXPONENT DIFFERENCE=56=70 (OCT) FD=1
4456  023266                                CC7:
4457  023266  104414                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4458  023270  012704  003200                        MOV     #3200,R4        ;SET FIV,FIV, AND FD
4459  023274  170104                                LDFPS   R4
4460  023276  012737  023316  001236                MOV     #CC8,@#$TMP2
4461  023304  012700  024576                        MOV     #CCP0,R0        ;SET AC0 OPERAND
4462  023310  172410                                LDD     (R0),AC0
4463  023312  012700  024606                        MOV     #CCP1,R0        ;FSRC
4464  023316  172010                        CC8:    ADDD    (R0),AC0        ;TEST INSTRUCTION
4465  023320  170205                                STFPS   R5              ;GET FPS
4466  023322  012700  024566                        MOV     #CCDATO,R0      ;GET THE RESULT
4467  023326  174010                                STD     AC0,(R0)
4468  023330  012701  024666                        MOV     #CCP7,R1        ;IS IT CORRECT
4469  023334  012702  000004                        MOV     #4,R2
4470  023340  022021                        CC9:    CMP     (R0)+,(R1)+
4471  023342  001415                                BEQ     CC12
4472  023344  012700  024566                        MOV     #CCDATO,R0      ;DID A BAD
4473  023350  012701  024606                        MOV     #CCP1,R1        ;CONSTANT (NOT 57)
4474  023354  012702  000004                        MOV     #4,R2           ;GET GENERATED
4475  023360  022021                        CC10:   CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
4476  023362  001402                                BEQ     CC11            ;FLOWS?
4477  023364  000137  024260                        JMP     @#CCER3         ;DATA ERROR.D
4478  023370  077205                        CC11:   SOB     R2,CC10
4479  023372  000137  024276                        JMP     @#CCER4         ;BAD CONSTANT.D
4480  023376  077220                        CC12:   SOB     R2,CC9
4481  023400  020405                                CMP     R4,R5           ;FPS CORRECT?
4482  023402  001402                                BEQ     CC13
4483  023404  000137  024130                        JMP     @#CCER0         ;BAD FPS.
4484                                        ;EXPONENT DIFFERENCE=25=31 (OCT) FD=0
4485  023410                                CC13:
4486  023410  104414                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4487  023412  012737  023440  001236                MOV     #CC14,@#$TMP2
4488  023420  012700  024576                        MOV     #CCP0,R0        ;SET UP ACO OPERAND.
4489  023424  172410                                LDD     (R0),AC0
4490  023426  012704  003000                        MOV     #3000,R4        ;SET FIV,FIV. CLEAR FD.
```

```
4491  023432  170104                        LDFPS   R4
4492  023434  012700  024656                MOV     #CCP6,R0        ;FSRC
4493  023440  172010                CC14:    ADDF    (R0),AC0        ;TEST INSTRUCTION
4494  023442  170205                        STFPS   R5
4495  023444  170011                        SETD                    ;REENTER DOUBLE MOVE
4496  023446  012700  024566                MOV     #CCDATO,R0      ;GET THE RESULT
4497  023452  174010                        STD     AC0,(R0)
4498  023454  012701  024656                MOV     #CCP6,R1        ;IS THE RESULT CORRECT?
4499  023460  012702  000002                MOV     #2,R2
4500  023464  022021                CC15:    CMP     (R0)+,(R1)+
4501  023466  001415                        BEQ     CC18
4502  023470  012700  024566                MOV     #CCDATO,R0      ;WAS A BAD CONSTANT
4503  023474  012701  024626                MOV     #CCP3,R1        ;USED (NOT 25) IN
4504  023500  012702  000002                MOV     #2,R2           ;THE ALIGN FLOWS?
4505  023504  022021                CC16:    CMP     (R0)+,(R1)+
4506  023506  001402                        BEQ     CC17
4507  023510  000137  024334                JMP     @#CCER5         ;DATA ERROR F
4508  023514  077205                CC17:    SOB     R2,CC16
4509  023516  000137  024370                JMP     @#CCER6         ;BAD CONSTANT F
4510  023522  077220                CC18:    SOB     R2,CC15
4511  023524  020405                        CMP     R4,R5
4512  023526  001402                        BEQ     CC19
4513  023530  000137  024146                JMP     @#CCER90        ;BAD FPS.
4514                                ;EXPONENT DIFFERENCE=24=30 (OCT) FD=0
4515  023534                        CC19:
4516  023534  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4517  023536  012737  023564  001236        MOV     #CC20,@#$TMP2
4518  023544  012700  024626                MOV     #CCP3,R0        ;SET UP AC0 OPERAND.
4519  023550  172410                        LDD     (R0),AC0
4520  023552  012704  003000                MOV     #3000,R4        ;SET FIV,FIV. CLEAR FD.
4521  023556  170104                        LDFPS   R4
4522  023560  012700  024646                MOV     #CCP5,R0        ;FSRC
4523  023564  172010                CC20:    ADDF    (R0),AC0        ;TEST INSTRUCTION
4524  023566  170205                        STFPS   R5
4525  023570  170011                        SETD                    ;REENTER DOUBLE MOVE
4526  023572  012700  024566                MOV     #CCDATO,R0      ;GET THE RESULT
4527  023576  174010                        STD     AC0,(R0)
4528  023600  012701  024676                MOV     #CCP10,R1       ;IS THE RESLT CORRECT?
4529  023604  012702  000002                MOV     #2,R2
4530  023610  022021                CC21:    CMP     (R0)+,(R1)+
4531  023612  001415                        BEQ     CC24
4532  023614  012700  024566                MOV     #CCDATO,R0      ;WAS A BAD CONSTANT
4533  023620  012701  024646                MOV     #CCP5,R1        ;USED (NOT 25) IN
4534  023624  012702  000002                MOV     #2,R2           ;THE ALIGN FLOWS?
4535  023630  022021                CC22:    CMP     (R0)+,(R1)+
4536  023632  001402                        BEQ     CC23
4537  023634  000137  024424                JMP     @#CCER7         ;DATA ERROR F
4538  023640  077205                CC23:    SOB     R2,CC22
4539  023642  000137  024442                JMP     @#CCER8 ;BAD CONSTANT F
4540  023646  077220                CC24:    SOB     R2,CC21
4541  023650  020405                        CMP     R4,R5
4542  023652  001402                        BEQ     CC25
4543  023654  000137  024146                JMP     @#CCER90        ;BAD FPS.
4544                                ;EXPONENT DIFFERENCE=1 FD=1
4545  023660                        CC25:
4546  023660  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
4547  023662  012704  003200                  MOV     #3200,R4        ;SET FIV,FIV, AND FD
4548  023666  170104                          LDFPS   R4
4549  023670  012737  023710  001236          MOV     #CC26,@#$TMP2
4550  023676  012700  024576                  MOV     #CCP0,R0        ;SET AC0 OPERAND
4551  023702  172410                          LDD     (R0),AC0
4552  023704  012700  024626                  MOV     #CCP3,R0        ;FSRC
4553  023710  172010                  CC26:   ADDD    (R0),AC0        ;TEST INSTRUCTION
4554  023712  170205                          STFPS   R5              ;GET FPS
4555  023714  012700  024566                  MOV     #CCDAT0,R0      ;GET THE RESULT
4556  023720  174010                          STD     AC0,(R0)
4557  023722  012701  024706                  MOV     #CCP11,R1       ;IS IT CORRECT
4558  023726  012702  000004                  MOV     #4,R2
4559  023732  022021                  CC27:   CMP     (R0)+,(R1)+
4560  023734  001415                          BEQ     CC30
4561  023736  012700  024566                  MOV     #CCDAT0,R0      ;DID A BAD
4562  023742  012701  024626                  MOV     #CCP3,R1        ;CONSTANT (NOT 57)
4563  023746  012702  000004                  MOV     #4,R2           ;GET GENERATED
4564  023752  022021                  CC28:   CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
4565  023754  001402                          BEQ     CC29            ;FLOWS?
4566  023756  000137  024476                  JMP     @#CCER10        ;DATA ERROR.D
4567  023762  077205                  CC29:   SOB     R2,CC28
4568  023764  000137  024514                  JMP     @#CCER11        ;BAD CONSTANT.D
4569  023770  077220                  CC30:   SOB     R2,CC27
4570  023772  020405                          CMP     R4,R5           ;FPS CORRECT?
4571  023774  001402                          BEQ     CC31
4572  023776  000137  024130                  JMP     @#CCER0         ;BAD FPS.
4573                                  ;EXPONENT DIFFERENCE=100=144 (OCT) FD=1
4574  024002                          CC31:
4575  024002  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
4576  024004  012704  003200                  MOV     #3200,R4        ;SET FIV,FIV, AND FD
4577  024010  170104                          LDFPS   R4
4578  024012  012737  024032  001236          MOV     #CC32,@#$TMP2
4579  024020  012700  024576                  MOV     #CCP0,R0        ;SET AC0 OPERAND
4580  024024  172410                          LDD     (R0),AC0
4581  024026  012700  024636                  MOV     #CCP4,R0        ;FSRC
4582  024032  172010                  CC32:   ADDD    (R0),AC0        ;TEST INSTRUCTION
4583  024034  170205                          STFPS   R5              ;GET FPS
4584  024036  012700  024566                  MOV     #CCDAT0,R0      ;GET THE RESULT
4585  024042  174010                          STD     AC0,(R0)
4586  024044  012701  024636                  MOV     #CCP4,R1        ;IS IT CORRECT
4587  024050  012702  000004                  MOV     #4,R2
4588  024054  022021                  CC33:   CMP     (R0)+,(R1)+
4589  024056  001415                          BEQ     CC36
4590  024060  012700  024566                  MOV     #CCDAT0,R0      ;DID A BAD
4591  024064  012701  024636                  MOV     #CCP4,R1        ;CONSTANT (NOT 57)
4592  024070  012702  000004                  MOV     #4,R2           ;GET GENERATED
4593  024074  022021                  CC34:   CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
4594  024076  001402                          BEQ     CC35            ;FLOWS?
4595  024100  000137  024532                  JMP     @#CCER12        ;DATA ERROR.D
4596  024104  077205                  CC35:   SOB     R2,CC34
4597  024106  000137  024550                  JMP     @#CCER13        ;BAD CONSTANT.D
4598  024112  077220                  CC36:   SOB     R2,CC33
4599  024114  020405                          CMP     R4,R5           ;FPS CORRECT?
4600  024116  001402                          BEQ     CC37
4601  024120  000137  024130                  JMP     @#CCER0         ;BAD FPS.
4602  024124  000137  024726          CC37:   JMP     @#CCDONE
```

```
 4603  024130  010437  001242          CCER0:  MOV     R4,@#$TMP4      ;FPS ERROR D
 4604  024134  010537  001240                  MOV     R5,@#$TMP3
 4605  024140  104002                  1$:     ERROR   2
 4606  024142  000137  024726                  JMP     @#CCDONE
 4607  024146  010437  001242          CCER90: MOV     R4,@#$TMP4      ;FPS ERROR F
 4608  024152  010537  001240                  MOV     R5,@#$TMP3
 4609  024156  104002                  1$:     ERROR   2
 4610  024160  000137  024726                  JMP     @#CCDONE
 4611  024164  012737  024616  001240  CCER1:  MOV     #CCP2,@#$TMP3   ;DATA ERROR D
 4612  024172  012737  024616  001246          MOV     #CCP2,@#$TMP6
 4613  024200  012737  024576  001242  CCER50: MOV     #CCP0,@#$TMP4
 4614  024206  012737  024566  001244          MOV     #CCDATO,@#$TMP5
 4615  024214  104002                  1$:     ERROR   2
 4616  024216  000137  024726                  JMP     @#CCDONE
 4617  024222  012737  024616  001240  CCER2:  MOV     #CCP2,@#$TMP3   ;CONSTANT BAD D(B)
 4618  024230  012737  024616  001246          MOV     #CCP2,@#$TMP6
 4619  024236  012737  024576  001242  CCER22: MOV     #CCP0,@#$TMP4
 4620  024244  012737  024566  001244          MOV     #CCDATO,@#$TMP5
 4621  024252  104002                  1$:     ERROR   2
 4622  024254  000137  024726                  JMP     @#CCDONE
 4623  024260  012737  024606  001240  CCER3:  MOV     #CCP1,@#$TMP3
 4624  024266  012737  024666  001246          MOV     #CCP7,@#$TMP6
 4625  024274  000741                          BR      CCER50
 4626  024276  012737  024606  001240  CCER4:  MOV     #CCP1,@#$TMP3   ;CONSTANT BAD D(G)
 4627  024304  012737  024666  001246          MOV     #CCP7,@#$TMP6
 4628  024312  012737  024576  001242  CCER44: MOV     #CCP0,@#$TMP4
 4629  024320  012737  024566  001244          MOV     #CCDATO,@#$TMP5
 4630  024326  104002                  1$:     ERROR   2
 4631  024330  000137  024726                  JMP     @#CCDONE
 4632  024334  012737  024656  001240  CCER5:  MOV     #CCP6,@#$TMP3   ;DATA ERROR F
 4633  024342  012737  024656  001246          MOV     #CCP6,@#$TMP6
 4634  024350  012737  024576  001242  CCER55: MOV     #CCP0,@#$TMP4
 4635  024356  012737  024566  001244          MOV     #CCDATO,@#$TMP5
 4636  024364  104002                  1$:     ERROR   2
 4637  024366  000557                          BR      CCDONE
 4638  024370  012737  024656  001240  CCER6:  MOV     #CCP6,@#$TMP3   ;CONSTANT BAD F(B)
 4639  024376  012737  024656  001246          MOV     #CCP6,@#$TMP6
 4640  024404  012737  024576  001242          MOV     #CCP0,@#$TMP4
 4641  024412  012737  024566  001244          MOV     #CCDATO,@#$TMP5
 4642  024420  104002                  1$:     ERROR   2
 4643  024422  000541                          BR      CCDONE
 4644  024424  012737  024646  001240  CCER7:  MOV     #CCP5,@#$TMP3   ;DATA ERROR F
 4645  024432  012737  024676  001246          MOV     #CCP10,@#$TMP6
 4646  024440  000743                          BR      CCER55
 4647  024442  012737  024646  001240  CCER8:  MOV     #CCP5,@#$TMP3   ;CONSTANT BAD F(G)
 4648  024450  012737  024676  001246          MOV     #CCP10,@#$TMP6
 4649  024456  012737  024566  001244          MOV     #CCDATO,@#$TMP5
 4650  024464  012737  024576  001242          MOV     #CCP0,@#$TMP4
 4651  024472  104002                  1$:     ERROR   2
 4652  024474  000514                          BR      CCDONE
 4653  024476  012737  024626  001240  CCER10: MOV     #CCP3,@#$TMP3   ;DATA ERROR D
 4654  024504  012737  024706  001246          MOV     #CCP11,@#$TMP6
 4655  024512  000632                          BR      CCER50
 4656  024514  012737  024626  001240  CCER11: MOV     #CCP3,@#$TMP3   ;CONSTANT BAD D(G)
 4657  024522  012737  024706  001246          MOV     #CCP11,@#$TMP6
 4658  024530  000670                          BR      CCER44
```

```
 4659  024532  012737  024636  001240  CCER12: MOV     #CCP4,a#$TMP3   ;DATA ERROR D
 4660  024540  012737  024636  001246          MOV     #CCP4,a#$TMP6
 4661  024546  000614                  BR      CCER50
 4662  024550  012737  024636  001240  CCER13: MOV     #CCP4,a#$TMP3   ;CONSTANT BAD D(B)
 4663  024556  012737  024636  001246          MOV     #CCP4,a#$TMP6
 4664  024564  000624                  BR      CCER22
 4665  024566  000000                  CCDATO: 0
 4666  024570  000000                          0
 4667  024572  000000                          0
 4668  024574  000000                          0
 4669  024576  000200                  CCP0:   200                     ;E(AC)=1
 4670  024600  000000                          0
 4671  024602  000000                          0
 4672  024604  000000                          0
 4673  024606  016200                  CCP1:   16200                   ;E(FSRC)=E(AC)+56=57
 4674  024610  000000                          0                       ;                 =71(OCT)
 4675  024612  000000                          0
 4676  024614  000000                          0
 4677  024616  016400                  CCP2:   16400                   ;E(FSRC)=E(AC)+57=58
 4678  024620  000000                          0                       ;                 =72(OCT)
 4679  024622  000000                          0
 4680  024624  000000                          0
 4681  024626  000400                  CCP3:   400                     ;E(FSRC)=E(AC)+1=2
 4682  024630  000000                          0
 4683  024632  000000                          0
 4684  024634  000000                          0
 4685  024636  031200                  CCP4:   31200                   ;E(FSRC)=E(AC)+100=101=145(OCT)
 4686  024640  000000                          0
 4687  024642  000000                          0
 4688  024644  000000                          0
 4689  024646  006200                  CCP5:   6200                    ;E(FSRC)=E(AC)+24=25=31(OCT)
 4690  024650  000000                          0
 4691  024652  000000                          0
 4692  024654  000000                          0
 4693  024656  006400                  CCP6:   6400                    ;E(FSRC)=E(AC)+25=26=32(OCT)
 4694  024660  000000                          0
 4695  024662  000000                          0
 4696  024664  000000                          0
 4697  024666  016200                  CCP7:   16200                   ;CCP1 RES
 4698  024670  000000                          0
 4699  024672  000000                          0
 4700  024674  000001                          1
 4701  024676  006200                  CCP10:  6200                    ;CCP5 RES
 4702  024700  000001                          1
 4703  024702  000000                          0
 4704  024704  000000                          0
 4705  024706  000500                  CCP11:  500                     ;CCP3 RES
 4706  024710  000000                          0
 4707  024712  000000                          0
 4708  024714  000000                          0
 4709  024716  000200                  CCP12:  200                     ;BAD CONSTANT
 4710  024720  000000                          0                       ;RES CCP2,CCP4
 4711  024722  000000                          0
 4712  024724  000000                          0
 4713
 4714  024726                          CCDONE:
```

```
                                        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 33        ADDF AND ADDD WITH E(AC) GREATER THAN E(FSRC) TEST
                                ;*
                                ;*THIS IS A TEST OF THE ADDD AND ADDF
                                ;*INSTRUCTIONS AND THE ALIGN FSRC ALGORITHM
                                ;*FLOWS.  FIRST THE CONSTANT USED IS CHECKED.
                                ;*THEN SIMPLE AND WORST CASE ALIGNMENT
                                ;*SITUATIONS ARE TRIED.  NOTE E(AC)
                                ;*IS GREATER THAN E(FSRC).
                                ;*
                                ;;**************************************************************
024730  000004                  TST33:  SCOPE
                                ;EXPONENT DIFFERENCE=57=71 (OCT) FD=1
024732                          BB1:
024732  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
024734  012704  003200                  MOV     #3200,R4        ;SET FIV FIV, AND FD
024740  170104                          LDFPS   R4
024742  012737  025570  000244          MOV     #BBER0,@#FPVECT ;SET UP FOR ERROR
024750  012737  024770  001236          MOV     #BB2,@#$TMP2    ;IN CASE THE OVER\
                                                                ;UNDER FLOWS FAIL.
024756  012700  026132                  MOV     #BBPAT2,R0      ;SET AC0 OPERAND.
024762  172410                          LDD     (R0),AC0
024764  012700  026122                  MOV     #BBPAT1,R0      ;FSRC
024770  172010                  BB2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
024772  170205                          STFPS   R5
024774  012700  026102          BB3:    MOV     #BBDAT0,R0      ;GET THE RESULT
025000  174010                          STD     AC0,(R0)
025002  012701  026132                  MOV     #BBPAT2,R1      ;RESULT CORRECT?
025006  012702  000004                  MOV     #4,R2
025012  022021                  BB4:    CMP     (R0)+,(R1)+
025014  001402                          BEQ     BB5
025016  000137  025630                  JMP     @#BBER1         ;DATA ERROR D
025022  077205                  BB5:    SOB     R2,BB4
                                                                ;WAS FPS CORRECT?
025024  020405                          CMP     R4,R5
025026  001402                          BEQ     BB6
025030  000137  025570                  JMP     @#BBER0         ;FPS ERROR
                                ;EXPONENT DIFFERENCE=56=70 (OCT) FD=1
025034                          BB6:
025034  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
025036  012704  003200                  MOV     #3200,R4        ;SET FIV,FIV, AND FD
025042  170104                          LDFPS   R4
025044  012737  025064  001236          MOV     #BB7,@#$TMP2
025052  012700  026152                  MOV     #BBPAT4,R0      ;SET AC0 OPERAND
025056  172410                          LDD     (R0),AC0
025060  012700  026122                  MOV     #BBPAT1,R0      ;FSRC
025064  172010                  BB7:    ADDD    (R0),AC0        ;TEST INSTRUCTION
025066  170205                          STFPS   R5              ;GET FPS
025070  012700  026102                  MOV     #BBDAT0,R0      ;GET THE RESULT
025074  174010                          STD     AC0,(R0)
```

```
 4771  025076  012701  026212                 MOV     #BBP10,R1       ;IS IT CORRECT
 4772  025102  012702  000004                 MOV     #4,R2
 4773  025106  022021                 BB10:   CMP     (R0)+,(R1)+
 4774  025110  001415                         BEQ     BB13
 4775  025112  012700  026102                 MOV     #BBDAT0,R0      ;DID A BAD
 4776  025116  012701  026152                 MOV     #BBPAT4,R1      ;CONSTANT (NOT 57)
 4777  025122  012702  000004                 MOV     #4,R2           ;GET GENERATED
 4778  025126  022021                 BB11:   CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
 4779  025130  001402                         BEQ     BB12            ;FLOWS?
 4780  025132  000137  025666                 JMP     @#BBER2         ;DATA ERROR.D
 4781  025136  077205                 BB12:   SOB     R2,BB11
 4782  025140  000137  025704                 JMP     @#BBER3         ;BAD CONSTANT.D
 4783  025144  077220                 BB13:   SOB     R2,BB10
 4784  025146  020405                         CMP     R4,R5           ;FPS CORRECT?
 4785  025150  001402                         BEQ     BB14
 4786  025152  000137  025570                 JMP     @#BBER0         ;BAD FPS.
 4787                                 ;EXPONENT DIFFERENCE=25=31 (OCT) FD=0
 4788  025156                         BB14:
 4789  025156  104414                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4790  025160  012737  025206  001236         MOV     #BB15,@#$TMP2
 4791  025166  012700  026112                 MOV     #BBPAT0,R0      ;SET UP AC0 OPERAND
 4792  025172  172410                         LDD     (R0),AC0
 4793  025174  012704  003000                 MOV     #3000,R4        ;SET FIV AND FIV
 4794                                                                 ;CLEAR FD
 4795  025200  170104                         LDFPS   R4
 4796  025202  012700  026122                 MOV     #BBPAT1,R0      ;FSRC
 4797  025206  172010                 BB15:   ADDF    (R0),AC0        ;TEST INSTRUCTION
 4798  025210  170205                         STFPS   R5
 4799  025212  170011                         SETD                    ;REENTED DOUBLE MODE.
 4800  025214  012700  026102                 MOV     #BBDAT0,R0      ;GET THE RESULT
 4801  025220  174010                         STD     AC0,(R0)
 4802  025222  012701  026112                 MOV     #BBPAT0,R1      ;IS THE RESULT
 4803  025226  012702  000002                 MOV     #2,R2           ;CORRECT?
 4804  025232  022021                 BB16:   CMP     (R0)+,(R1)+
 4805  025234  001402                         BEQ     BB17
 4806  025236  000137  025740                 JMP     @#BBER4         ;DATA ERROR F
 4807  025242  077205                 BB17:   SOB     R2,BB16
 4808  025244  020405                         CMP     R4,R5           ;IS FPS CORRECT?
 4809  025246  001402                         BEQ     BB20
 4810  025250  000137  025610                 JMP     @#BBER10        ;FPS ERROR.
 4811                                 ;EXPONENT DIFFERENCE=24=30 (OCT)
 4812  025254                         BB20:
 4813  025254  104414                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4814  025256  012737  025304  001236         MOV     #BB21,@#$TMP2
 4815  025264  012700  026142                 MOV     #BBPAT3,R0      ;SET UP AC0 OPERAND.
 4816  025270  172410                         LDD     (R0),AC0
 4817  025272  012704  003000                 MOV     #3000,R4        ;SET FIU,FIV. CLEAR FD.
 4818  025276  170104                         LDFPS   R4
 4819  025300  012700  026122                 MOV     #BBPAT1,R0      ;FSRC
 4820  025304  172010                 BB21:   ADDF    (R0),AC0        ;TEST INSTRUCTION
 4821  025306  170205                         STFPS   R5
 4822  025310  170011                         SETD                    ;REENTER DOUBLE MODE
 4823  025312  012700  026102                 MOV     #BBDAT0,R0      ;GET THE RESULT
 4824  025316  174010                         STD     AC0,(R0)
 4825  025320  012701  026202                 MOV     #BBP7,R1        ;IS THE RESULT CORRECT?
 4826  025324  012702  000002                 MOV     #2,R2
```

```
 4827  025330  022021                          BB22:   CMP     (R0)+,(R1)+
 4828  025332  001415                                  BEQ     BB25
 4829  025334  012700  026102                        MOV     #BBDAT0,R0      ;WAS A BAD CONSTANT
 4830  025340  012701  026142                        MOV     #BBPAT3,R1      ;USED (NOT 25) IN
 4831  025344  012702  000002                        MOV     #2,R2           ;THE ALLIGN FLOWS?
 4832  025350  022021                          BB23:   CMP     (R0)+,(R1)+
 4833  025352  001402                                  BEQ     BB24
 4834  025354  000137  025774                        JMP     @#BBER5         ;DATA ERROR F
 4835  025360  077205                          BB24:   SOB     R2,BB23
 4836  025362  000137  026012                        JMP     @#BBER6         ;BAD CONSTANT F
 4837  025366  077220                          BB25:   SOB     R2,BB22
 4838  025370  020405                                  CMP     R4,R5
 4839  025372  001402                                  BEQ     BB26
 4840  025374  000137  025610                        JMP     @#BBER10        ;BAD FPS.
 4841                                          ;EXPONENT DIFFERENCE=1
 4842  025400                                  BB26:
 4843  025400  104414                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4844  025402  012737  025430  001236                MOV     #BB27,@#$TMP2
 4845  025410  012704  003200                        MOV     #3200,R4
 4846  025414  170104                                LDFPS   R4              ;SET UP AC0 OPERAND
 4847  025416  012700  026162                        MOV     #BBPAT5,R0
 4848  025422  172410                                LDD     (R0),AC0
 4849  025424  012700  026122                        MOV     #BBPAT1,R0      ;FSRC
 4850  025430  172010                          BB27:   ADDD    (R0),AC0        ;TEST INSTRUCTION
 4851  025432  170205                                STFPS   R5
 4852  025434  012700  026102                        MOV     #BBDAT0,R0      ;GET THE RESULT.
 4853  025440  174010                                STD     AC0,(R0)
 4854  025442  012701  026222                        MOV     #BBP11,R1       ;IS IT CORRECT?
 4855  025446  012702  000004                        MOV     #4,R2
 4856  025452  022021                          BB30:   CMP     (R0)+,(R1)+
 4857  025454  001402                                  BEQ     BB31
 4858  025456  000137  026046                        JMP     @#BBER7         ;DATA ERROR D
 4859  025462  077205                          BB31:   SOB     R2,BB30
 4860  025464  020405                                  CMP     R4,R5           ;IS FPS CORRECT
 4861  025466  001402                                  BEQ     BB32
 4862  025470  000137  025570                        JMP     @#BBER0
 4863                                          ;EXPONENT DIFFERENCE=100=144 (OCT)
 4864  025474                                  BB32:
 4865  025474  104414                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 4866  025476  012737  025524  001236                MOV     #BB33,@#$TMP2
 4867  025504  012704  003200                        MOV     #3200,R4
 4868  025510  170104                                LDFPS   R4              ;SET FIV,FIV AND FD
 4869  025512  012700  026172                        MOV     #BBPAT6,R0      ;SET UP AC0 OPERAND.
 4870  025516  172410                                LDD     (R0),AC0
 4871  025520  012700  026122                        MOV     #BBPAT1,R0      ;FSRC
 4872  025524  172010                          BB33:   ADDD    (R0),AC0        ;TEST INSTRUCTION
 4873  025526  170205                                STFPS   R5
 4874  025530  012700  026102                        MOV     #BBDAT0,R0      ;GET THE RESULT
```

```
 4875  025534  174010                          STD     AC0,(R0)
 4876  025536  012701  026172                  MOV     #BBPAT6,R1      ;IS IT CORRECT
 4877  025542  012702  000004                  MOV     #4,R2
 4878  025546  022021                  BB34:   CMP     (R0)+,(R1)+
 4879  025550  001402                          BEQ     BB35
 4880  025552  000137  026064                  JMP     @#BBER8         ;DATA ERROR D
 4881  025556  077205                  BB35:   SOB     R2,BB34
 4882  025560  020405                          CMP     R4,R5           ;IS FPS CORRECT
 4883  025562  001002                          BNE     BBER0
 4884  025564  000167  000442                  JMP     BBDONE
 4885  025570  010437  001242          BBER0:  MOV     R4,@#$TMP4      ;FPS ERROR D
 4886  025574  010537  001240                  MOV     R5,@#$TMP3
 4887  025600  104001                  1$:     ERROR   1
 4888  025602  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 4889                                          ;SEE IF THE USER HAS EXPRESSED
 4890                                          ;THE DESIRE TO CHANGE THE SOFTWARE
 4891                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 4892                                          ;THE USER TYPED CONTROL G?).
 4893  025604  000137  026232                  JMP     @#BBDONE
 4894  025610  010437  001242          BBER10: MOV     R4,@#$TMP4      ;FPS ERROR F
 4895  025614  010537  001240                  MOV     R5,@#$TMP3
 4896  025620  104002                  1$:     ERROR   2
 4897  025622  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 4898                                          ;SEE IF THE USER HAS EXPRESSED
 4899                                          ;THE DESIRE TO CHANGE THE SOFTWARE
 4900                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 4901                                          ;THE USER TYPED CONTROL G?).
 4902  025624  000137  026232                  JMP     @#BBDONE
 4903  025630  012737  026132  001242  BBER1:  MOV     #BBPAT2,@#$TMP4 ;DATA ERROR D
 4904  025636  012737  026132  001246          MOV     #BBPAT2,@#$TMP6
 4905  025644  012737  026122  001240  BBER11: MOV     #BBPAT1,@#$TMP3
 4906  025652  012737  026102  001244          MOV     #BBDAT0,@#$TMP5
 4907  025660  104002                  1$:     ERROR   2
 4908  025662  000137  026232                  JMP     @#BBDONE
 4909  025666  012737  026152  001242  BBER2:  MOV     #BBPAT4,@#$TMP4
 4910  025674  012737  026212  001246          MOV     #BBP10,@#$TMP6
 4911  025702  000760                          BR      BBER11
 4912  025704  012737  026152  001242  BBER3:  MOV     #BBPAT4,@#$TMP4 ;BAD CONSTANT D
 4913  025712  012737  026212  001246          MOV     #BBP10,@#$TMP6
 4914  025720  012737  026122  001240          MOV     #BBPAT1,@#$TMP3
 4915  025726  012737  026102  001244          MOV     #BBDAT0,@#$TMP5
 4916  025734  104002                  1$:     ERROR   2
 4917  025736  000535                          BR      BBDONE
 4918  025740  012737  026112  001242  BBER4:  MOV     #BBPAT0,@#$TMP4 ;DATA ERROR F
 4919  025746  012737  026112  001246          MOV     #BBPAT0,@#$TMP6
 4920  025754  012737  026122  001240  BBER40: MOV     #BBPAT1,@#$TMP3
 4921  025762  012737  026102  001244          MOV     #BBDAT0,@#$TMP5
 4922  025770  104002                  1$:     ERROR   2
 4923  025772  000517                          BR      BBDONE
 4924  025774  012737  026142  001242  BBER5:  MOV     #BBPAT3,@#$TMP4
 4925  026002  012737  026202  001246          MOV     #BBP7,@#$TMP6
 4926  026010  000761                          BR      BBER40
 4927  026012  012737  026142  001242  BBER6:  MOV     #BBPAT3,@#$TMP4 ;CONSTANT ERROR F
 4928  026020  012737  026202  001246          MOV     #BBP7,@#$TMP6
 4929  026026  012737  026122  001240          MOV     #BBPAT1,@#$TMP3
 4930  026034  012737  026102  001244          MOV     #BBDAT0,@#$TMP5
```

```
 4931  026042  104002                  1$:     ERROR   2
 4932  026044  000472                          BR      BBDONE
 4933  026046  012737  026162  001242  BBER7:  MOV     #BBPAT5,a#$TMP4
 4934  026054  012737  026122  001246          MOV     #BBPAT11,a#$TMP6
 4935  026062  000670                          BR      BBER11
 4936  026064  012737  026172  001242  BBER8:  MOV     #BBPAT6,a#$TMP4
 4937  026072  012737  026172  001246          MOV     #BBPAT6,a#$TMP6
 4938  026100  000661                          BR      BBER11
 4939  026102  000000                  BBDAT0: 0
 4940  026104  000000                          0
 4941  026106  000000                          0
 4942  026110  000000                          0
 4943  026112  006400                  BBPAT0: 6400                    ;F(AC)=E(FSRC)+25=26
 4944  026114  000000                          0                       ;                 =32(OCT)
 4945  026116  000000                          0
 4946  026120  000000                          0
 4947  026122  000200                  BBPAT1: 200
 4948  026124  000000                          0                       ;E(FSRC)=1
 4949  026126  000000                          0
 4950  026130  000000                          0
 4951  026132  016400                  BBPAT2: 16400
 4952  026134  000000                          0                       ;E(AC)=E(FSRC)+57=58
 4953  026136  000000                          0                       ;                 =72(OCT)
 4954  026140  000000                          0
 4955  026142  006200                  BBPAT3: 6200                    ;E(AC)=E(FSRC)+24=25
 4956  026144  000000                          0                       ;                 =31(OCT)
 4957  026146  000000                          0
 4958  026150  000000                          0
 4959  026152  016200                  BBPAT4: 16200                   ;E(AC)=E(FSR6)+56=57
 4960  026154  000000                          0                       ;                 =71(OCT)
 4961  026156  000000                          0
 4962  026160  000000                          0
 4963  026162  000400                  BBPAT5: 400                     ;E(AC)=E(FSRC)+1=2
 4964  026164  000000                          0
 4965  026166  000000                          0
 4966  026170  000000                          0
 4967  026172  031200                  BBPAT6: 31200                   ;E(AC)=E(FSRC)+100=101
 4968  026174  000000                          0                       ;                 =145(OCT)
 4969  026176  000000                          0
 4970  026200  000000                          0
 4971  026202  006200                  BBP7:   6200                    ;BBPAT3 RES
 4972  026204  000001                          1
 4973  026206  000000                          0
 4974  026210  000000                          0
 4975  026212  016200                  BBP10:  16200                   ;BBPAT4 RES
 4976  026214  000000                          0
 4977  026216  000000                          0
 4978  026220  000001                          1
 4979  026222  000500                  BBP11:  500                     ;BBPAT5 RES
 4980  026224  000000                          0
 4981  026226  000000                          0
 4982  026230  000000                          0
 4983  026232                          BBDONE:
 4984  026232  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 4985                                                                  ;SEE IF THE USER HAS EXPRESSED
 4986                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
```

```
                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                        ;THE USER TYPED CONTROL G?).
                                ;;*****************************************************************
                                ;*TEST 34        ADDD WITH NEGATVE OPRANDS TEST
                                ;*
                                ;*THIS IS A TEST OF THE ADDD INSTRUCTION
                                ;*WITH NEGATIVE OPERANDS.  EVERY COMBINATION OF
                                ;*OPERAND SIGNS IS TRIED.
                                ;*
                                ;;*****************************************************************
026234  000004                  TST34:  SCOPE
                                ;BOTH OPERANDS NEGATIVE
026236                          DD1:
026236  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
026240  012704  003200                  MOV     #3200,R4        ;SET FIO, FIV, AND FD
026244  170104                          LDFPS   R4
026246  012737  026266  001236          MOV     #DD2,@#$TMP2
026254  012700  030116                  MOV     #DDP1,R0                ;SET AC0 OPERAND
026260  172410                          LDD     (R0),AC0
026262  012700  030116                  MOV     #DDP1,R0                ;ESRC
026266  172010                  DD2:    ADDD    (R0),AC0                ;TEST INSTRUCTION
026270  170205                          STFPS   R5                      ;GET FPS
026272  012700  030076                  MOV     #DDDAT0,R0              ;GET THE RESULT
026276  174010                          STD     AC0,(R0)
026300  012701  030216                  MOV     #DDP9,R1                ;IS IT CORRECT
026304  012702  000004                  MOV     #4,R2
026310  022021                  DD3:    CMP     (R0)+,(R1)+
026312  001415                          BEQ     DD6
026314  012700  030076                  MOV     #DDDAT0,R0              ;DID A ADD-SUB
026320  012701  030146                  MOV     #DDP4,R1                ;FLOW A FAILURE
026324  012702  000004                  MOV     #4,R2
026330  022021                  DD4:    CMP     (R0)+,(R1)+
026332  001402                          BEQ     DD5                     ;216,442,500
026334  000137  027326                  JMP     @#DDER1                 ;DATA ERROR,D
026340  077205                  DD5:    SOB     R2,DD4
026342  000137  027364                  JMP     @#DDER2                 ;FLOW FAILURE,D
026346  077220                  DD6:    SOB     R2,DD3
026350  052704  000010                  BIS     #10,R4
026354  020405                          CMP     R4,R5                   ;FPS CORRECT?
026356  001402                          BEQ     DD7
026360  000137  027310                  JMP     @#DDER0                 ;BAD,FPS
                                ;AC POS FSRC NEG        AC=-FSRC
026364                          DD7:
026364  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
026366  012704  003200                  MOV     #3200,R4                ;SET FIO, FIV, AND FD
026372  170104                          LDFPS   R4
026374  012737  026414  001236          MOV     #DD8,@#$TMP2
026402  012700  030126                  MOV     #DDP2,R0                ;SET AC0 OPERAND
026406  172410                          LDD     (R0),AC0
026410  012700  030116                  MOV     #DDP1,R0                ;FSPC
026414  172010                  DD8:    ADDD    (R0),AC0                ;TEST INSTRUCTION
026416  170205                          STFPS   R5                      ;GET FPS
026420  012700  030076                  MOV     #DDDAT0,R0              ;GET THE RESULT
026424  174010                          STD     AC0,(R0)
026426  012701  030106                  MOV     #DDP0,R1                ;IS IT CORRECT
026432  012702  000004                  MOV     #4,R2
```

```
5043  026436  022021                DD10:   CMP     (R0)+,(R1)+
5044  026440  001402                        BEQ     DD11
5045  026442  000137  027422                JMP     @#DDER3                 ;FLOW FAILURE
5046  026446  077205                DD11:   SOB     R2,DD10
5047  026450  052704  000004              BIS     #4,R4
5048  026454  020405                        CMP     R4,R5                   ;FPS CORRECT?
5049  026456  001402                        BEQ     DD12
5050  026460  000137  027310                JMP     @#DDER0                 ;BAD FPS
5051                                ;AC NEG FSRC POS                AC=-FSRC
5052  026464                        DD12:
5053  026464  104414                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
5054  026466  012704  003200              MOV     #3200,R4                ;SET FIU, FIV, AND FD
5055  026472  170104                        LDFPS   R4
5056  026474  012737  026514  001236        MOV     #DD13,@#$TMP2
5057  026502  012700  030116              MOV     #DDP1,R0                ;SET AC0 OPERAND
5058  026506  172410                        LDD     (R0),AC0
5059  026510  012700  030126              MOV     #DDP2,R0                ;FSRC
5060  026514  172010                DD13:   ADDD    (R0),AC0                ;TEST INSTRUCTION
5061  026516  170205                        STFPS   R5                      ;GET FPS
5062  026520  012700  030076              MOV     #DDDAT0,R0              ;GET THE RESULT
5063  026524  174010                        STD     AC0,(R0)
5064  026526  012701  030106              MOV     #DDP0,R1                ;IS IT CORRECT
5065  026532  012702  000004              MOV     #4,R2
5066  026536  022021                DD14:   CMP     (R0)+,(R1)+
5067  026540  001402                        BEQ     DD15
5068  026542  000137  027460              JMP     @#DDER4                 ;FLOW FAILURE 216,440,121
5069  026546  077205                DD15:   SOB     R2,DD14
5070  026550  052704  000004              BIS     #4,R4
5071  026554  020405                        CMP     R4,R5                   ;EPS CORRECT?
5072  026556  001402                        BEQ     DD16
5073  026560  000137  027310                JMP     @#DDER0                 ;BAD FPS
5074                                ;AC0 POC        FSRC    NEG     /AC/ > /FSRC/
5075  026564                        DD16:
5076  026564  104414                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
5077  026566  012704  003200              MOV     #3200,R4                ;SET FIV, FIV AND FD
5078  026572  170104                        LDFPS   R4
5079  026574  012737  026614  001236        MOV     #DD17,@#$TMP2
5080  026602  012700  030136              MOV     #DDP3,R0                ;SET AC0 OPERAND
5081  026606  172410                        LDD     (R0),AC0
5082  026610  012700  030166              MOV     #DDP6,R0                ;ESPC
5083  026614  172010                DD17:   ADDD    (R0),AC0                ;TEST INSTRUCTION
5084  026616  170205                        STFPS   R5                      ;GET FPS
5085  026620  012700  030076              MOV     #DDDAT0,R0              ;GET THE RESULT
5086  026624  174010                        STD     AC0,(R0)
5087  026626  012701  030176              MOV     #DDP7,R1                ;IS IT CORRECT
5088  026632  012702  000004              MOV     #4,R2
5089  026636  022021                DD18:   CMP     (R0)+,(R1)+
5090  026640  001415                        BEQ     DD21
5091  026642  012700  030076              MOV     #DDDAT0,R0              ;FLOWS FAILURE
5092  026646  012701  030206              MOV     #DDP8,R1                ;216,440,101
5093  026652  012702  000004              MOV     #4,R2                   ;GET GENERATED
5094  026656  022021                DD19:   CMP     (R0)+,(R1)+
5095  026660  001402                        BEQ     DD20
5096  026662  000137  027516              JMP     @#DDER5         ;DATA ERROR.
5097  026666  077205                DD20:   SOB     R2,DD19
5098  026670  000137  027554              JMP     @#DDER6
```

```
5099  026674  077220                  DD21:   SOB     R2,DD18
5100  026676  020405                          CMP     R4,R5                   ;EPS CORRECT?
5101  026700  001402                          BEQ     DD22
5102  026702  000137  027310                  JMP     @#DDERO                 ;BAD FPS
5103                                  ;AC NEG FSRC    POS     /FSRC/ > /AC/
5104  026706                          DD22:
5105  026706  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5106  026710  012704  003200                  MOV     #3200,R4                ;SET FIO,FIV, AND FD
5107  026714  170104                          LDFPS   R4
5108  026716  012737  026736  001236          MOV     #DD23,@#$TMP2
5109  026724  012700  030166                  MOV     #DDP6,R0                ;SET ACO OPERAND
5110  026730  172410                          LDD     (R0),AC0
5111  026732  012700  030136                  MOV     #DDP3,R0                ;FSPC
5112  026736  172010                  DD23:   ADDD    (R0),AC0                ;TEST INSTRUCTION
5113  026740  170205                          STFPS   R5                      ;GET FPS
5114  026742  012700  030076                  MOV     #DDDATO,R0              ;GET THE RESULT
5115  026746  174010                          STD     AC0,(R0)
5116  026750  012701  030176                  MOV     #DDP7,R1                ;IS IT CORRECT?
5117  026754  012702  000004                  MOV     #4,R2
5118  026760  022021                  DD24:   CMP     (R0)+,(R1)+
5119  026762  001415                          BEQ     DD27
5120  026764  012700  030076                  MOV     #DDDATO,R0              ;FLO,S FAILURE
5121  026770  012701  030206                  MOV     #DDP8,R1                ;CONSTANT (NOT 57)
5122  026774  012702  000004                  MOV     #4,R2                   ;216,042,101
5123  027000  021011                  DD25:   CMP     (R0),(R1)
5124  027002  001402                          BEQ     DD26
5125  027004  000137  027612                  JMP     @#DDER7                 ;DATA ERROR.
5126  027010  077205                  DD26:   SOB     R2,DD25
5127  027012  000137  027650                  JMP     @#DDER8
5128  027016  077220                  DD27:   SOB     R2,DD24
5129  027020  020405                          CMP     R4,R5                   ;FPS CORRECT?
5130  027022  001402                          BEQ     DD30
5131  027024  000137  027310                  JMP     @#DDERO                 ;BAD FPS
5132                                  ;AC0 POS        FSRC    NEG     /AC/</FRSRC/
5133  027030                          DD30:
5134  027030  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5135  027032  012704  003200                  MOV     #3200,R4                ;SET FIO,FIV,AND FD
5136  027036  170104                          LDFPS   R4
5137  027040  012737  027060  001236          MOV     #DD31,@#$TMP2
5138  027046  012700  030146                  MOV     #DDP4,R0                ;SET ACO OPERAND
5139  027052  172410                          LDD     (R0),AC0
5140  027054  012700  030150                  MOV     #DDP5,R0                ;FSPC
5141  027060  172010                  DD31:   ADDD    (R0),AC0                ;TEST INSTRUCTION
5142  027062  170205                          STFPS   R5                      ;GET FPS
5143  027064  012700  030076                  MOV     #DDDATO,R0              ;GET THE RESULT
5144  027070  174010                          STD     AC0,(R0)
5145  027072  012701  030206                  MOV     #DDP8,R1                ;IS IT CORRECT
5146  027076  012702  000004                  MOV     #4,R2
5147  027102  022021                  DD32:   CMP     (R0)+,(R1)+
5148  027104  001415                          BEQ     DD35                    ;ADD-SUB
5149  027106  012700  030076                  MOV     #DDDATO,R0              ;FLOWAS FAILURE
5150  027112  012701  030176                  MOV     #DDP7,R1                ;CON 216 N440 NOT 141
5151  027116  012702  000004                  MOV     #4,R2           ;GET GENERATED
5152  027122  022021                  DD33:   CMP     (R0)+,(R1)+             ;FOR THE ALLIGNMENT
5153  027124  001402                          BEQ     DD34                    ;FLOWS?
5154  027126  000137  027706                  JMP     @#DDER9                 ;DATA ERROR, D
```

```
5155  027132  077205                  DD34:   SOB     R2,DD33
5156  027134  000137  027744                  JMP     @#DDER10
5157  027140  077220                  DD35:   SOB     R2,DD32
5158  027142  052704  000010                  BIS     #10,R4
5159  027146  020405                          CMP     R4,R5                   ;FPS CORRECT?
5160  027150  001402                          BEQ     #DD36
5161  027152  000137  027310                  JMP     @#DDER0                 ;BAD FPS
5162                                  ;ACO NEG        FSRC    POS     /FSRC/</AC/
5163  027156                          DD36:
5164  027156  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
5165  027160  012704  003200                  MOV     #3200,R4                ;SET FIO, FIV, AND FD
5166  027164  170104                          LDFPS   R4
5167  027166  012737  027206  001236          MOV     #DD37,@#$TMP2
5168  027174  012700  030156                  MOV     #DDP5,R0                ;SET ACO OPERAND
5169  027200  172410                          LDD     (R0),AC0
5170  027202  012700  030146                  MOV     #DDP4,R0                ;FSPC
5171  027206  172010                  DD37:   ADDD    (R0),AC0                ;TEST INSTRUCTION
5172  027210  170205                          STFPS   R5                      ;GET FPS
5173  027212  012700  030076                  MOV     #DDDAT0,R0              ;GET THE RESULT
5174  027216  174010                          STD     AC0,(R0)
5175
5176  027220  012701  030206                  MOV     #DDP8,R1                ;IS IT CORRECT
5177  027224  012702  000004                  MOV     #4,R2
5178  027230  022021                  DD38:   CMP     (R0)+,(R1)+
5179  027232  001415                          BEQ     DD41
5180  027234  012700  030076                  MOV     #DDDAT0,R0              ;ADD SUB
5181  027240  012701  030176                  MOV     #DDP7,R1                ;FLOWS FAILURES
5182  027244  012702  000004                  MOV     #4,R2                   ;GET 216,042,141
5183  027250  022021                  DD39:   CMP     (R0)+,(R1)+             ;FOR THE ALLIGNMENT
5184  027252  001402                          BEQ     DD40                    ;FLOWS?
5185  027254  000137  030002                  JMP     @#DDER11                ;DATA ERROR. D
5186  027260  077205                  DD40:   SOB     R2,DD39
5187  027262  000137  030040                  JMP     @#DDER12                ;BAD CONSTANT.D
5188  027266  077220                  DD41:   SOB     R2,DD38
5189  027270  052704  000010                  BIS     #10,R4
5190  027274  020405                          CMP     R4,R5                   ;FPS CORRECT?
5191  027276  001402                          BEQ     DD42
5192  027300  000137  027310                  JMP     @#DDER0                 ;BAD FPS
5193  027304  000137  030226          DD42:   JMP     @#DDDONE
5194  027310  010437  001242          DDER0:  MOV     R4,@#$TMP4              ;FPS ERROR
5195  027314  010537  001240                  MOV     R5,@#$TMP3
5196  027320  104002                  1$:     ERROR   2
5197  027322  000137  030226                  JMP     @#DDDONE
5198  027326                          DDER1:
5199  027326  012737  030116  001240          MOV     #DDP1,@#$TMP3
5200  027334  012737  030116  001242          MOV     #DDP1,@#$TMP4
5201  027342  012737  030076  001244          MOV     #DDDAT0,@#$TMP5
5202  027350  012737  030116  001246          MOV     #DDP1,@#$TMP6
5203  027356  104002                  1$:     ERROR   2
5204  027360  000137  030226                  JMP     @#DDDONE
5205  027364                          DDER2:
5206  027364  012737  030116  001240          MOV     #DDP1,@#$TMP3
5207  027372  012737  030116  001242          MOV     #DDP1,@#$TMP4
5208  027400  012737  030076  001244          MOV     #DDDAT0,@#$TMP5
5209  027406  012737  030216  001246          MOV     #DDP9,@#$TMP6
5210  027414  104002                  1$:     ERROR   2
```

```
 5211  027416  000137  030226                    JMP     @#DDDONE
 5212  027422                            DDER3:
 5213  027422  012737  030116  001240            MOV     #DDP1,@#$TMP3
 5214  027430  012737  030126  001242            MOV     #DDP2,@#$TMP4
 5215  027436  012737  030076  001244            MOV     #DDDAT0,@#$TMP5
 5216  027444  012737  030106  001246            MOV     #DDP0,@#$TMP6
 5217  027452  104002                    1$:     ERROR   2
 5218  027454  000137  030226                    JMP     @#DDDONE
 5219  027460                            DDER4:
 5220  027460  012737  030126  001240            MOV     #DDP2,@#$TMP3
 5221  027466  012737  030116  001242            MOV     #DDP1,@#$TMP4
 5222  027474  012737  030076  001244            MOV     #DDDAT0,@#$TMP5
 5223  027502  012737  030106  001246            MOV     #DDP0,@#$TMP6
 5224  027510  104002                    1$:     ERROR   2
 5225  027512  000137  030226                    JMP     @#DDDONE
 5226  027516                            DDER5:
 5227  027516  012737  030166  001240            MOV     #DDP6,@#$TMP3
 5228  027524  012737  030136  001242            MOV     #DDP3,@#$TMP4
 5229  027532  012737  030076  001244            MOV     #DDDAT0,@#$TMP5
 5230  027540  012737  030176  001246            MOV     #DDP7,@#$TMP6
 5231  027546  104002                    1$:     ERROR   2
 5232  027550  000137  030226                    JMP     @#DDDONE
 5233  027554                            DDER6:
 5234  027554  012737  030166  001240            MOV     #DDP6,@#$TMP3
 5235  027562  012737  030136  001242            MOV     #DDP3,@#$TMP4
 5236  027570  012737  030076  001244            MOV     #DDDAT0,@#$TMP5
 5237  027576  012737  030176  001246            MOV     #DDP7,@#$TMP6
 5238  027604  104002                    1$:     ERROR   2
 5239  027606  000137  030226                    JMP     @#DDDONE
 5240  027612                            DDER7:
 5241  027612  012737  030136  001240            MOV     #DDP3,@#$TMP3
 5242  027620  012737  030166  001242            MOV     #DDP6,@#$TMP4
 5243  027626  012737  030076  001244            MOV     #DDDAT0,@#$TMP5
 5244  027634  012737  030176  001246            MOV     #DDP7,@#$TMP6
 5245  027642  104002                    1$:     ERROR   2
 5246  027644  000137  030226                    JMP     @#DDDONE
 5247  027650                            DDER8:
 5248  027650  012737  030136  001240            MOV     #DDP3,@#$TMP3
 5249  027656  012737  030166  001242            MOV     #DDP6,@#$TMP4
 5250  027664  012737  030076  001244            MOV     #DDDAT0,@#$TMP5
 5251  027672  012737  030176  001246            MOV     #DDP7,@#$TMP6
 5252  027700  104002                    1$:     ERROR   2
 5253  027702  000137  030226                    JMP     @#DDDONE
 5254  027706                            DDER9:
 5255  027706  012737  030156  001240            MOV     #DDP5,@#$TMP3
 5256  027714  012737  030146  001242            MOV     #DDP4,@#$TMP4
 5257  027722  012737  030076  001244            MOV     #DDDAT0,@#$TMP5
 5258  027730  012737  030206  001246            MOV     #DDP8,@#$TMP6
 5259  027736  104002                    1$:     ERROR   2
 5260  027740  000137  030226                    JMP     @#DDDONE
 5261  027744                            DDER10:
 5262  027744  012737  030156  001240            MOV     #DDP5,@#$TMP3
 5263  027752  012737  030146  001242            MOV     #DDP4,@#$TMP4
 5264  027760  012737  030076  001244            MOV     #DDDAT0,@#$TMP5
 5265  027766  012737  030206  001246            MOV     #DDP8,@#$TMP6
 5266  027774  104002                    1$:     ERROR   2
```

```
 5267  027776  000137  030226              JMP     @#DDDONE
 5268  030002                      DDER11:
 5269  030002  012737  030146  001240      MOV     #DDP4,@#$TMP3
 5270  030010  012737  030156  001242      MOV     #DDP5,@#$TMP4
 5271  030016  012737  030076  001244      MOV     #DDDATO,@#$TMP5
 5272  030024  012737  030206  001246      MOV     #DDP8,@#$TMP6
 5273  030032  104002              1$:     ERROR   2
 5274  030034  000137  030226              JMP     @#DDDONE
 5275  030040                      DDER12:
 5276  030040  012737  030146  001240      MOV     #DDP4,@#$TMP3
 5277  030046  012737  030156  001242      MOV     #DDP5,@#$TMP4
 5278  030054  012737  030076  001244      MOV     #DDDATO,@#$TMP5
 5279  030062  012737  030206  001246      MOV     #DDP8,@#$TMP6
 5280  030070  104002              1$:     ERROR   2
 5281  030072  000137  030226              JMP     @#DDDONE
 5282  030076  000000              DDDATO: 0
 5283  030100  000000                      0
 5284  030102  000000                      0
 5285  030104  000000                      0
 5286  030106  000000              DDP0:   0
 5287  030110  000000                      0
 5288  030112  000000                      0
 5289  030114  000000                      0
 5290  030116  100200              DDP1:   100200                  ;-DDP2
 5291  030120  000000                      0
 5292  030122  000000                      0
 5293  030124  000000                      0
 5294  030126  000200              DDP2:   200                     ;-DDP1
 5295  030130  000000                      0
 5296  030132  000000                      0
 5297  030134  000000                      0
 5298  030136  001100              DDP3:   1100                    ;EXP=4
 5299  030140  000000                      0                       ;FRAC=...110...
 5300  030142  000000                      0
 5301  030144  000000                      0
 5302  030146  000600              DDP4:   600                     ;EXP=3
 5303  030150  000000                      0                       ;FRAC=...100...
 5304  030152  000000                      0
 5305  030154  000000                      0
 5306  030156  101100              DDP5:   101100                  ;-DDP3
 5307  030160  000000                      0
 5308  030162  000000                      0
 5309  030164  000000                      0
 5310  030166  100600              DDP6:   100600                  ;-DDP4
 5311  030170  000000                      0
 5312  030172  000000                      0
 5313  030174  000000                      0
 5314  030176  001000              DDP7:   1000                    ;DDP3+DDP6
 5315  030200  000000                      0
 5316  030202  000000                      0
 5317  030204  000000                      0
 5318  030206  101000              DDP8:   101000                  ;DDP5+DDP4
 5319  030210  000000                      0
 5320  030212  000000                      0
 5321  030214  000000                      0
 5322  030216  100400              DDP9:   100400                  ;DDP1+DDP1
```

```
 5323  030220  000000                          0
 5324  030222  000000                          0
 5325  030224  000000                          0
 5326  030226  000005                DDDONE:   RESET
 5327                                ;;*****************************************************************
 5328                                ;*TEST 35       SUBD TEST
 5329                                ;*
 5330                                ;* THIS IS A TEST OF THE SUBD INSTRUCTION.
 5331                                ;* BOTH A POSITIVE AND A NEGATIVE NUMBER
 5332                                ;* IS SUBTRACTED FROM IT SELF
 5333                                ;*
 5334                                ;;*****************************************************************
 5335  030230  000004                TST35:  SCOPE
 5336                                ;       USE POSITIVE OPERANDS
 5337  030232                        EE1:
 5338  030232  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5339  030234  012704  003200                MOV     #3200,R4        ;SET FIU, FIV, AND FD
 5340  030240  170104                        LDFPS   R4
 5341  030242  012737  030262  001236        MOV     #EE2,@#$TMP2
 5342  030250  012700  030742                MOV     #EEP1,R0        ;SET AC0 OPERAND
 5343  030254  172410                        LDD     (R0),AC0
 5344  030256  012700  030742                MOV     #EEP1,R0        ;FSPC
 5345  030262  173010                EE2:    SUBD    (R0),AC0        ;TEST INSTRUCTION
 5346  030264  170205                        STFPS   R5              ;GET FPS
 5347  030266  012700  030720                MOV     #EEDATO,R0      ;GET THE RESULT
 5348  030272  174010                        STD     AC0,(R0)
 5349  030274  012701  030730                MOV     #EEP0,R1        ;IS IT CORRECT?
 5350  030300  012702  000004                MOV     #4,R2
 5351  030304  022021                EE3:    CMP     (R0)+,(R1)+
 5352  030306  001415                        BEQ     EE6
 5353  030310  012700  030720                MOV     #EEDATO,R0      ;DID A BAD
 5354
 5355  030314  012701  030752                MOV     #EEP2,R1        ;CONSTANT (NOT 57)
 5356  030320  012702  000004                MOV     #4,R2           ;GET GENERATED
 5357  030324  022021                EE4:    CMP     (R0)+,(R1)+     ;FOR THE ALLIGNMENT
 5358  030326  001402                        BEQ     EE5             ;FLOWS?
 5359  030330  000137  030530                JMP     @#LEER1         ;DATA ERROR.D
 5360  030334  077205                EE5:    SOB     R2,EE4
 5361  030336  000137  030566                JMP     @#EEER2         ;BAD CONSTANT.D
 5362  030342  077220                EE6:    SOB     R2,EE3
 5363  030344  052704  000004                BIS     #4,R4
 5364  030350  020405                        CMP     R4,R5           ;FPS CORRECT?
 5365  030352  001402                        BEQ     EE7
 5366  030354  000137  030512                JMP     @#EEER0         ;BAD FPS
 5367                                ;USE NEGATIVE OPERANDS
 5368  030360                        EE7:
 5369  030360  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5370  030362  012704  003200                MOV     #3200,R4        ;SET FIO, FIV, AND FD
 5371  030366  170104                        LDFPS   R4
 5372  030370  012737  030410  001236        MOV     #EE8,@#$TMP2
 5373  030376  012700  030762                MOV     #EEP3,R0        ;SET AC0 OPERAND
 5374  030402  172410                        LDD     (R0),AC0
 5375  030404  012700  030762                MOV     #EEP3,R0        ;FSPC
 5376  030410  173010                EE8:    SUBD    (R0),AC0        ;TEST INSTRUCTION
 5377  030412  170205                        STFPS   R5              ;GET FPS
 5378  030414  012700  030720                MOV     #EEDATO,R0      ;GET THE RESULT
```

```
 5379  030420  174010                        STD     AC0,(R0)
 5380  030422  012701  030730                MOV     #EEP0,R1            ;IS IT CORRECT?
 5381  030426  012702  000004                MOV     #4,R2
 5382  030432  022021                EE9:    CMP     (R0)+,(R1)+
 5383  030434  001415                        BEQ     EE12
 5384  030436  012700  030720                MOV     #EEDAT0,R0          ;DID A BAD
 5385  030442  012701  030772                MOV     #EEP4,R1            ;CONSTANT (NOT 57)
 5386  030446  012702  000004                MOV     #4,R2               ;GET GENERATED
 5387  030452  022021                EE10:   CMP     (R0)+,(R1)+         ;FOR THE ALLIGNMENT
 5388  030454  001402                        BEQ     EE11                ;FLOWS?
 5389  030456  000137  030624                JMP     @#EEER3             ;DATA ERROR.D
 5390  030462  077205                EE11:   SOB     R2,EE10
 5391  030464  000137  030662                JMP     @#EEER4             ;BAD CONSTANT.D
 5392  030470  077220                EE12:   SOB     R2,EE9
 5393  030472  052704  000004                BIS     #4,R4
 5394  030476  020405                        CMP     R4,R5               ;FPS CORRECT?
 5395  030500  001402                        BEQ     EE13
 5396  030502  000137  030512                JMP     @#EEER0             ;BAD FPS.
 5397  030506  000137  031002        EE13:   JMP     @#EEDONE
 5398  030512  010437  001242        EEER0:  MOV     R4,@#$TMP4          ;BAD FPS
 5399  030516  010537  001240                MOV     R5,@#$TMP3
 5400  030522  104002                1$:     ERROR   2
 5401  030524  000137  031002                JMP     @#EEDONE
 5402  030530                        EEER1:
 5403  030530  012737  030742  001240        MOV     #EEP1,@#$TMP3
 5404  030536  012737  030742  001242        MOV     #EEP1,@#$TMP4
 5405  030544  012737  030720  001244        MOV     #EEDAT0,@#$TMP5
 5406  030552  012737  030730  001246        MOV     #EEP0,@#$TMP6
 5407  030560  104002                1$:     ERROR   2
 5408  030562  000137  031002                JMP     @#EEDONE
 5409  030566                        EEER2:
 5410  030566  012737  030742  001240        MOV     #EEP1,@#$TMP3
 5411  030574  012737  030742  001242        MOV     #EEP1,@#$TMP4
 5412  030602  012737  030720  001244        MOV     #EEDAT0,@#$TMP5
 5413  030610  012737  030730  001246        MOV     #EEP0,@#$TMP6
 5414  030616  104002                1$:     ERROR   2
 5415  030620  000137  031002                JMP     @#EEDONE
 5416  030624                        EEER3:
 5417  030624  012737  030762  001240        MOV     #EEP3,@#$TMP3
 5418  030632  012737  030762  001242        MOV     #EEP3,@#$TMP4
 5419  030640  012737  030720  001244        MOV     #EEDAT0,@#$TMP5
 5420  030646  012737  030730  001246        MOV     #EEP0,@#$TMP6
 5421  030654  104002                1$:     ERROR   2
 5422  030656  000137  031002                JMP     @#EEDONE
 5423  030662                        EEER4:
 5424  030662  012737  030762  001240        MOV     #EEP3,@#$TMP3
 5425  030670  012737  030762  001242        MOV     #EEP3,@#$TMP4
 5426  030676  012737  030720  001244        MOV     #EEDAT0,@#$TMP5
 5427  030704  012737  030730  001246        MOV     #EEP0,@#$TMP6
 5428  030712  104002                1$:     ERROR   2
 5429  030714  000137  031002                JMP     @#EEDONE
 5430  030720  000000                EEDAT0: 0
 5431  030722  000000                        0
 5432  030724  000000                        0
 5433  030726  000000                        0
 5434  030730  000000                EEP0:   0
```

```
 5435  030732  000000                          0
 5436  030734  000000                  00000
 5437  030736  000000                          0
 5438  030740  000000                          0
 5439  030742  000200                  EEP1:   200
 5440  030744  000000                          0
 5441  030746  000000                          0
 5442  030750  000000                          0
 5443  030752  000400                  EEP2:   400
 5444  030754  000000                          0
 5445  030756  000000                          0
 5446  030760  000000                          0
 5447  030762  100200                  EEP3:   100200
 5448  030764  000000                          0
 5449  030766  000000                          0
 5450  030770  000000                          0
 5451  030772  100400                  EEP4:   100400
 5452  030774  000000                          0
 5453  030776  000000                          0
 5454  031000  000000                          0
 5455  031002                          EEDONE:
 5456  031002  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 5457                                                                  ;SEE IF THE USER HAS EXPRFSSED
 5458                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 5459                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 5460                                                                  ;THE USER TYPED CONTROL G?).
 5461                                  ;;*****************************************************************
 5462                                  ;*TEST 36       NORMALIZE ALGORITHM TEST
 5463                                  ;*
 5464                                  ;* THIS IS A TEST OF THE NORMALIZE
 5465                                  ;* FLOW ALGORITHM.  TWO PATTERNS ARE USED,
 5466                                  ;* FIRST THE MINIMUM SITUATION REQUIRING ONE
 5467                                  ;* LEFT SHIFT AND THEN THE MAXIMUM SITUATION
 5468                                  ;* REQUIRING 56 SHIFTS.
 5469                                  ;*
 5470                                  ;;*****************************************************************
 5471  031004  000004                  TST36:  SCOPE
 5472                                  ;USE DATA PATTERNS THAT REQUIRE ONLY ONE LEFT SHIFT TO NORMALIZE
 5473  031006                          FF1:
 5474  031006  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5475  031010  012704  003200                  MOV     #3200,R4                ;SET FIO, FIV, AND FD
 5476  031014  170104                          LDFPS   R4
 5477  031016  012737  031036  001236          MOV     #FF2,@#$TMP2
 5478  031024  012700  031330                  MOV     #FFP2,R0                ;SET AC0 OPERAND
 5479  031030  172410                          LDD     (R0),AC0
 5480  031032  012700  031340                  MOV     #FFP3,R0                ;FSPC
 5481  031036  172010                  FF2:    ADDD    (R0),AC0                ;TEST INSTRUCTION
 5482  031040  170205                          STFPS   R5                      ;GET FPS
 5483  031042  012700  031300                  MOV     #FFDATO,R0              ;GET THE RESULT
 5484  031046  174010                          STD     AC0,(R0)
 5485  031050  012701  031350                  MOV     #FFP4, R1               ;IS IT CORRECT
 5486  031054  012702  000004                  MOV     #4,R2
 5487  031060  022021                  FF3:    CMP     (R0)+,(R1)+
 5488  031062  001401                          BEQ     FF4
 5489  031064  000466                          BR      FFER2                   ;BAD DATA
 5490  031066  077204                  FF4:    SOB     R2,FF3
```

```
5491  031070  020405                          CMP     R4,R5                   ;FPS CORRECT?
5492  031072  001401                          BEQ     FF5
5493  031074  000435                          BR      FFER0                   ;BAD FPS
5494                                  ;USE DATA PATTERNS WHICH REQUIRE 56 LEFT SHIFTS TO NORMALIZE
5495                                  ;THE RESULT
5496  031076                          FF5:
5497  031076  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5498  031100  012704  003200                  MOV     #3200,R4                ;SET FIU, FIV, AND FD
5499  031104  170104                          LDFPS   R4
5500  031106  012737  031126  001236          MOV     #FF6,@#$TMP2
5501  031114  012700  031310                  MOV     #FFP0,R0                ;SET AC0 OPERAND
5502  031120  172410                          LDD     (R0),AC0
5503  031122  012700  031320                  MOV     #FFP1,R0                ;FSRC
5504  031126  172010                  FF6:    ADDD    (R0),AC0                ;TEST INSTRUCTION
5505  031130  170205                          STFPS   R5                      ;GET FPS
5506  031132  012700  031300                  MOV     #FFDATO,R0              ;GET THE RESULT
5507  031136  174010                          STD     AC0,(R0)
5508  031140  012701  031350                  MOV     #FFP4,R1                ;IS IT CORRECT
5509  031144  012702  000004                  MOV     #4,R2
5510  031150  022021                  FF7:    CMP     (R0)+,(R1)+
5511  031152  001401                          BEQ     FF10
5512  031154  000413                          BR      FFER1                   ;BATA
5513  031156  077204                  FF10:   SOB     R2,FF7
5514  031160  020405                          CMP     R4,R5                   ;FPS CORRECT?
5515  031162  001401                          BEQ     FF11
5516  031164  000401                          BR      FFER0           ;BAD FPS
5517  031166  000474                  FF11:   BR      FFDONE
5518
5519  031170  010537  001240          FFER0:  MOV     R5,@#$TMP3
5520  031174  010437  001242                  MOV     R4,@#$TMP4
5521  031200  104002                  1$:     ERROR   2
5522  031202  000466                          BR      FFDONE
5523
5524  031204                          FFER1:
5525  031204  012737  031320  001240          MOV     #FFP1,@#$TMP3
5526  031212  012737  031310  001242          MOV     #FFP0,@#$TMP4
5527  031220  012737  031300  001244          MOV     #FFDATO,@#$TMP5
5528  031226  012737  031350  001246          MOV     #FFP4,@#$TMP6
5529  031234  104001                  1$:     ERROR   1
5530  031236  000137  031360                  JMP     @#FFDONE
5531
5532  031242                          FFER2:
5533  031242  012737  031340  001240          MOV     #FFP3,@#$TMP3
5534  031250  012737  031330  001242          MOV     #FFP2,@#$TMP4
5535  031256  012737  031300  001244          MOV     #FFDATO,@#$TMP5
5536  031264  012737  031350  001246          MOV     #FFP4,@#$TMP6
5537  031272  104001                  1$:     ERROR   1
5538  031274  000137  031360                  JMP     @#FFDONE
5539
5540
5541  031300  000000                  FFDATO: 0
5542  031302  000000                          0
5543  031304  000000                          0
5544  031306  000000                          0
5545
5546  031310  016000                  FFP0:   16000
```

```
 5547  031312  000000                    0
 5548  031314  000000                    0
 5549  031316  000001                    1
 5550  031320  116000            FFP1:   116000
 5551  031322  000000                    0
 5552  031324  000000                    0
 5553  031326  000000                    0
 5554  031330  000500            FFP2:   500
 5555  031332  000000                    0
 5556  031334  000000                    0
 5557  031336  000000                    0
 5558  031340  100400            FFP3:   100400
 5559  031342  000000                    0
 5560  031344  000000                    0
 5561  031346  000000                    0
 5562  031350  000200            FFP4:   200                     ;FFP4=FFP0+FFP1
 5563  031352  000000                    0                       ;    =FFP3+FFP4
 5564  031354  000000                    0
 5565  031356  000000                    0
 5566
 5567  031360                    FFDONE:
 5568
 5569
 5570  031360                    TST37:
 5571
 5572
 5573
 5574                            ;*************************************************************************
 5575                            ;*************************************************************************
 5576                            ;FLOATING POINT SECOND PART
 5577                            ;*************************************************************************
 5578                            ;*************************************************************************
 5579
 5580
 5581                            ;;*****************************************************************
 5582                            ;*TEST 37       ROUND\TRUNK TEST
 5583                            ;*
 5584                            ;* THIS IS A TEST OF THE ROUND\TRUNK
 5585                            ;* FLOWS.  IN PARTICULAR TWO THINGS ARE TESTED:
 5586                            ;* FIRST A CONDITION IN WHICH ROUNDING
 5587                            ;* RESULTS IN THE NEED FOR RENORMALIZATION, AND
 5588                            ;* SECOND  THE PSW CONDITION CODES N AND
 5589                            ;* Z BIT COMMBINATIONS
 5590                            ;*
 5591                            ;;*****************************************************************
 5592  031360  000004            TST37:  SCOPE
 5593
 5594                                    ;ROUND AND NORMALIZE TEST
 5595
 5596  031362                    HH1:
 5597  031362  104414                    LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5598  031364  012704  003200            MOV     #3200,R4                ;SET FIU, FIV, AND FD
 5599  031370  170104                    LDFPS   R4
 5600  031372  012737  031412  001236    MOV     #HH2,@#$TMP2
 5601  031400  012700  033142            MOV     #HHP0,R0                ;SET AC0 OPERAND
 5602  031404  172410                    LDD     (R0),AC0
```

```
5603  031406  012700  033152          MOV     #HHP1,R0            ;FSPC
5604  031412  172010                HH2:  ADDD    (R0),AC0            ;TEST INSTRUCTION
5605  031414  170205                      STFPS   R5                  ;GET FPS
5606  031416  012700  033132              MOV     #HHDATO,R0          ;GET THE RESULT
5607  031422  174010                      STD     AC0,(R0)
5608  031424  012701  033162              MOV     #HHP2,R1            ;IS IT CORRECT
5609  031430  012702  000004              MOV     #4,R2
5610  031434  022021                HH3:  CMP     (R0)+,(R1)+
5611  031436  001415                      BEQ     HH6
5612  031440  012700  033132              MOV     #HHDATO,R0          ;DID FLOW GO
5613  031444  012701  033172              MOV     #HHP3,R1            ;FROM STATE 663
5614  031450  012702  000004              MOV     #4,R2               ;TO 313 INSTEAD
5615  031454  022021                HH4:  CMP     (R0)+,(R1)+         ;OF TO 353
5616  031456  001402                      BEQ     HH5
5617  031460  000137  032152              JMP     @#HHER0
5618  031464  077205                HH5:  SOB     R2,HH4
5619  031466  000137  032220              JMP     @#HHER1
5620  031472  077220                HH6:  SOB     R2,HH3
5621  031474  020405                      CMP     R4,R5               ;FPS CORRECT?
5622  031476  001402                      BEQ     HH7
5623  031500  000137  032266              JMP     @#HHER00
5624
5625                                  ;THIS IS A TEST OF THE ABILITY
5626                                  ;OF NORMALIZE TO PRODUCE A ZERO EXP. AND
5627                                  ;OF THE R\T ALGORITHM TO PROPERLY SET THE FPS
5628
5629  031504                        HH7:
5630  031504  104414                      LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
5631  031506  012704  043200              MOV     #043200,R4                  ;SET FIU,FIV,AND FD
5632                                                                      ;FID
5633  031512  170104                      LDFPS   R4
5634  031514  012737  031542  001236      MOV     #HH8,@#$TMP2                ;IN CASE UNDERFLOW
5635  031522  012737  033062  000244      MOV     #HHTRAP,@#FPVECT            ;TRAP OCCURS
5636  031530  012700  033212              MOV     #HHP5,R0                    ;SET AC0 OPERAND
5637  031534  172410                      LDD     (R0),AC0
5638  031536  012700  033222              MOV     #HHP6,R0                    ;FSPC
5639  031542  172010                HH8:  ADDD    (R0),AC0                    ;TEST INSTRUCTION
5640  031544  170205                      STFPS   R5                          ;GET FPS
5641  031546  012700  033132              MOV     #HHDATO,R0                  ;GET THE RESULT
5642  031552  174010                      STD     AC0,(R0)
5643  031554  012701  033202              MOV     #HHP4,R1                    ;IS IT CORRECT
5644  031560  012702  000004              MOV     #4,P2
5645  031564  022021                HH9:  CMP     (R0)+,(R1)+
5646  031566  001402                      BEQ     HH10
5647  031570  000137  032334              JMP     @#HHER2
5648  031574  077205                HH10: SOB     R2,HH9
5649  031576  052704  100004              BIS     #100004,R4                  ;FPS CORRECT?
5650  031602  020405                      CMP     R4,R5
5651  031604  001402                      BEQ     HH11
5652  031606  000137  032402              JMP     @#HHER3
5653                                  ;THIS IS A TEST OF THE R\T ALGORITHM'S
5654                                  ;ABILITY TO SET BOTH N AND Z ON A - 0 RESULT.
5655  031612                        HH11:
5656  031612  104414                      LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
5657                                                                      ;SET FIV, FIV, AND FD
5658  031614  012704  043200              MOV     #043200,R4
```

```
 5659  031620  170104                          LDFPS   R4
 5660  031622  012737  031642  001236          MOV     #HH12,@#$TMP2
 5661  031630  012700  033242                  MOV     #HHP8,R0                ;SET AC0 OPERAND
 5662  031634  172410                          LDD     (R0),AC0
 5663  031636  012700  033252                  MOV     #HHP9,R0                ;FSPC
 5664  031642  172010                  HH12:   ADDD    (R0),AC0                ;TEST INSTRUCTION
 5665  031644  170205                          STFPS   R5                      ;GET FPS
 5666  031646  012700  033132                  MOV     #HHDAT0,R0              ;GET THE RESULT
 5667  031652  174010                          STD     AC0,(R0)
 5668  031654  012701  033232                  MOV     #HHP7,R1                ;IS IT CORRECT
 5669  031660  012702  000004                  MOV     #4,R2
 5670  031664  022021                  HH13:   CMP     (R0)+,(R1)+
 5671  031666  001415                          BEQ     HH16
 5672  031670  012700  033132                  MOV     #HHDAT0,R0
 5673  031674  012701  033202                  MOV     #HHP4,R1
 5674  031700  012702  000004                  MOV     #4,R2
 5675  031704  022021                  HH14:   CMP     (R0)+,(R1)+
 5676  031706  001402                          BEQ     HH15
 5677  031710  000137  032450                  JMP     @#HHER4
 5678  031714  077205                  HH15:   SOB     R2,HH14
 5679  031716  000137  032516                  JMP     @#HHER5
 5680  031722  077220                  HH16:   SOB     R2,HH13
 5681  031724  052704  100014                  BIS     #100014,R4              ;FPS CORRECT?
 5682  031730  020405                          CMP     R4,R5
 5683  031732  001402                          BEQ     HH17
 5684  031734  000137  032564                  JMP     @#HHER6
 5685                                  ;TEST THAT CC ARE CLEARED BY R\T
 5686  031740                          HH17:
 5687  031740  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 5688  031742  012704  000200                  MOV     #00200,R4               ;SET FIV, FIV, AND FD
 5689  031746  170104                          LDFPS   R4
 5690  031750  012737  031776  001236          MOV     #HH18,@#$TMP2
 5691  031756  012737  062524  000244          MOV     #FPSPUR,@#FPVECT
 5692  031764  012700  033242                  MOV     #HHP8,R0                ;SET AC0 OPERAND
 5693  031770  172410                          LDD     (R0),AC0
 5694  031772  012700  033242                  MOV     #HHP8,R0                ;FSPC
 5695  031776  172010                  HH18:   ADDD    (R0),AC0                ;TEST INSTRUCTION
 5696  032000  170205                          STFPS   R5                      ;GET FPS
 5697  032002  012700  033132                  MOV     #HHDAT0,R0              ;GET THE RESULT
 5698  032006  174010                          STD     AC0,(R0)
 5699  032010  012701  033262                  MOV     #HHP10,R1               ;IS IT CORRECT
 5700  032014  012702  000004                  MOV     #4,R2
 5701  032020  022021                  HH19:   CMP     (R0)+,(R1)+
 5702  032022  001402                          BEQ     HH20
 5703  032024  000137  032632                  JMP @#HHER7
 5704  032030  077205                  HH20:   SOB     R2,HH19
 5705  032032  052704  000000                  BIS     #00000,R4               ;FPS CORRECT?
 5706  032036  020405                          CMP     R4,R5
 5707  032040  001402                          BEQ     HH21
 5708  032042  000137  032700                  JMP     @#HHER8
 5709                                  ;TEST THAT N IS SET BY R\T
 5710  032046                          HH21:
 5711  032046  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 5712  032050  012704  003200                  MOV     #3200,R4                ;SET FIV, FIV, AND FD
 5713  032054  170104                          LDFPS   R4
 5714  032056  012737  032076  001236          MOV     #HH22,@#$TMP2
```

```
5715  032064  012700  033212            MOV     #HHP5,R0                ;SET AC0 OPERAND
5716  032070  172410                    LDD     (R0),AC0
5717  032072  012700  033212            MOV     #HHP5,R0                ;FSPC
5718  032076  172010            HH22:   ADDD    (R0),AC0                ;TEST INSTRUCTION
5719  032100  170205                    STFPS   R5                      ;GET FPS
5720  032102  012700  033132            MOV     #HHDAT0,R0              ;GET THE RESULT
5721  032106  174010                    STD     AC0,(R0)
5722  032110  012701  033272            MOV     #HHP11,R1               ;IS IT CORRECT
5723  032114  012702  000004            MOV     #4,R2
5724  032120  022021            HH23:   CMP     (R0)+,(R1)+
5725  032122  001402                    BEQ     HH24
5726  032124  000137  032746            JMP     @#HHER9
5727  032130  077205            HH24:   SOB     R2,HH23
5728  032132  052704  000010            BIS     #10,R4
5729  032136  020405                    CMP     R4,R5                   ;FPS CORRECT?
5730  032140  001402                    BEQ     HH25
5731  032142  000137  033014            JMP     @#HHER10
5732  032146  000137  033302    HH25:   JMP     @#HHDONE
5733  032152            HHER0:
5734  032152  010537  001252            MOV     R5,@#$TMP10
5735  032156  010437  001254            MOV     R4,@#$TMP11
5736  032162  012737  033152  001240    MOV     #HHP1,@#$TMP3
5737  032170  012737  033142  001242    MOV     #HHP0,@#$TMP4
5738  032176  012737  033132  001244    MOV     #HHDAT0,@#$TMP5
5739  032204  012737  033162  001246    MOV     #HHP2,@#$TMP6
5740  032212  104002            1$:     ERROR   2
5741  032214  000137  033302            JMP     @#HHDONE
5742  032220            HHER1:
5743  032220  010537  001252            MOV     R5,@#$TMP10
5744  032224  010437  001254            MOV     R4,@#$TMP11
5745  032230  012737  033152  001240    MOV     #HHP1,@#$TMP3
5746  032236  012737  033142  001242    MOV     #HHP0,@#$TMP4
5747  032244  012737  033132  001244    MOV     #HHDAT0,@#$TMP5
5748  032252  012737  033162  001246    MOV     #HHP2,@#$TMP6
5749  032260  104002            1$:     ERROR   2
5750  032262  000137  033302            JMP     @#HHDONE
5751  032266            HHER00:
5752  032266  010537  001252            MOV     R5,@#$TMP10
5753  032272  010437  001254            MOV     R4,@#$TMP11
5754  032276  012737  033152  001240    MOV     #HHP1,@#$TMP3
5755  032304  012737  033142  001242    MOV     #HHP0,@#$TMP4
5756  032312  012737  033132  001244    MOV     #HHDAT0,@#$TMP5
5757  032320  012737  033162  001246    MOV     #HHP2,@#$TMP6
5758  032326  104002            1$:     ERROR   2
5759  032330  000137  033302            JMP     @#HHDONE
5760  032334            HHER2:
5761  032334  010537  001252            MOV     R5,@#$TMP10
5762  032340  010437  001254            MOV     R4,@#$TMP11
5763  032344  012737  033222  001240    MOV     #HHP6,@#$TMP3
5764  032352  012737  033212  001242    MOV     #HHP5,@#$TMP4
5765  032360  012737  033132  001244    MOV     #HHDAT0,@#$TMP5
5766  032366  012737  033202  001246    MOV     #HHP4,@#$TMP6
5767  032374  104002            1$:     ERROR   2
5768  032376  000137  033302            JMP     @#HHDONE
5769  032402            HHER3:
5770  032402  010537  001252            MOV     R5,@#$TMP10
```

```
 5771  032406  010437  001254                 MOV     R4,@#$TMP11
 5772  032412  012737  033222  001240         MOV     #HHP6,@#$TMP3
 5773  032420  012737  033212  001242         MOV     #HHP5,@#$TMP4
 5774  032426  012737  033132  001244         MOV     #HHDATO,@#$TMP5
 5775  032434  012737  033202  001246         MOV     #HHP4,@#$TMP6
 5776  032442  104002                 1$:     ERROR   2
 5777  032444  000137  033302                 JMP     @#HHDONE
 5778  032450                         HHER4:
 5779  032450  010537  001252                 MOV     R5,@#$TMP10
 5780  032454  010437  001254                 MOV     R4,@#$TMP11
 5781  032460  012737  033252  001240         MOV     #HHP9,@#$TMP3
 5782  032466  012737  033242  001242         MOV     #HHP8,@#$TMP4
 5783  032474  012737  033132  001244         MOV     #HHDATO,@#$TMP5
 5784  032502  012737  033232  001246         MOV     #HHP7,@#$TMP6
 5785  032510  104002                 1$:     ERROR   2
 5786  032512  000137  033302                 JMP     @#HHDONE
 5787  032516                         HHFR5:
 5788  032516  010537  001252                 MOV     R5,@#$TMP10
 5789  032522  010437  001254                 MOV     R4,@#$TMP11
 5790  032526  012737  033252  001240         MOV     #HHP9,@#$TMP3
 5791  032534  012737  033242  001242         MOV     #HHP8,@#$TMP4
 5792  032542  012737  033132  001244         MOV     #HHDATO,@#$TMP5
 5793  032550  012737  033232  001246         MOV     #HHP7,@#$TMP6
 5794  032556  104002                 1$:     ERROR   2
 5795  032560  000137  033302                 JMP     @#HHDONE
 5796  032564                         HHER6:
 5797  032564  010537  001252                 MOV     R5,@#$TMP10
 5798  032570  010437  001254                 MOV     R4,@#$TMP11
 5799  032574  012737  033252  001240         MOV     #HHP9,@#$TMP3
 5800  032602  012737  033242  001242         MOV     #HHP8,@#$TMP4
 5801  032610  012737  033132  001244         MOV     #HHDATO,@#$TMP5
 5802  032616  012737  033232  001246         MOV     #HHP7,@#$TMP6
 5803  032624  104002                 1$:     ERROR   2
 5804  032626  000137  033302                 JMP     @#HHDONE
 5805  032632                         HHER7:
 5806  032632  010537  001252                 MOV     R5,@#$TMP10
 5807  032636  010437  001254                 MOV     R4,@#$TMP11
 5808  032642  012737  033242  001240         MOV     #HHP8,@#$TMP3
 5809  032650  012737  033242  001242         MOV     #HHP8,@#$TMP4
 5810  032656  012737  033132  001244         MOV     #HHDATO,@#$TMP5
 5811  032664  012737  033262  001246         MOV     #HHP10,@#$TMP6
 5812  032672  104002                 1$:     ERROR   2
 5813  032674  000137  033302                 JMP     @#HHDONE
 5814  032700                         HHER8:
 5815  032700  010537  001252                 MOV     R5,@#$TMP10
 5816  032704  010437  001254                 MOV     R4,@#$TMP11
 5817  032710  012737  033242  001240         MOV     #HHP8,@#$TMP3
 5818  032716  012737  033242  001242         MOV     #HHP8,@#$TMP4
 5819  032724  012737  033132  001244         MOV     #HHDATO,@#$TMP5
 5820  032732  012737  033262  001246         MOV     #HHP10,@#$TMP6
 5821  032740  104002                 1$:     ERROR   2
 5822  032742  000137  033302                 JMP     @#HHDONE
 5823  032746                         HHER9:
 5824  032746  010537  001252                 MOV     R5,@#$TMP10
 5825  032752  010437  001254                 MOV     R4,@#$TMP11
 5826  032756  012737  033212  001240         MOV     #HHP5,@#$TMP3
```

```
 5827  032764  012737  033212  001242          MOV     #HHP5,@#$TMP4
 5828  032772  012737  033132  001244          MOV     #HHDATO,@#$TMP5
 5829  033000  012737  033272  001246          MOV     #HHP11,@#$TMP6
 5830  033006  104002                  1$:     ERROR   2
 5831  033010  000137  033302                  JMP     @#HHDONE
 5832  033014                          HHER10:
 5833  033014  010537  001252                  MOV     R5,@#$TMP10
 5834  033020  010437  001254                  MOV     R4,@#$TMP11
 5835  033024  012737  033212  001240          MOV     #HHP5,@#$TMP3
 5836  033032  012737  033212  001242          MOV     #HHP5,@#$TMP4
 5837  033040  012737  033132  001244          MOV     #HHDATO,@#$TMP5
 5838  033046  012737  033272  001246          MOV     #HHP11,@#$TMP6
 5839  033054  104002                  1$:     ERROR   2
 5840  033056  000137  033302                  JMP     @#HHDONE
 5841  033062  013703  001236          HHTRAP: MOV     @#$TMP2,R3      ;WAS THE TRAP TO 244
 5842  033066  062703  000002                  ADD     #2,R3           ;ON THE INSTRUCTION
 5843  033072  020316                          CMP     R3,(SP)         ;BEING TESTED?
 5844  033074  001402                          BEQ     1$
 5845  033076  000137  062524                  JMP     @#FPSPUR
 5846  033102  011637  001236          1$:     MOV     (SP),@#$TMP2    ;FAILURE OF FPS INTERRUPT
 5847                                                                  ;DISABLE BIT (FID=1)
 5848  033106  022626                          CMP     (SP)+,(SP)+     ;TO INHIBIT TRAP.
 5849  033110  170201                          STFPS   R1
 5850  033112  010137  001240                  MOV     R1,@#$TMP3
 5851  033116  170301                          STST    R1
 5852  033120  010137  001242                  MOV     R1,@#$TMP4
 5853  033124  104002                  2$:     ERROR   2
 5854  033126  000137  033302                  JMP     @#HHDONE
 5855  033132  000000                  HHDATO: 0
 5856  033134  000000                          0
 5857  033136  000000                          0
 5858  033140  000000                          0
 5859  033142  000452                  HHP0:   452
 5860  033144  125252                          125252
 5861  033146  125252                          125252
 5862  033150  125253                          125253
 5863  033152  000252                  HHP1:   252
 5864  033154  125252                          125252
 5865  033156  125252                          125252
 5866  033160  125252                          125252
 5867  033162  000600                  HHP2:   600                     ;HHP0 + HHP1 WITH
 5868  033164  000000                          0                       ;PROPER NORMALIZATION
 5869  033166  000000                          0
 5870  033170  000000                          0
 5871  033172  000400                  HHP3:   400                     ;HHP0 + HHP1 WITH
 5872  033174  000000                          0                       ;BAD NORMALIZATION
 5873  033176  000000                          0
 5874  033200  000000                          0
 5875  033202  000000                  HHP4:   0
 5876  033204  000000                          0
 5877  033206  000000                          0
 5878  033210  000000                          0
 5879  033212  100200                  HHP5:   100200
 5880  033214  000000                          0
 5881  033216  000000                          0
 5882  033220  000000                          0
```

```
 5883  033222  000300                  HHP6:   300
 5884  033224  000000                          0
 5885  033226  000000                          0
 5886  033230  000000                          0
 5887  033232  100000                  HHP7:   100000                  ;HHP7 = HHP8 + HHP9
 5888  033234  000000                          0                       ;      = HHP5 + HHP6
 5889  033236  000000                          0
 5890  033240  000000                          0
 5891  033242  000200                  HHP8:   200
 5892  033244  000000                          0
 5893  033246  000000                          0
 5894  033250  000000                          0
 5895  033252  100300                  HHP9:   100300
 5896  033254  000000                          0
 5897  033256  000000                          0
 5898  033260  000000                          0
 5899  033262  000400                  HHP10:  400                     ;HHP10 = HHP8 + HHP8
 5900  033264  000000                          0
 5901  033266  000000                          0
 5902  033270  000000                          0
 5903  033272  100400                  HHP11:  100400                  ;HHP11 = HHP5 + HHP5
 5904  033274  000000                          0
 5905  033276  000000                          0
 5906  033300  000000                          0
 5907  033302                          HHDONE:
 5908  033302  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 5909                                                                  ;SEE IF THE USER HAS EXPRESSED
 5910                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 5911                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 5912                                                                  ;THE USER TYPED CONTROL G?).
 5913
 5914
 5915                                  ;;*****************************************************************
 5916                                  ;*TEST 40        OVER\UNDER TEST
 5917                                  ;*
 5918                                  ;*THIS IS A PARTIAL TEST OF THE OVER\UNDER
 5919                                  ;*FLOWS.  ONE OVERFLOW AND TWO UNDERFLOW
 5920                                  ;*CONDITIONS ARE CHECKED.  THE REMAINING
 5921                                  ;*UNDERFLOW COND. AND THE REMAINING OVERFLOW
 5922                                  ;*COND. WILL BE CHECKED LATER USING THE
 5923                                  ;*XXX INSTRUCTION.  HERE EACH CONDITION TESTED
 5924                                  ;*IS CHECKED BOTH WITH TRAPS ENABLED
 5925                                  ;*(FIU=1 OR FIV=1) AND ALSO WITH TRAPS
 5926                                  ;*DISABLED (FIU=0 OR FIV=0).
 5927                                  ;*
 5928                                  ;;*****************************************************************
 5929  033304  000004                  TST40:  SCOPE
 5930
 5931                                  ;TEST OVERFLOW CONDITION WITH TRAP DISABLER FIV=0
 5932  033306                          GG1:
 5933  033306  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 5934  033310  012704  000200                  MOV     #200,R4         ;CLEAR FIU, FIV, AND SET FD
 5935  033314  170104                          LDFPS   R4
 5936  033316  012737  033344  001236          MOV     #GG2,@#$TMP2
 5937  033324  012737  034376  000244          MOV     #GGER0,@#FPVECT
 5938  033332  012700  036150                  MOV     #GGP5,R0        ;SET AC0 OPERAND
```

```
5939  033336  172410                          LDD     (R0),AC0
5940  033340  012700  036150                  MOV     #GGP5,R0        ;FSRC
5941  033344  172010                  GG2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
5942  033346  170205                          STFPS   R5              ;GET FPS
5943  033350  012700  036100                  MOV     #GGDAT0,R0      ;GET THE RESULT
5944  033354  174010                          STD     AC0,(R0)
5945  033356  012701  036160                  MOV     #GGP6,R1        ;IS IT CORRECT
5946  033362  012702  000004                  MOV     #4,R2
5947  033366  022021                  GG3:    CMP     (R0)+,(R1)+
5948  033370  001402                          BEQ     GG4
5949  033372  000137  034474                  JMP     @#GGER1
5950  033376  077205                  GG4:    SOB     R2,GG3
5951  033400  052704  000006                  BIS     #6,R4           ;FPS CORRECT?
5952  033404  020405                          CMP     R4,R5
5953  033406  001402                          BEQ     GG5
5954  033410  000137  034542                  JMP     @#GGER2
5955                                          ;TEST OVERFLOW WITH TRAPS ENABLED
5956                                          ;FIV = 1
5957  033414                          GG5:
5958  033414  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
5959  033416  012704  001200                  MOV     #1200,R4        ;CLEAR FIU, SET FIV, AND FD
5960  033422  170104                          LDFPS   R4
5961  033424  012737  033452  001236          MOV     #GG6,@#$TMP2
5962  033432  012737  033470  000244          MOV     #GG7,@#FPVECT
5963  033440  012700  036150                  MOV     #GGP5,R0        ;SET AC0 OPERAND
5964  033444  172410                          LDD     (R0),AC0
5965  033446  012700  036150                  MOV     #GGP5,R0        ;FSPC
5966  033452  172010                  GG6:    ADDD    (R0),AC0        ;TEST INSTRUCTION
5967  033454  170000                          CFCC                    ;NO OVERFLOW TRAP OCCURED
5968  033456  012700  036100                  MOV     #GGDAT0,R0
5969  033462  174010                          STD     AC0,(R0)
5970  033464  000137  034610                  JMP     @#GGER3
5971  033470  013703  001236          GG7:    MOV     @#$TMP2,R3
5972  033474  062703  000002                  ADD     #2,R3
5973  033500  020316                          CMP     R3,(SP)         ;CHECK STACK DATA
5974  033502  001402                          BEQ     1$
5975  033504  000137  062524                  JMP     @#FPSPUR
5976  033510  011637  001236          1$:     MOV     (SP),@#$TMP2
5977  033514  022626                          CMP     (SP)+,(SP)+
5978  033516  170205                          STFPS   R5
5979  033520  012700  036100                  MOV     #GGDAT0,R0      ;GET THE RESULT
5980  033524  174010                          STD     AC0,(R0)
5981  033526  012701  036160                  MOV     #GGP6,R1        ;IS IT CORRECT
5982  033532  012702  000004                  MOV     #4,R2
5983  033536  022021                  GG8:    CMP     (R0)+,(R1)+
5984  033540  001402                          BEQ     GG9
5985  033542  000137  034656                  JMP     @#GGER4
5986  033546  077205                  GG9:    SOB     R2,GG8
5987  033550  052704  100006                  BIS     #100006,R4      ;EXACT ZERO RESULTED IF OVERFLOW
5988  033554  020405                          CMP     R4,R5           ;FPS CORRECT?, CHECK FER, FZ, FV
5989  033556  001402                          BEQ     1$
5990  033560  000137  034726                  JMP     @#GGER6
5991  033564  012704  000010          1$:     MOV     #10,R4
5992                                  ;CHECK FEC
5993  033570  170305                          STST    R5
5994  033572  020405                          CMP     R4,R5
```

```
 5995  033574  001402                        BEQ     GG10
 5996  033576  000137  034660                JMP     @#GGER5
 5997                                ;CHECK UNDER FLOW CONDITION WITH
 5998                                ;TRAPS DISABLED (FIU = 0)
 5999  033602                        GG10:
 6000  033602  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6001  033604  012704  000200                MOV     #0200,R4        ;SET FIU, FIV, AND FD
 6002  033610  170104                        LDFPS   R4
 6003  033612  012737  033640  001236        MOV     #GG11,@#$TMP2
 6004  033620  012737  034774  000244        MOV     #GGER7,@#FPVECT
 6005  033626  012700  036120                MOV     #GGP2,R0        ;SET AC0 OPERAND
 6006  033632  172410                        LDD     (R0),AC0        ;FSRC
 6007  033634  012700  036130                MOV     #GGP3,R0
 6008  033640  172010                GG11.   ADDD    (R0),AC0        ;TEST INSTRUCTION
 6009  033642  170205                        STFPS   R5              ;GET FPS
 6010  033644  012700  036100                MOV     #GGDATO,R0      ;GET THE RESULT
 6011  033650  174010                        STD     AC0,(R0)
 6012  033652  012701  036160                MOV     #GGP6,R1        ;IS IT CORRECT
 6013  033656  012702  000004                MOV     #4,R2
 6014  033662  022021                GG12:   CMP     (R0)+,(R1)+
 6015  033664  001402                        BEQ     GG13
 6016  033666  000137  035072                JMP     @#GGER8
 6017  033672  077205                GG13:   SOB     R2,GG12
 6018  033674  052704  000004                BIS     #4,R4           ;FPS CORRECT?
 6019  033700  020405                        CMP     R4,R5
 6020  033702  001402                        BEQ     GG14
 6021  033704  000137  035140                JMP     @#GGER9
 6022                                ;CHECK UNDERFLOW CONDITION WITH
 6023                                ;TRAP ENABLED (FIU = 1)
 6024  033710                        GG14:
 6025  033710  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6026  033712  012704  002200                MOV     #2200,R4        ;SET FIU, FIV, AND FD
 6027  033716  170104                        LDFPS   R4
 6028  033720  012737  033746  001236        MOV     #GG15,@#$TMP2
 6029  033726  012737  033764  000244        MOV     #GG16,@#FPVECT
 6030  033734  012700  036120                MOV     #GGP2,R0        ;SET AC0 OPERAND
 6031  033740  172410                        LDD     (R0),AC0        ;FSPC
 6032  033742  012700  036130                MOV     #GGP3,R0
 6033  033746  172010                GG15:   ADDD    (R0),AC0        ;TEST INSTRUCTION
 6034  033750  170000                        CFCC
 6035  033752  012700  036100                MOV     #GGDATO,R0
 6036  033756  174010                        STD     AC0,(R0)
 6037  033760  000137  035206                JMP     @#GGER10
 6038  033764  013703  001236        GG16:   MOV     @#$TMP2,R3
 6039  033770  062703  000002                ADD     #2,R3
 6040  033774  021603                        CMP     (SP),R3
 6041  033776  001402                        BEQ     1$
 6042  034000  000137  062524                JMP     @#FPSPUR
 6043  034004  011637  001236        1$:     MOV     (SP),@#$TMP2
 6044  034010  022626                        CMP     (SP)+,(SP)+
 6045  034012  170205                        STFPS   R5              ;GET FPS
 6046  034014  012700  036100                MOV     #GGDATO,R0      ;GET THE RESULT
 6047  034020  174010                        STD     AC0,(R0)
 6048  034022  012701  036170                MOV     #GGP7,R1        ;IS IT CORRECT
 6049  034026  012702  000004                MOV     #4,R2
 6050  034032  022021                GG17:   CMP     (R0)+,(R1)+
```

```
6051  034034  001402                    BEQ     GG18
6052  034036  000137  035254            JMP     @#GGER11
6053  034042  077205                GG18: SOB     R2,GG17
6054  034044  052704  100000            BIS     #100000,R4
6055  034050  020405                    CMP     R4,R5           ;FPS CORRECT?
6056  034052  001402                    BEQ     2$
6057  034054  000137  035322            JMP     @#GGER12
6058  034060                        2$:
6059  034060  012704  000012        1$: MOV     #12,R4
6060                                  ;CHECK FEC
6061  034064  170305                    STST    R5
6062  034066  020405                    CMP     R4,R5
6063  034070  001402                    BEQ     GG19
6064  034072  000177  001272            JMP     @GGER13
6065                                  ;CHECK UNDERFLOW CONDITION WITH TRAPS
6066                                  ;DISABLED (FIU = 0)
6067  034076                        GG19:
6068  034076  104414                    LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
6069  034100  012704  000200            MOV     #0200,R4        ;SET FIU, FIV, AND FD
6070  034104  170104                    LDFPS   R4
6071  034106  012737  034134  001236    MOV     #GG20,@#$TMP2
6072  034114  012737  035436  000244    MOV     #GGER14,@#FPVECT
6073  034122  012700  036120            MOV     #GGP2,R0        ;SET AC0 OPERAND
6074  034126  172410                    LDD     (R0),AC0
6075  034130  012700  036200            MOV     #GGP8,R0        ;FSPC
6076  034134  172010                GG20: ADDD    (R0),AC0        ;TEST INSTRUCTION
6077  034136  170205                    STFPS   R5              ;GET FPS
6078  034140  012700  036100            MOV     #GGDAT0,R0      ;GET THE RESULT
6079  034144  174010                    STD     AC0,(R0)
6080  034146  012701  036160            MOV     #GGP6,R1        ;IS IT CORRECT
6081  034152  012702  000004            MOV     #4,R2
6082  034156  022021                GG21: CMP     (R0)+,(R1)+
6083  034160  001402                    BEQ     GG22
6084  034162  000137  035534            JMP     @#GGER15
6085  034166  077205                GG22: SOB     R2,GG21
6086  034170  052704  000004            BIS     #4,R4           ;FPS CORRECT?
6087  034174  020405                    CMP     R4,R5
6088  034176  001402                    BEQ     GG23
6089  034200  000137  035602            JMP     @#GGER16
6090                                  ;CHECK UNDERFLOW CONDITION WITH TRAP
6091                                  ;ENABLED (FIU = 1)
6092  034204                        GG23:
6093  034204  104414                    LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
6094  034206  012704  002200            MOV     #2200,R4        ;SET FIU, FIV, AND FD
6095  034212  170104                    LDFPS   R4
6096  034214  012737  034242  001236    MOV     #GG24,@#$TMP2
6097  034222  012737  034260  000244    MOV     #GG25,@#FPVECT
6098  034230  012700  036120            MOV     #GGP2,R0        ;SET AC0 OPERAND
6099  034234  172410                    LDD     (R0),AC0
6100  034236  012700  036200            MOV     #GGP8,R0        ;FSRC
6101  034242  172010                GG24: ADDD    (R0),AC0        ;TEST INSTRUCTION
6102  034244  170000                    CFCC
6103  034246  012700  036100            MOV     #GGDAT0,R0
6104  034252  174010                    STD     AC0,(R0)
6105  034254  000137  035650            JMP     @#GGER17
6106  034260  013700  001236        GG25: MOV     @#$TMP2,R0
```

```
 6107  034264  062700  000002            ADD     #2,R0
 6108  034270  020016                    CMP     R0,(SP)
 6109  034272  001402                    BEQ     1$
 6110  034274  000137  062524            JMP     @#FPSPUR
 6111  034300  011637  001236    1$:     MOV     (SP),@#$TMP2
 6112  034304  022626                    CMP     (SP)+,(SP)+
 6113  034306  170205                    STFPS   R5              ;GET FPS
 6114  034310  012700  036100            MOV     #GGDAT0,R0      ;GET THE RESULT
 6115  034314  174010                    STD     AC0,(R0)
 6116  034316  012701  036210            MOV     #GGP9,R1        ;IS IT CORRECT
 6117  034322  012702  000004            MOV     #4,R2
 6118  034326  022021            GG26:   CMP     (R0)+,(R1)+
 6119  034330  001402                    BEQ     GG27
 6120  034332  000137  035716            JMP     @#GGER18
 6121  034336  077205            GG27:   SOB     R2,GG26
 6122  034340  052704  100004            BIS     #100004,R4
 6123  034344  020405                    CMP     R4,R5           ;FPS CORRECT?
 6124  034346  001402                    BEQ     1$
 6125  034350  000167  001456            JMP     GGER20
 6126  034354  012704  000012    1$:     MOV     #12,R4
 6127                                    ;CHECK FEC
```

```
 6128  034360  170305                          STST    R5
 6129  034362  020405                          CMP     R4,R5
 6130  034364  001402                          BEQ     GG28
 6131  034366  000167  001372                  JMP     GGER19
 6132
 6133  034372  000137  036220          GG28:   JMP     @#GGDONE
 6134
 6135  034376  013701  001236          GGER0:  MOV     @#$TMP2,R1
 6136  034402  062701  000002                  ADD     #2,R1
 6137  034406  020116                          CMP     R1,(SP)
 6138  034410  001402                          BEQ     10$
 6139  034412  000137  062524          5$:     JMP     @#FPSPUR
 6140  034416                          10$:
 6141  034416  170301                          STST    R1
 6142  034420  020127  000010                  CMP     R1,#10
 6143  034424  001372                          BNE     5$
 6144  034426  022626                          CMP     (SP)+,(SP)+
 6145  034430  012700  036100                  MOV     #GGDATO,R0
 6146  034434  174010                          STD     AC0,(R0)
 6147  034436  012737  036150  001240          MOV     #GGP5,@#$TMP3
 6148  034444  012737  036150  001242          MOV     #GGP5,@#$TMP4
 6149  034452  012737  036100  001244          MOV     #GGDATO,@#$TMP5
 6150  034460  012737  036160  001246          MOV     #GGP6,@#$TMP6
 6151  034466  104001                  1$:     ERROR   1
 6152  034470  000137  036220                  JMP     @#GGDONE
 6153
 6154  034474                          GGER1:
 6155  034474  010537  001252                  MOV     R5,@#$TMP10
 6156  034500  010437  001254                  MOV     R4,@#$TMP11
 6157  034504  012737  036150  001240          MOV     #GGP5,@#$TMP3
 6158  034512  012737  036150  001242          MOV     #GGP5,@#$TMP4
 6159  034520  012737  036100  001244          MOV     #GGDATO,@#$TMP5
 6160  034526  012737  036160  001246          MOV     #GGP6,@#$TMP6
 6161  034534  104002                  1$:     ERROR   2
 6162  034536  000137  036220                  JMP     @#GGDONE
 6163
 6164  034542                          GGER2:
 6165  034542  010537  001252                  MOV     R5,@#$TMP10
 6166  034546  010437  001254                  MOV     R4,@#$TMP11
 6167  034552  012737  036150  001240          MOV     #GGP5,@#$TMP3
 6168  034560  012737  036150  001242          MOV     #GGP5,@#$TMP4
 6169  034566  012737  036100  001244          MOV     #GGDATO,@#$TMP5
 6170  034574  012737  036160  001246          MOV     #GGP6,@#$TMP6
 6171  034602  104002                  1$:     ERROR   2
 6172  034604  000137  036220                  JMP     @#GGDONE
 6173
 6174  034610                          GGER3:
 6175  034610  010537  001252                  MOV     R5,@#$TMP10
 6176  034614  010437  001254                  MOV     R4,@#$TMP11
 6177  034620  012737  036150  001240          MOV     #GGP5,@#$TMP3
 6178  034626  012737  036150  001242          MOV     #GGP5,@#$TMP4
 6179  034634  012737  036100  001244          MOV     #GGDATO,@#$TMP5
 6180  034642  012737  036160  001246          MOV     #GGP6,@#$TMP6
 6181  034650  104002                  1$:     ERROR   2
 6182  034652  000137  036220                  JMP     @#GGDONE
 6183
```

```
 6184  034656  000706                    GGER4:  BR      GGER1
 6185
 6186  034660                            GGER5:
 6187  034660  010537  001252                    MOV     R5,@#$TMP10
 6188  034664  010437  001254                    MOV     R4,@#$TMP11
 6189  034670  012737  036150  001240            MOV     #GGP5,@#$TMP3
 6190  034676  012737  036150  001242            MOV     #GGP5,@#$TMP4
 6191  034704  012737  036100  001244            MOV     #GGDATO,@#$TMP5
 6192  034712  012737  036160  001246            MOV     #GGP6,@#$TMP6
 6193  034720  104002                    1$:     ERROR   2
 6194  034722  000137  036220                    JMP     @#GGDONE
 6195
 6196  034726                            GGER6:
 6197  034726  010537  001252                    MOV     R5,@#$TMP10
 6198  034732  010437  001254                    MOV     R4,@#$TMP11
 6199  034736  012737  036150  001240            MOV     #GGP5,@#$TMP3
 6200  034744  012737  036150  001242            MOV     #GGP5,@#$TMP4
 6201  034752  012737  036100  001244            MOV     #GGDATO,@#$TMP5
 6202  034760  012737  036160  001246            MOV     #GGP6,@#$TMP6
 6203  034766  104002                    1$:     ERROR   2
 6204  034770  000137  036220                    JMP     @#GGDONE
 6205
 6206  034774  013701  001236            GGER7:  MOV     @#$TMP2,R1
 6207  035000  062701  000002                    ADD     #2,R1
 6208  035004  020116                            CMP     R1,(SP)
 6209  035006  001402                            BEQ     10$
 6210  035010  000137  062524            5$:     JMP     @#FPSPUR
 6211  035014                            10$:
 6212  035014  170301                            STST    R1
 6213  035016  020127  000012                    CMP     R1,#12
 6214  035022  001372                            BNE     5$
 6215  035024  022626                            CMP     (SP)+,(SP)+
 6216  035026  012700  036100                    MOV     #GGDATO,R0
 6217  035032  174010                            STD     AC0,(R0)
 6218  035034  012737  036130  001240            MOV     #GGP3,@#$TMP3
 6219  035042  012737  036120  001242            MOV     #GGP2,@#$TMP4
 6220  035050  012737  036100  001244            MOV     #GGDATO,@#$TMP5
 6221  035056  012737  036160  001246            MOV     #GGP6,@#$TMP6
 6222  035064  104002                    1$:     ERROR   2
 6223  035066  000137  036220                    JMP     @#GGDONE
 6224
 6225  035072                            GGER8:
 6226  035072  010537  001252                    MOV     R5,@#$TMP10
 6227  035076  010437  001254                    MOV     R4,@#$TMP11
 6228  035102  012737  036130  001240            MOV     #GGP3,@#$TMP3
 6229  035110  012737  036120  001242            MOV     #GGP2,@#$TMP4
 6230  035116  012737  036100  001244            MOV     #GGDATO,@#$TMP5
 6231  035124  012737  036160  001246            MOV     #GGP6,@#$TMP6
 6232  035132  104002                    1$:     ERROR   2
 6233  035134  000137  036220                    JMP     @#GGDONE
 6234
 6235  035140                            GGER9:
 6236  035140  010537  001252                    MOV     R5,@#$TMP10
 6237  035144  010437  001254                    MOV     R4,@#$TMP11
 6238  035150  012737  036130  001240            MOV     #GGP3,@#$TMP3
 6239  035156  012737  036120  001242            MOV     #GGP2,@#$TMP4
```

```
6240  035164  012737  036100  001244          MOV     #GGDATO,a#$TMP5
6241  035172  012737  036160  001246          MOV     #GGP6,a#$TMP6
6242  035200  104002                  1$:     ERROR   2
6243  035202  000137  036220                  JMP     a#GGDONE
6244
6245  035206                          GGER10:
6246  035206  010537  001252                  MOV     R5,a#$TMP10
6247  035212  010437  001254                  MOV     R4,a#$TMP11
6248  035216  012737  036130  001240          MOV     #GGP3,a#$TMP3
6249  035224  012737  036120  001242          MOV     #GGP2,a#$TMP4
6250  035232  012737  036100  001244          MOV     #GGDATO,a#$TMP5
6251  035240  012737  036170  001246          MOV     #GGP7,a#$TMP6
6252  035246  104002                  1$:     ERROR   2
6253  035250  000137  036220                  JMP     a#GGDONE
6254
6255  035254                          GGER11:
6256  035254  010537  001252                  MOV     R5,a#$TMP10
6257  035260  010437  001254                  MOV     R4,a#$TMP11
6258  035264  012737  036130  001240          MOV     #GGP3,a#$TMP3
6259  035272  012737  036120  001242          MOV     #GGP2,a#$TMP4
6260  035300  012737  036100  001244          MOV     #GGDATO,a#$TMP5
6261  035306  012737  036170  001246          MOV     #GGP7,a#$TMP6
6262  035314  104002                  1$:     ERROR   2
6263  035316  000137  036220                  JMP     a#GGDONE
6264
6265  035322                          GGER12:
6266  035322  010537  001252                  MOV     R5,a#$TMP10
6267  035326  010437  001254                  MOV     R4,a#$TMP11
6268  035332  012737  036130  001240          MOV     #GGP3,a#$TMP3
6269  035340  012737  036120  001242          MOV     #GGP2,a#$TMP4
6270  035346  012737  036100  001244          MOV     #GGDATO,a#$TMP5
6271  035354  012737  036170  001246          MOV     #GGP7,a#$TMP6
6272  035362  104002                  1$:     ERROR   2
6273  035364  000137  036220                  JMP     a#GGDONE
6274
6275  035370                          GGER13:
6276  035370  010537  001252                  MOV     R5,a#$TMP10
6277  035374  010437  001254                  MOV     R4,a#$TMP11
6278  035400  012737  036130  001240          MOV     #GGP3,a#$TMP3
6279  035406  012737  036120  001242          MOV     #GGP2,a#$TMP4
6280  035414  012737  036100  001244          MOV     #GGDATO,a#$TMP5
6281  035422  012737  036170  001246          MOV     #GGP7,a#$TMP6
6282  035430  104002                  1$:     ERROR   2
6283  035432  000137  036220                  JMP     a#GGDONE
6284
6285  035436  013701  001236          GGER14: MOV     a#$TMP2,R1
6286  035442  062701  000002                  ADD     #2,R1
6287  035446  020116                          CMP     R1,(SP)
6288  035450  001402                          BEQ     10$
6289  035452  000137  062524          5$:     JMP     a#FPSPUR
6290  035456                          10$:
6291  035456  170301                          STST    R1
6292  035460  020127  000012                  CMP     R1,#12
6293  035464  001372                          BNE     5$
6294  035466  022626                          CMP     (SP)+,(SP)+
6295  035470  012700  036100                  MOV     #GGDATO,R0
```

```
6296  035474  174010                          STD     AC0,(R0)
6297
6298  035476  012737  036110  001240          MOV     #GGP1,a#$TMP3
6299  035504  012737  036130  001242          MOV     #GGP3,a#$TMP4
6300  035512  012737  036100  001244          MOV     #GGDAT0,a#$TMP5
6301  035520  012737  036160  001246          MOV     #GGP6,a#$TMP6
6302  035526  104002                  1$:     ERROR   2
6303  035530  000137  036220                  JMP     a#GGDONE
6304
6305  035534                          GGER15:
6306  035534  010537  001252                  MOV     R5,a#$TMP10
6307  035540  010437  001254                  MOV     R4,a#$TMP11
6308  035544  012737  036120  001240          MOV     #GGP2,a#$TMP3
6309  035552  012737  036200  001242          MOV     #GGP8,a#$TMP4
6310  035560  012737  036100  001244          MOV     #GGDAT0,a#$TMP5
6311  035566  012737  036160  001246          MOV     #GGP6,a#$TMP6
6312  035574  104002                  1$:     ERROR   2
6313  035576  000137  036220                  JMP     a#GGDONE
6314
6315  035602                          GGER16:
6316  035602  010537  001252                  MOV     R5,a#$TMP10
6317  035606  010437  001254                  MOV     R4,a#$TMP11
6318  035612  012737  036120  001240          MOV     #GGP2,a#$TMP3
6319  035620  012737  036200  001242          MOV     #GGP8,a#$TMP4
6320  035626  012737  036100  001244          MOV     #GGDAT0,a#$TMP5
6321  035634  012737  036160  001246          MOV     #GGP6,a#$TMP6
6322  035642  104002                  1$:     ERROR   2
6323  035644  000137  036220                  JMP     a#GGDONE
6324
6325  035650                          GGER17:
6326  035650  010537  001252                  MOV     R5,a#$TMP10
6327  035654  010437  001254                  MOV     R4,a#$TMP11
6328  035660  012737  036120  001240          MOV     #GGP2,a#$TMP3
6329  035666  012737  036200  001242          MOV     #GGP8,a#$TMP4
6330  035674  012737  036100  001244          MOV     #GGDAT0,a#$TMP5
6331  035702  012737  036210  001246          MOV     #GGP9,a#$TMP6
6332  035710  104002                  1$:     ERROR   2
6333  035712  000137  036220                  JMP     a#GGDONE
6334
6335  035716                          GGER18:
6336  035716  010537  001252                  MOV     R5,a#$TMP10
6337  035722  010437  001254                  MOV     R4,a#$TMP11
6338  035726  012737  036120  001240          MOV     #GGP2,a#$TMP3
6339  035734  012737  036200  001242          MOV     #GGP8,a#$TMP4
6340  035742  012737  036100  001244          MOV     #GGDAT0,a#$TMP5
6341  035750  012737  036210  001246          MOV     #GGP9,a#$TMP6
6342  035756  104002                  1$:     ERROR   2
6343  035760  000137  036220                  JMP     a#GGDONE
6344
6345  035764                          GGER19:
6346  035764  010537  001252                  MOV     R5,a#$TMP10
6347  035770  010437  001254                  MOV     R4,a#$TMP11
6348  035774  012737  036120  001240          MOV     #GGP2,a#$TMP3
6349  036002  012737  036200  001242          MOV     #GGP8,a#$TMP4
6350  036010  012737  036100  001244          MOV     #GGDAT0,a#$TMP5
6351  036016  012737  036210  001246          MOV     #GGP9,a#$TMP6
```

```
 6352  036024  104002                  1$:     ERROR   2
 6353  036026  000137  036220                  JMP     @#GGDONE
 6354
 6355  036032                          GGER20:
 6356  036032  010537  001252                  MOV     R5,@#$TMP10
 6357  036036  010437  001254                  MOV     R4,@#$TMP11
 6358  036042  012737  036120  001240          MOV     #GGP2,@#$TMP3
 6359  036050  012737  036200  001242          MOV     #GGP8,@#$TMP4
 6360  036056  012737  036100  001244          MOV     #GGDATO,@#$TMP5
 6361  036064  012737  036210  001246          MOV     #GGP9,@#$TMP6
 6362  036072  104002                  1$:     ERROR   2
 6363  036074  000137  036220                  JMP     @#GGDONE
 6364
 6365  036100  000000                  GGDATO: 0
 6366  036102  000000                          0
 6367  036104  000000                          0
 6368  036106  000000                          0
 6369
 6370  036110  000300                  GGP1:   300
 6371  036112  000000                          0
 6372  036114  000000                          0
 6373  036116  000000                          0
 6374  036120  100200                  GGP2:   100200
 6375  036122  000000                          0
 6376  036124  000000                          0
 6377  036126  000000                          0
 6378  036130  000200                  GGP3:   200
 6379  036132  000000                          0
 6380  036134  000000                          0
 6381  036136  000001                          1
 6382  036140  010200                  GGP4:   10200
 6383  036142  000000                          0
 6384  036144  000000                          0
 6385  036146  000000                          0
 6386  036150  077600                  GGP5:   77600                   ;OVER FLOW = GGP5 + GGP5
 6387  036152  000000                          0
 6388  036154  000000                          0
 6389  036156  000000                          0
 6390  036160  000000                  GGP6:   0                       ;OVERFLOW RESULT
 6391  036162  000000                          0                       ;UNDERFLOW RESULT
 6392  036164  000000                          0                       ;GGP6 = GGP4 + GGP5
 6393  036166  000000                          0                       ;     = GGP3 + GGP2 (FIU - 0)
 6394                                                                  ;     = GGP3 + GGP1
 6395  036170  062400                  GGP7:   62400                   ;GGP7 = GGP3 + GGP2 (FIU   1)
 6396  036172  000000                          0
 6397  036174  000000                          0
 6398  036176  000000                          0
 6399  036200  000340                  GGP8:   340
 6400  036202  000000                          0
 6401  036204  000000                          0
 6402  036206  000000                          0
 6403  036210  000100                  GGP9:   100
 6404  036212  000000                          0
 6405  036214  000000                          0
 6406  036216  000000                          0
 6407  036220                          GGDONE:
```

```
 6408  036220  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 6409                                                                    ;SEE IF THE USER HAS EXPRESSED
 6410                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
 6411                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 6412                                                                    ;THE USER TYPED CONTROL G?).
 6413
 6414
 6415                                  ;;*****************************************************************
 6416                                  ;*TEST 41       LDCFD AND LDCDF TEST
 6417                                  ;*
 6418                                  ;*THIS IS A TEST OF LDCFD AND LDCDF.
 6419                                  ;*
 6420                                  ;;*****************************************************************
 6421  036222  000004                  TST41:  SCOPE
 6422                                  ;TEST FOR CORRECT AUTO INCREMENT CONSTANT.
 6423  036224                          HX1:
 6424  036224  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6425  036226  012704  000200                  MOV     #200,R4         ;SET LONG INTEGER MODE
 6426  036232  170104                          LDFPS   R4
 6427  036234  012700  037640                  MOV     #HXP1,R0
 6428  036240  172410                          LDD     (R0),AC0
 6429  036242  012700  037650                  MOV     #HXP2,R0
 6430  036246  012737  036254  001236          MOV     #HX2,@#$TMP2
 6431  036254  177420                  HX2:    LDCFD   (R0)+,AC0
 6432  036256  020027  037654                  CMP     R0,#HXP2+4      ;;S R0 CORRECT
 6433  036262  001402                          BEQ     HX3
 6434  036264  000137  037216                  JMP     @#HXER1
 6435  036270                          HX3:
 6436  036270  170205                          STFPS   R5              ;GET FPS
 6437  036272  012700  037630                  MOV     #HXDAT0,R0
 6438  036276  174010                          STD     AC0,(R0)        ;GET AC0
 6439  036300  012701  037720                  MOV     #HXP7,R1        ;SEE IF RESULT IS
 6440  036304  012702  000004                  MOV     #4,R2           ;CORRECT
 6441  036310  022120                  HX4:    CMP     (R1)+,(R0)+
 6442  036312  001415                          BEQ     HX7
 6443  036314  012701  037650                  MOV     #HXP2,R1        ;DID FD GET
 6444  036320  012700  037630                  MOV     #HXDAT0,R0      ;COMPLIMENTED?
 6445  036324  012702  000004                  MOV     #4,R2
 6446  036330  022120                  HX5:    CMP     (R1)+,(R0)+
 6447  036332  001402                          BEQ     HX6
 6448  036334  000137  037256                  JMP     @#HXER2
 6449  036340  077205                  HX6.    SOB     R2,HX5
 6450  036342  000137  037306                  JMP     @#HXER3
 6451  036346  077220                  HX7:    SOB     R2,HX4
 6452  036350  012704  000200                  MOV     #200,R4         ;FPS CORRECT?
 6453  036354  020405                          CMP     R4,R5
 6454  036356  001402                          BEQ     HX8
 6455  036360  000137  037354                  JMP     @#HXER8
 6456                                  ;NOW    TEST    LDCDF
 6457  036364                          HX8:
 6458  036364  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6459  036366  012704  000200                          MOV #200,R4
 6460  036372  170104                          LDFPS R4
 6461
 6462  036374  012700  037640                  MOV     #HXP1,R0
 6463  036400  172410                          LDD     (R0),AC0
```

```
6464
6465  036402  012700  037650            MOV     #HXP2,R0
6466  036406  012737  036416  001236    MOV     #HX9,@#$TMP2
6467
6468  036414  170001                    SETF
6469
6470  036416  177420            HX9:    LDCDF   (R0)+,AC0       ;TEST INSTRUCTION
6471
6472  036420  020027  037660            CMP     R0,#HXP2+10     ;WAS A GOOD
6473  036424  001402                    BEQ     HX10            ;CONSTANT USED
6474  036426  000137  037236            JMP     @#HXER5         ;TO INCREMENT R0?
6475
6476  036432                    HX10:
6477  036432  170205                    STFPS   R5
6478  036434  012700  037630            MOV     #HXDAT0,R0
6479  036440  170011                    SETD
6480  036442  174010                    STD     AC0,(R0)        ;GET RESULT
6481  036444  012701  037730            MOV     #HXP8,R1
6482  036450  012702  000004            MOV     #4,R2
6483  036454  022120            HX11:   CMP     (R1)+,(R0)+     ;IS IT CORRECT?
6484  036456  001415                    BEQ     HX14
6485
6486  036460  012701  037720            MOV     #HXP7,R1
6487  036464  012700  037630            MOV     #HXDAT0,R0
6488  036470  012702  000004            MOV     #4,R2
6489  036474  022110            HX12:   CMP     (R1)+,(R0)      ;DID FD FAIL TO GET
6490  036476  001402                    BEQ     HX13            ;COMPLIMENTED?
6491  036500  000137  037372            JMP     @#HXER6
6492  036504  077205            HX13:   SOB     R2,HX12
6493  036506  000137  037422            JMP     @#HXER7
6494
6495  036512  077220            HX14:   SOB     R2,HX11
6496
6497  036514  012704  000000            MOV     #0,R4           ;FPS CORRECT?
6498  036520  020405                    CMP     R4,R5
6499  036522  001402                    BEQ     HX15
6500  036524  000137  037354            JMP     @#HXER8
6501
6502                            ;TEST GR7 IMMEDIATE MODE CONSTANT
6503
6504  036530                    HX15:
6505  036530  104414                    LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
6506
6507  036532  012704  000200            MOV     #200,R4
6508  036536  170104                    LDFPS   R4              ;SET FD
6509  036540  012737  036556  001236    MOV     #HX16,@#$TMP2
6510  036546  012737  037452  000004    MOV     #HXER9,@#ERRVECT
6511  036554  005001                    CLR     R1
6512  036556  177427  043243    HX16:   LDCFD   #5201,AC0
6513  036562  005201            HX165:  INC     R1
6514  036564  005201                    INC     R1
6515  036566  005201                    INC     R1
6516  036570  012737  062556  000004    MOV     #CPSPUR,@#ERRVECT
6517  036576  020127  000003            CMP     R1,#3           ;SEE IF PC WAS
6518  036602  001402                    BEQ     HX17            ;CORRECT
6519  036604  000137  037506            JMP     @#HXER10
```

```
6520
6521  036610                                HX17:
6522  036610  104414                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
6523
6524  036612  012704  000200                        MOV     #200,R4
6525  036616  170104                                LDFPS   R4
6526  036620  012737  036646  001236                MOV     #HX18,@#$TMP2
6527  036626  012700  037710                        MOV     #HXP6,R0
6528  036632  172410                                LDD     (R0),AC0
6529  036634  012737  062556  000004                MOV     #CPSPUR,@#ERRVECT
6530  036642  012700  037650                        MOV     #HXP2,R0
6531  036646  177410                    HX18:       LDCFD   (R0),AC0
6532
6533  036650  012700  037630                        MOV     #HXDATO,R0
6534  036654  174010                                STD     AC0,(R0)        ;GET RESULT.
6535  036656  012701  037720                        MOV     #HXP7,R1
6536  036662  012702  000004                        MOV     #4,R2
6537  036666  022021                    HX19:       CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
6538  036670  001402                                BEQ     HX20
6539  036672  000137  037256                        JMP     @#HXER2
6540  036676  077205                    HX20:       SOB     R2,HX19
6541
6542                                    ;TEST LDCFD WITH NEGATIVE OPERAND
6543  036700                            HX21:
6544  036700  104414                                LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
6545  036702  012704  000200                        MOV     #200,R4
6546  036706  170104                                LDFPS   R4
6547  036710  012737  036730  001236                MOV     #HX22,@#$TMP2
6548  036716  012700  037710                        MOV     #HXP6,R0
6549  036722  172410                                LDD     (R0),AC0
6550  036724  012700  037670                        MOV     #HXP4,R0
6551  036730  177410                    HX22:       LDCFD   (R0),AC0
6552
6553  036732  012700  037630                        MOV     #HXDATO,R0
6554  036736  174010                                STD     AC0,(R0)        ;GET RESULT
6555
6556  036740  012701  037700                        MOV     #HXP5,R1
6557  036744  012702  000004                        MOV     #4,R2
6558  036750  022120                    HX23:       CMP     (R1)+,(R0)+
6559  036752  001415                                BEQ     HX26
6560
6561  036754  012701  037720                        MOV     #HXP7,R1
6562  036760  012700  037630                        MOV     #HXDATO,R0
6563  036764  012702  000004                        MOV     #4,R2
6564  036770  022120                    HX24:       CMP     (R1)+,(R0)+     ;WAS SIGN INCORRECT
6565  036772  001402                                BEQ     HX25
6566  036774  000137  037540                        JMP     @#HXER11
6567  037000  077205                    HX25:       SOB     R2,HX24
6568  037002  000137  037560                        JMP     @#HXER12
6569
6570
6571  037006  077220                    HX26:       SOB     R2,HX23
6572
6573                                    ;TEST   LDCFD   0
6574
6575  037010                            HX27:
```

```
 6576  037010  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6577  037012  012704  000200                  MOV     #200,R4
 6578  037016  170104                          LDFPS   R4
 6579
 6580  037020  012700  037640                  MOV     #HXP1,R0
 6581  037024  172410                          LDD     (R0),AC0
 6582  037026  172010                          ADDD    (R0),AC0
 6583
 6584  037030  012737  037042  001236          MOV     #HX28,@#$TMP2
 6585  037036  012700  037640                  MOV     #HXP1,R0
 6586  037042  177410                  HX28:   LDCFD   (R0),AC0
 6587
 6588  037044  170205                          STFPS   R5
 6589
 6590  037046  012700  037630                  MOV     #HXDAT0,R0
 6591  037052  174010                          STD     AC0,(R0)        ;GET RESULT
 6592
 6593  037054  012701  037640                  MOV     #HXP1,R1
 6594  037060  012702  000004                  MOV     #4,R2
 6595  037064  022120                  HX29:   CMP     (R1)+,(R0)+     ;IS IT 0?
 6596  037066  001402                          BEQ     HX30
 6597  037070  000137  037610                  JMP     @#HXER13
 6598  037074  077205                  HX30:   SOB     R2,HX29
 6599
 6600  037076  012704  000204                  MOV     #204,R4         ;FPS CORRECT
 6601  037102  020405                          CMP     R4,R5
 6602  037104  001402                          BEQ     HX31
 6603  037106  000137  037336                  JMP     @#HXER4
 6604
 6605                                  ;TEST   LDCFD   0
 6606
 6607  037112                          HX31:
 6608  037112  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6609  037114  012704  000200                  MOV     #200,R4
 6610  037120  170104                          LDFPS   R4
 6611
 6612  037122  012700  037710                  MOV     #HXP6,R0
 6613  037126  172410                          LDD     (R0),AC0
 6614
 6615  037130  012737  037142  001236          MOV     #HX32,@#$TMP2
 6616  037136  012700  037640                  MOV     #HXP1,R0
 6617  037142  177410                  HX32:   LDCFD   (R0),AC0
 6618
 6619  037144  170205                          STFPS   R5
 6620
 6621  037146  012700  037630                  MOV     #HXDAT0,R0
 6622  037152  174010                          STD     AC0,(R0)        ;GET RESULT
 6623
 6624  037154  012701  037640                  MOV     #HXP1,R1
 6625  037160  012702  000004                  MOV     #4,R2
 6626  037164  022120                  HX33:   CMP     (R1)+,(R0)+     ;IS IT ZERO?
 6627  037166  001402                          BEQ     HX34
 6628  037170  000137  037610                  JMP     @#HXER13
 6629  037174  077205                  HX34:   SOB     R2,HX33
 6630
 6631  037176  012704  000204                  MOV     #204,R4         ;FPS CORRECT?
```

```
6632  037202  020405                        CMP     R4,R5
6633  037204  001402                        BEQ     HX35
6634  037206  000137  037336                JMP     @#HXER4
6635  037212  000137  037740        HX35:   JMP     @#HXDONE
6636
6637                                ;R0 INCORRECT
6638
6639  037216  012737  037654  001242  HXER1:  MOV     #HXP2+4,@#$TMP4
6640  037224  010037  001240                MOV     R0,@#$TMP3
6641  037230  104002                1$:     ERROR   2
6642  037232  000137  037740                JMP     @#HXDONE
6643
6644  037236  012737  037660  001242  HXER5:  MOV     #HXP2+10,@#$TMP4
6645  037244  010037  001240                MOV     R0,@#$TMP3
6646  037250  104002                1$:     ERROR   2
6647  037252  000137  037740                JMP     @#HXDONE
6648                                ;REPORT BAD DATA
6649  037256  012737  037650  001244  HXER2:  MOV     #HXP2,@#$TMP5
6650  037264  012737  037720  001250          MOV     #HXP7,@#$TMP7
6651  037272  012737  037630  001246  HXER22: MOV     #HXDATO,@#$TMP6
6652  037300  104002                1$:     ERROR   2
6653  037302  000137  037740                JMP     @#HXDONE
6654                                ;
6655  037306  012737  037650  001244  HXER3:  MOV     #HXP2,@#$TMP5
6656  037314  012737  037720  001250          MOV     #HXP7,@#$TMP7
6657  037322  012737  037630  001246  HXER33: MOV     #HXDATO,@#$TMP6
6658  037330  104002                1$:     ERROR   2
6659  037332  000137  037740                JMP     @#HXDONE
6660
6661  037336  010537  001240        HXER4:  MOV     R5,@#$TMP3
6662  037342  010437  001242                MOV     R4,@#$TMP4
6663  037346  104002                1$:     ERROR   2
6664  037350  000137  037740                JMP     @#HXDONE
6665
6666  037354  010537  001240        HXER8:  MOV     R5,@#$TMP3
6667  037360  010437  001242                MOV     R4,@#$TMP4
6668  037364  104002                1$:     ERROR   2
6669  037366  000137  037740                JMP     @#HXDONE
6670  037372  012737  037650  001244  HXER6:  MOV     #HXP2,@#$TMP5
6671  037400  012737  037730  001250          MOV     #HXP8,@#$TMP7
6672  037406  012737  037630  001246  HXER66: MOV     #HXDATO,@#$TMP6
6673  037414  104002                1$:     ERROR   2
6674  037416  000137  037740                JMP     @#HXDONE
6675
6676  037422  012737  037650  001244  HXER7:  MOV     #HXP2,@#$TMP5
6677  037430  012737  037730  001250          MOV     #HXP8,@#$TMP7
6678
6679  037436  012737  037630  001246          MOV     #HXDATO,@#$TMP6
6680  037444  104002                1$:     ERROR   2
6681  037446  000137  037740                JMP     @#HXDONE
6682
6683  037452  032716  000001        HXER9:  BIT     #1,(SP)         ;SEE IF IT
6684  037456  001005                        BNE     1$              ;AN ODD ADDRESS
6685  037460  022716  036562                CMP     #HX165,(SP)
6686  037464  001402                        BEQ     1$
6687  037466  000137  062556                JMP     @#CPSPUR
```

```
6688
6689  037472  011637  001236          1$:     MOV     (SP),@#$TMP2
6690  037476  022626                          CMP     (SP)+,(SP)+
6691  037500  104002                  2$:     ERROR   2
6692  037502  000137  037740                  JMP     @#HXDONE
6693
6694  037506  162701  000003          HXER10: SUB     #3,R1
6695  037512  006301                          ASL     R1
6696  037514  012702  036562                  MOV     #HX165,R2
6697  037520  010237  001242                  MOV     R2,@#$TMP4
6698  037524  160102                          SUB     R1,R2
6699  037526  010237  001240                  MOV     R2,@#$TMP3
6700  037532  104002                  1$:     ERROR   2
6701  037534  000137  037740                  JMP     @#HXDONE
6702
6703  037540  012737  037670  001244  HXER11: MOV     #HXP4,@#$TMP5
6704  037546  012737  037700  001250          MOV     #HXP5,@#$TMP7
6705  037554  000137  037272                  JMP     @#HXER22
6706  037560  012737  037670  001244  HXER12: MOV     #HXP4,@#$TMP5
6707  037566  012737  037700  001250          MOV     #HXP5,@#$TMP7
6708  037574  012737  037630  001246          MOV     #HXDATO,@#$TMP6
6709  037602  104002                  1$:     ERROR   2
6710  037604  000137  037740                  JMP     @#HXDONE
6711
6712  037610  012737  037640  001244  HXER13: MOV     #HXP1,@#$TMP5
6713  037616  012737  037640  001250          MOV     #HXP1,@#$TMP7
6714  037624  000137  037272                  JMP     @#HXER22
6715
6716  037630  000000                  HXDATO: 0
6717  037632  000000                          0
6718  037634  000000                          0
6719  037636  000000                          0
6720
6721  037640  000000                  HXP1:   0
6722  037642  000000                          0
6723  037644  000000                          0
6724  037646  000000                          0
6725
6726  037650  000577                  HXP2:   577
6727  037652  177776                          177776
6728  037654  177777                          177777
6729  037656  177776                          177776
6730  037660  005201                  HXP3:   5201
6731  037662  000000                          0
6732  037664  000000                          0
6733  037666  000000                          0
6734  037670  100577                  HXP4:   100577
6735  037672  177776                          177776
6736  037674  177777                          177777
6737  037676  177776                          177776
6738  037700  100577                  HXP5:   100577
6739  037702  177776                          177776
6740  037704  000000                          0
6741  037706  000000                          0
6742  037710  000252                  HXP6:   252
6743  037712  125252                          125252
```

```
 6744  037714  125252                          125252
 6745  037716  125252                          125252
 6746
 6747  037720  000577                HXP7:     577
 6748  037722  177776                          177776
 6749  037724  000000                          0
 6750  037726  000000                          0
 6751  037730  000577                HXP8:     577
 6752  037732  177777                          177777
 6753  037734  000000                          0
 6754  037736  000000                          0
 6755
 6756  037740                      HXDONE:
 6757  037740  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 6758                                                                  ;SEE IF THE USER HAS EXPRESSED
 6759                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 6760                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 6761                                                                  ;THE USER TYPED CONTROL G?).
 6762
 6763
 6764
 6765
 6766                                ;;*****************************************************************
 6767                                ;*TEST 42        CMPD TEST
 6768                                ;*
 6769                                ;*THIS IS A TEST OF THE CMPD INSTRUCTION. NOTE THAT A SUBROUTINE
 6770                                ;*IS USED TO SET UP OPERANDS, EXECUTE THE INSTRUCTION AND CHECK THE
 6771                                ;*RESULTS
 6772                                ;*
 6773                                ;;*****************************************************************
 6774  037742  000004                TST42:  SCOPE
 6775
 6776                                ;TEST THE CMPD INSTRUCTION WITH (FSRC=AC=0)
 6777  037744                        AAA1:
 6778  037744  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6779  037746  004737  040556                  JSR     PC,@#CMPSUB
 6780  037752  000000  000000  000000  1$:     .WORD   0,0,0,0                 ;AC0
 6781  037760  000000
 6782  037762  000000  000000  000000  2$:     .WORD   0,0,0,0                 ;FSRC
 6783  037770  000000
 6784  037772  000200                  3$:     200                             ;FPS BEFORE EXECUTION
 6785  037774  000204                          204                             ;FPS AFTER EXECUTION
 6786  037776  000200                          200                             ;ERROR FPS
 6787  040000  104002                  4$:     ERROR   2                       ;FPS ERROR
 6788
 6789
 6790                                ;TEST CMPD WITH (AC=0) AND FSRC POSITIVE.
 6791  040002                        AAA2:
 6792  040002  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6793  040004  004737  040556                  JSR     PC,@#CMPSUB
 6794  040010  000000  000000  000000  1$:     .WORD   0,0,0,0                 ;AC
 6795  040016  000000
 6796  040020  025252                  2$:     25252                           ;FSRC
 6797  040022  052525                          52525
 6798  040024  125252                          125252
 6799  040026  052525                          52525
```

```
 6800  040030  000200                    3$:     200                             ;FPS BEFORE EXECUTION
 6801  040032  000200                            200                             ;FPS AFTER EXECUTION
 6802  040034  000210                            210                             ;ERROR FPS
 6803  040036  104002                    4$:     ERROR   2
 6804
 6805                                    ;TEST CMPD WITH (AC=0) AND FSRC NEGATIVE
 6806  040040                            AAA3:
 6807  040040  104414                            LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 6808  040042  004737  040556                    JSR     PC,@#CMPSUB
 6809  040046  000000  000000  000000    1$:     .WORD   0,0,0,0                 ;AC
 6810  040054  000000
 6811  040056  125252                    2$:     125252                          ;FSRC
 6812  040060  125252                            125252
 6813  040062  052525                            52525
 6814  040064  125252                            125252
 6815  040066  000200                    3$:     200                             ;FPS BEFORE EXECUTION
 6816  040070  000210                            210                             ;FPS AFTER EXECUTION
 6817  040072  000200                            200                             ;ERROR FPS
 6818  040074  104002                    4$:     ERROR   2
 6819
 6820                                    ;TEST CMPD WITH (FSRC=0) AND AC POSITIVE
 6821  040076                            AAA4:
 6822  040076  104414                            LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 6823  040100  004737  040556                    JSR     PC,@#CMPSUB
 6824  040104  025252                    1$:     25252                           ;AC
 6825  040106  052525                            52525
 6826  040110  125252                            125252
 6827  040112  052525                            52525
 6828  040114  000000  000000  000000    2$:     .WORD   0,0,0,0                 ;FSRC
 6829  040122  000000
 6830  040124  000200                    3$:     200                             ;FPS BEFORE EXECUTION
 6831  040126  000210                            210                             ;FPS AFTER EXECUTION
 6832  040130  000200                            200                             ;ERROR FPS
 6833  040132  104002                    4$:     ERROR   2
 6834
 6835
 6836                                    ;TEST CMPD WITH (FSRC=0) AND AC NEGATIVE
 6837  040134                            AAA5:
 6838  040134  104414                            LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 6839  040136  004737  040556                    JSR     PC,@#CMPSUB
 6840  040142  125252                    1$:     125252                          ;AC
 6841  040144  125252                            125252
 6842  040146  052525                            52525
 6843  040150  125252                            125252
 6844  040152  000000  000000  000000    2$:     .WORD   0,0,0,0                 ;FSRC
 6845  040160  000000
 6846  040162  000200                    3$:     200                             ;FPS BEFORE EXECUTION
 6847  040164  000200                            200                             ;FPS AFTER EXECUTION
 6848  040166  000210                            210                             ;ERROR FPS
 6849  040170  104002                    4$:     ERROR   2
 6850
 6851                                    ;TEST CMPD WITH AC POSITIVE AND FSRC NEGATIVE
 6852  040172                            AAA6:
 6853  040172  104414                            LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 6854  040174  004737  040556                    JSR     PC,@#CMPSUB
 6855  040200  052525                    1$:     52525                           ;AC
```

```
6856  040202  125252                        125252
6857  040204  052525                        52525
6858  040206  125252                        125252
6859  040210  125252              2$:       125252                          ;;FSRC
6860  040212  052525                        52525
6861  040214  125252                        125252
6862  040216  052525                        52525
6863  040220  000200              3$:       200                             ;FPS BEFORE EXECUTION
6864  040222  000210                        210                             ;FPS AFTER EXECUTION
6865  040224  000200                        200                             ;ERROR FPS
6866  040226  104002              4$:       ERROR   2
6867
6868
6869                              ;TEST CMPD WITH AC NEGATIVE AND FSRC POSITIVE
6870  040230                      AAA7:
6871  040230  104414                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
6872  040232  004737  040556                JSR     PC,@#CMPSUB
6873  040236  125252              1$:       125252                          ;AC
6874  040240  052525                        52525
6875  040242  125252                        125252
6876  040244  052525                        52525
6877  040246  052525              2$:       52525                           ;FSRC
6878  040250  125252                        125252
6879  040252  052525                        52525
6880  040254  125252                        125252
6881  040256  000200              3$:       200                             ;FPS BEFORE EXECUTION
6882  040260  000200                        200                             ;FPS AFTER EXECUTION
6883  040262  000210                        210                             ;ERROR FPS
6884  040264  104002              4$:       ERROR   2
6885
6886                              ;TEST CMPD WITH AC POSITIVE AND FSRC POSITIVE
6887                              ;AND EAC LESS THAN EFSRC.
6888  040266                      AAA8:
6889  040266  104414                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
6890  040270  004737  040556                JSR     PC,@#CMPSUB
6891  040274  012345              1$:       12345                           ;AC
6892  040276  067654                        67654
6893  040300  032101                        32101
6894  040302  023456                        23456
6895  040304  023456              2$:       23456                           ;FSRC
6896  040306  076543                        76543
6897  040310  021012                        21012
6898  040312  034567                        34567
6899  040314  000200              3$:       200                             ;FPS BEFORE EXECUTION
6900  040316  000200                        200                             ;FPS AFTER EXECUTION
6901  040320  000210                        210                             ;ERROR FPS
6902  040322  104002              4$:       ERROR   2
6903
6904
6905                              ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE AND EAC GREATER THAN EFSRC
6906  040324                      AAA9:
6907  040324  104414                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
6908  040326  004737  040556                JSR     PC,@#CMPSUB
6909  040332  045676              1$:       45676                           ;AC
6910  040334  054321                        54321
6911  040336  012345                        12345
```

```
 6912  040340  067654                          67654
 6913  040342  034567                  2$:     34567                   ;FSRC
 6914  040344  065432                          65432
 6915  040346  101234                          101234
 6916  040350  056765                          56765
 6917  040352  000200                  3$:     200                     ;FPS BEFORE EXECUTION
 6918  040354  000210                          210                     ;FPS AFTER EXECUTION
 6919  040356  000200                          200                     ;ERROR FPS
 6920  040360  104002                  4$:     ERROR   2
 6921
 6922                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE AND AC EQUAL TO FSRC
 6923  040362                          AAA10:
 6924  040362  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6925  040364  004737  040556                  JSR     PC,@#CMPSUB
 6926  040370  012345                  1$:     12345                   ;AC
 6927  040372  067012                          67012
 6928  040374  034567                          34567
 6929  040376  012345                          012345
 6930  040400  012345                  2$:     12345                   ;FSRC
 6931  040402  067012                          67012
 6932  040404  034567                          34567
 6933  040406  012345                          012345
 6934  040410  000200                  3$:     200                     ;FPS BEFORE EXECUTION
 6935  040412  000204                          204                     ;FPS AFTER EXECUTION
 6936  040414  000200                          200                     ;ERROR FPS
 6937  040416  104002                  4$:     ERROR   2
 6938
 6939                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE, EAC EQUAL TO EFSRC,
 6940                                  ;AND FSRC GREATER THAN AC.
 6941  040420                          AAA11:
 6942  040420  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6943  040422  004737  040556                  JSR     PC,@#CMPSUB
 6944  040426  012345                  1$:     12345                   ;AC
 6945  040430  067012                          67012
 6946  040432  034567                          34567
 6947  040434  012345                          012345
 6948  040436  012345                  2$:     12345                   ;FSRC
 6949  040440  070123                          70123
 6950  040442  045670                          45670
 6951  040444  123456                          123456
 6952  040446  000200                  3$:     200                     ;FPS BEFORE EXECUTION
 6953  040450  000200                          200                     ;FPS AFTER EXECUTION
 6954  040452  000210                          210                     ;ERROR FPS
 6955  040454  104002                  4$:     ERROR   2
 6956
 6957                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE, EAC EQUAL TO EFSRC,
 6958                                  ;AND AC GREATER THAN FSRC.
 6959  040456                          AAA12:
 6960  040456  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6961  040460  004737  040556                  JSR     PC,@#CMPSUB
 6962  040464  054321                  1$:     54321                   ;AC
 6963  040466  076543                          76543
 6964  040470  021076                          21076
 6965  040472  054321                          54321
 6966  040474  054321                  2$:     54321                   ;FSRC
 6967  040476  065432                          65432
```

```
 6968  040500  107654                          107654
 6969  040502  032107                          32107
 6970  040504  000200                  3$:     200                             ;FPS BEFORE EXECUTION
 6971  040506  000210                          210                             ;FPS AFTER EXECUTION
 6972  040510  000200                          200                             ;ERROR FPS
 6973  040512  104002                  4$:     ERROR   2
 6974
 6975                                  ;TEST CMPD WITH AC NEGATIVE, FSRC NEGATIVE, EAC EQUAL TO EFSRC,
 6976                                  ;AND AC GREATER THAN FSRC
 6977  040514                          AAA13:
 6978  040514  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 6979  040516  004737  040556                  JSR     PC,@#CMPSUB
 6980  040522  112345                  1$:     112345                          ;AC
 6981  040524  043210                          43210
 6982  040526  076543                          76543
 6983  040530  021076                          21076
 6984  040532  112345                  2$:     112345                          ;FSRC
 6985  040534  054321                          54321
 6986  040536  007654                          07654
 6987  040540  032107                          32107
 6988  040542  000200                  3$:     200                             ;FPS BEFORE EXECUTION
 6989  040544  000210                          210                             ;FPS AFTER EXECUTION
 6990  040546  000200                          200                             ;ERROR FPS
 6991  040550  104002                  4$:     ERROR   2
 6992
 6993
 6994  040552  000137  040746                  JMP     @#AAADONE               ;FINISHED CMPD TEST.
 6995
 6996
 6997                                  ;THIS SUBROUTINE, CMPSUB, IS CALLED TO SET UP, EXECUTE
 6998                                  ;AND CHECK THE RESULTS OF A CMPD INSTRUCTION.
 6999                                  ;IT IS CALLED THUS:
 7000                                  ;
 7001                                  ;               JSR     PC,@#CMPSUB
 7002                                  ;       ACARG:  .WORD   X,X,X,X                 ;AC OPERAND
 7003                                  ;       FSRCARG:.WORD   X,X,X,X                 ;FSRC OPERAND
 7004                                  ;       FPSB:   .WORD   X                       ;FPS BEFORE EXECUTION
 7005                                  ;       FPSA:   .WORD   X                       ;FPS AFTER EXECUTION
 7006                                  ;       FPSE:   .WORD   X                       ;ERROR FPS
 7007                                  ;       ERR:    ERROR   X                       ;FPS ERROR
 7008                                  ;       CONT:                                   ;RETURN ADDRESS
 7009                                  ;
 7010                                  ;THE OPERANDS ARE SET UP (USING ACC FOR THE AC OPERAND). THEN
 7011                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, CMPD, IS EXECUTED.
 7012                                  ;AFTER THE EXECUTION THE FPS IS CHECKED AGAINST FPSA. IF IT IS A MATCH
 7013                                  ;THEN THERE WAS NO ERROR AND CONTROL IS RETURNED TO CONT. IF
 7014                                  ;THE FPS IS INCORRECT IT IS COMPARED WITH FPSE IN AN ATTEMPT TO ANALYSE
 7015                                  ;THE FAILURE. IF THE FPS IS THE SAME AS FPSE THEN CONTROL IS
 7016                                  ;RETURNED TO THE ERROR CALL AT LOCATION ERR. IF THE FPS WAS
 7017                                  ;NOT CORRECT BUT DIDN'T MATCH FPSE A GENERAL ERROR IS REPORTED
 7018                                  ;AND CONTROL IS PASSED TO CONT.
 7019
 7020  040556  012601                  CMPSUB: MOV     (SP)+,R1                ;PICK UP A POINTER TO THE
 7021                                                                          ;ARGUMENTS.
 7022  040560  016100  000020                  MOV     20(R1),R0               ;GET THE FPS BEFORE EXECUTION.
 7023  040564  170100                          LDFPS   R0                      ;LOAD IT INTO THE FPS.
```

```
7024
7025  040566  012737  040610  001236          MOV     #1$,@#$TMP2             ;SAVE ADDRESS OF CMPD INSTRUCTION.
7026  040574  010100                          MOV     R1,R0                   ;GET ADDRESS OF AC OPERAND.
7027  040576  172410                          LDD     (R0),AC0                ;LOAD AC0 OPERAND
7028
7029  040600  010100                          MOV     R1,R0                   ;COMPUTE FSRC OPERAND
7030  040602  062700  000010                  ADD     #10,R0                  ;ADDRESS
7031
7032  040606  000240                          NOP                             ;FOR SCOPING.
7033  040610  173410                  1$:     CMPD    (R0),AC0                ;EXECUTE THE TEST INSTRUCTION.
7034
7035  040612  170205                          STFPS   R5                      ;SAVE FPS AFTER INSTRUCTION.
7036
7037  040614  016104  000022                  MOV     22(R1),R4               ;GET EXPECTED FPS.
7038                                                                          ;IF INCORRECT SET UP FOR
7039  040620  010137  001240                  MOV     R1,@#$TMP3              ;AN ERROR CALL.
7040  040624  010137  001242                  MOV     R1,@#$TMP4
7041  040630  062737  000010  001242          ADD     #10,@#$TMP4
7042  040636  010537  001244                  MOV     R5,@#$TMP5
7043  040642  010437  001246                  MOV     R4,@#$TMP6
7044  040646  020405                          CMP     R4,R5                   ;WAS FPS CORRECT?
7045  040650  001410                          BEQ     3$                      ;BRANCH IF YES.
7046
7047
7048  040652  026105  000024                  CMP     24(R1),R5               ;WAS THE FPS THE SAME
7049                                                                          ;AS THE EXPECTED INCORRECT FPS?
7050  040656  001003                          BNE     2$                      ;BRANCH IF NO MATCH.
7051
7052  040660  062701  000026                  ADD     #26,R1                  ;IF THE EXPECTED INCORRECT
7053                                                                  ;FPS MATCHED THE RESULTANT FPS
7054  040664  000111                          JMP     (R1)                    ;RETURN TO THE ERROR CALL
7055                                                                          ;IN THE CALLING ROUTINE.
7056
7057  040666  104002                  2$:     ERROR   2                       ;OTHERWISE REPORT INCORRECT FPS
7058  040670  000411                          BR      5$
7059
7060  040672  012700  040736          3$:     MOV     #CMPTMP,R0              ;IF FPS WAS CORRECT MAKE SURE
7061  040676  174010                          STD     AC0,(R0)                ;AC0 WAS NOT AFFECTED BY CMPD.
7062  040700  010102                          MOV     R1,R2
7063  040702  012703  000004                  MOV     #4,R3
7064  040706  022220                  4$:     CMP     (R2)+,(R0)+
7065  040710  001003                          BNE     6$
7066  040712  077303                          SOB     R3,4$
7067
7068  040714  000161  000030          5$:     JMP     30(R1)                  ;RETURN
7069
7070  040720                          6$:                                     ;REPORT AC0 MODIFIED BY CMPD
7071  040720  010137  001240                  MOV     R1,@#$TMP3
7072  040724  012737  040736  001242          MOV     #CMPTMP,@#$TMP4
7073  040732  104002                  7$:     ERROR   2
7074  040734  000767                          BR      5$                      ;RETURN
7075
7076  040736  000000  000000  000000  CMPTMP: .WORD   0,0,0,0
7077  040744  000000
7078
7079
```

```
7080
7081  040746                          AAADONE:
7082  040746  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
7083                                                                  ;SEE IF THE USER HAS EXPRESSED
7084                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
7085                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7086                                                                  ;THE USER TYPED CONTROL G?).
7087
7088
7089
7090
7091                                  ;;*****************************************************************
7092                                  ;*TEST 43       DIVD WITH (FSRC=0) AND (BUT FD) TEST
7093                                  ;*
7094                                  ;*THIS IS A TEST OF THE DIVD INSTRUCTION WITH A
7095                                  ;*ZERO DIVISOR. THE CONDITION IS CHECKED WITH BOTH
7096                                  ;*TRAP ENABLED AND TRAPS DISABLED.
7097                                  ;*
7098                                  ;;*****************************************************************
7099  040750  000004                  TST43:  SCOPE
7100
7101                                  ;FIRST TEST DIVD WITH (FSRC=AC=0) AND TRAPS DISABLED.
7102  040752  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7103  040754  012704  040200          BBB0:   MOV     #40200,R4       ;SET UP FPS
7104                                                                  ;WITH INTERRUPTS
7105                                                                  ;DISABLED.
7106  040760  170104                          LDFPS   R4
7107  040762  012737  041224  000244          MOV     #BBBER1,@#FPVECT;SET UP FOR ANY FP INTERRUPTS.
7108  040770  012737  041010  001236          MOV     #BBB1,@#$TMP2
7109  040776  012700  041430                  MOV     #BBBP1,R0       ;SET UP AC0 = 0
7110  041002  172410                          LDD     (R0),AC0
7111  041004  012701  041430                  MOV     #BBBP1,R1       ;FSRC - 0
7112
7113  041010  174411                  BBB1:   DIVD    (R1),AC0        ;TEST INSTRUCTION
7114
7115  041012  170205                          STFPS   R5              ;GET FPS
7116  041014  170303                          STST    R3              ;GET FEC
7117
7118  041016  012704  140204                  MOV     #140204,R4      ;EXPECTED FPS.
7119  041022  020405                          CMP     R4,R5           ;IS FPS CORRECT.
7120  041024  001131                          BNE     BBBER2          ;IF INCORRECT BRANCH.
7121
7122  041026  012702  000004                  MOV     #4,R2           ;EXPECTED FEC.
7123  041032  020203                          CMP     R2,R3           ;IS FEC CORRECT?
7124  041034  001140                          BNE     BBBER3          ;IF INCORRECT BRANCH.
7125
7126                                  ;TEST DIVD WITH (FSRC=0) AND TRAPS DISABLED.
7127  041036                          BBB2:
7128  041036  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7129  041040  012704  040200                  MOV     #40200,R4       ;LOAD FPS WITH TRAPS DISABLED.
7130  041044  170104                          LDFPS   R4
7131
7132  041046  012737  041066  001236          MOV     #BBB3,@#$TMP2
7133  041054  012700  041440                  MOV     #BBBP2,R0       ;SET UP AC0 OPERAND (NON ZERO).
7134  041060  172410                          LDD     (R0),AC0
7135  041062  012700  041430                  MOV     #BBBP1,R0       ;FSRC-0
```

```
7136  041066  174410                  BBB3:   DIVD    (R0),AC0
7137
7138  041070  170205                          STFPS   R5              ;GET FPS.
7139  041072  170303                          STST    R3              ;GET FEC.
7140
7141  041074  012704  140200                  MOV     #140200,R4              ;EXPECTED FPS.
7142  041100  020405                          CMP     R4,R5           ;IS FPS CORRECT?
7143  041102  001102                          BNE     BBBER2          ;IF INCORRECT BRANCH.
7144
7145  041104  012702  000004                  MOV     #4,R2           ;EXPECTED FEC.
7146  041110  020203                          CMP     R2,R3           ;WAS FEC CORRECT?
7147  041112  001111                          BNE     BBBER3          ;IF INCORRECT BRANCH.
7148
7149                                  ;TEST DIVD WITH FSRC=0) AND TRAPS ENABLED.
7150  041114                          BBB4:
7151  041114  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7152  041116  012704  000200                  MOV     #200,R4         ;SET UP FPS. TRAP ENABLED.
7153  041122  170104                          LDFPS   R4
7154
7155  041124  012737  041152  001236          MOV     #BBB5,@#$TMP2
7156  041132  012700  041440                  MOV     #BBBP2,R0       ;SET UP AC0 OPERAND (NON ZERO).
7157  041136  172410                          LDD     (R0),AC0
7158
7159  041140  012737  041160  000244          MOV     #BBB6,@#FPVECT  ;SET UP FOR THE EXPECTED INTERRUPT.
7160  041146  012700  041430                  MOV     #BBBP1,R0       ;FSRC=0
7161
7162  041152  174410                  BBB5:   DIVD    (R0),AC0        ;TEST INSTRUCTION (SHOULD RESULT IN TRAP).
7163  041154  170000                          CFCC
7164
7165  041156  000502                          BR      BBBER4          ;GO REPORT FAILURE, NO TRAP.
7166
7167  041160  022716  041154          BBB6:   CMP     #BBB5+2,(SP)    ;TRAP TO HERE WHEN THE DIVISION BY 0
7168                                                                  ;OCCURS. FIRST SEE IF THE ADDRESS OF
7169                                                                  ;THE TRAP IS 2+THE ADDRESS OF THE TEST
7170                                                                  ;DIVD INSTRUCTION.
7171  041164  001402                          BEQ     1$
7172  041166  000137  062524                  JMP     @#FPSPUR        ;IF NOT THEN REPORT AN UNEXPECTED
7173                                                                  ;FP TRAP.
7174  041172  170205                  1$:     STFPS   R5              ;GET FPS.
7175  041174  170303                          STST    R3              ;GET FEC.
7176  041176  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK.
7177
7178  041200  012704  100200                  MOV     #100200,R4      ;EXPECTED FPS.
7179  041204  020405                          CMP     R4,R5           ;IS FPS CORRECT?
7180  041206  001040                          BNE     BBBER2          ;IF INCORRECT BRANCH.
7181
7182  041210  012702  000004                  MOV     #4,R2           ;EXPECTED FEC.
7183  041214  020203                          CMP     R2,R3           ;IS FEC CORRECT?
7184  041216  001047                          BNE     BBBER3          ;IF INCORRECT BRANCH.
7185
7186  041220  000137  041450                  JMP     @#BBBDONE       ;OTHERWISE GO TO NEXT TEST.
7187
7188
7189                                  ;TRAP HERE IF AN UNEXPECTED INTERRUPT OCCURS.
7190  041224  062737  000002  001236  BBBER1: ADD     #2,@#$TMP2      ;SEE IF THE INTERRUPT OCCURRED
7191                                                                  ;DURING THE EXECUTION OF THE DIVD
```

```
7192                                                                    ;INSTRUCTION BEING TESTED.
7193  041232  021637  001236                CMP     (SP),@#$TMP2
7194  041236  001402                        BEQ     1$
7195  041240  000137  062524                JMP     @#FPSPUR          ;IF NOT REPORT UNEXPECTED FP TRAP.
7196
7197  041244  022626                1$:     CMP     (SP)+,(SP)+       ;RESET THE STACK.
7198  041246  170303                        STST    R3                ;GET FEC.
7199  041250  170205                        STFPS   R5                ;GET FPS.
7200  041252  012737  000004  001240        MOV     #4,@#$TMP3        ;EXPECTED FEC.
7201  041260  010337  001242                MOV     R3,@#$TMP4
7202  041264  010537  001244                MOV     R5,@#$TMP5
7203  041270  010037  001250                MOV     R0,@#$TMP7
7204  041274  012737  140200  001246        MOV     #140200,@#$TMP6
7205  041302  104002                2$:     ERROR   2
7206                                                                    ;WITH TRAPS DISABLED.
7207  041304  000137  041450                JMP     @#BBBDONE
7208
7209                                ;REPORT FPS INCORRECT:
7210  041310  010537  001242        BBBER2: MOV     R5,@#$TMP4
7211  041314  010437  001244                MOV     R4,@#$TMP5
7212  041320  010037  001246                MOV     R0,@#$TMP6
7213  041324  010137  001250                MOV     R1,@#$TMP7
7214  041330  104002                1$:     ERROR   2
7215  041332  000137  041450                JMP     @#BBBDONE
7216
7217                                ;REPORT FEC INCORRECT:
7218  041336  010337  001242        BBBER3: MOV     R3,@#$TMP4
7219  041342  010237  001240                MOV     R2,@#$TMP3
7220  041346  010037  001246                MOV     R0,@#$TMP6
7221  041352  010137  001250                MOV     R1,@#$TMP7
7222  041356  104002                1$:     ERROR   2
7223  041360  000137  041450                JMP     @#BBBDONE
7224
7225                                ;REPORT NO TRAP OCCURRED AFTER TRYING TO DIVIDE
7226                                ;BY ZERO WITH ALL TRAPS ENABLED.
7227  041364  170303        BBBER4: STST    R3                ;GET FEC.
7228  041366  170205                STFPS   R5                ;GET FPS.
7229  041370  012737  000004  001242        MOV     #4,@#$TMP4
7230  041376  010337  001240                MOV     R3,@#$TMP3
7231  041402  010537  001244                MOV     R5,@#$TMP5
7232  041406  012737  100200  001246        MOV     #100200,@#$TMP6
7233  041414  010037  001250                MOV     R0,@#$TMP7
7234  041420  010137  001252                MOV     R1,@#$TMP10
7235  041424  104002                1$:     ERROR   2
7236  041426  000410                        BR      BBBDONE
7237
7238  041430  000000  000000  000000  BBBP1:  .WORD   0,0,0,0
7239  041436  000000
7240  041440  012345  054321  023456  BBBP2:  .WORD   12345,54321,23456,76543
7241  041446  076543
7242
7243
7244
7245  041450                        BBBDONE:
7246  041450  104413                        RSETUP                    ;GO INITIALIZE THE FPS AND STACK; AND
7247                                                                    ;SEE IF THE USER HAS EXPRESSED
```

```
7248                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
7249                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7250                                                    ;THE USER TYPED CONTROL G?).
7251
7252
7253
7254                            ;;**********************************************************************
7255                            ;*TEST 44        DIVF TEST
7256                            ;*
7257                            ;*THIS IS A TEST OF THE DIVF INSTRUCTION. NOTE THAT A SUBROUTINE IS
7258                            ;*USED TO SET UP THE OPERANDS, EXECUTE THE INSTRUCTION AND CHECK THE
7259                            ;*RESULTS.
7260                            ;*
7261                            ;;**********************************************************************
7262  041452  000004            TST44:  SCOPE
7263
7264                            ;CHECK DIVF WITH (AC=0).
7265  041454                    CCC1:
7266  041454  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7267  041456  004767  000552                   JSR     PC,DIVFSUB
7268  041462  000000  000000   1$:     .WORD   0,0                     ;AC
7269  041466  012345  067012   2$:     .WORD   12345,67012             ;FSRC
7270  041472  000000  000000   3$:     .WORD   0,0                     ;RES
7271  041476  000000           4$:     0                               ;FPS BEFORE EXECUTION.
7272  041500  000004                   4                               ;FPS AFTER EXECUTION
7273  041502  012345  067012   5$:     .WORD   12345,67012             ;ERROR RESULT
7274  041506  104002           6$:     ERROR   2
7275
7276                            ;TEST DIVF WITH AC POSITIVE, FSRC POSITIVE AND IN ROUND MODE.
7277  041510                    CCC2:
7278  041510  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7279  041512  004737  042234                   JSR     PC,@#DIVFSUB
7280  041516  065652  125252   1$:     .WORD   65652,125252            ;AC
7281  041522  065600  000000   2$:     .WORD   65600,0                 ;FSRC
7282  041526  040252  125252   3$:     .WORD   40252,125252            ;RES
7283  041532  003000           4$:     3000                            ;FPS BEFORE EXECUTION.
7284  041534  003000                   3000                            ;FPS AFTER EXECUTION.
7285  041536  040052  125252   5$:     .WORD   40052,125252            ;ERROR RESULT.
7286  041542  104002           6$:     ERROR   2
7287
7288                            ;TEST DIVF WITH AC POSITIVE, FSRC POSITIVE.
7289  041544                    CCC3:
7290  041544  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7291  041546  004767  000462                   JSR     PC,DIVFSUB
7292  041552  076400  000000   1$:     .WORD   76400,0                 ;AC
7293  041556  076400  000000   2$:     .WORD   76400,0                 ;FSRC
7294  041562  040200  000000   3$:     .WORD   40200,0                 ;RES
7295  041566  001000           4$:     1000                            ;FPS BEFORE EXECUTION.
7296  041570  001000                   1000                            ;FPS AFTER EXECUTION.
7297  041572  140200  000000   5$:     .WORD   140200,0                ;ERROR RES.
7298
7299  041576  104002           6$:     ERROR   2                       ;SIGN BAD
7300
7301                            ;TEST DIVF WITH BOTH OPERANDS POSITIVE.
7302  041600                    CCC4:
7303  041600  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
7304  041602  004737  042234                  JSR     PC,@#DIVFSUB
7305  041606  056777  177777          1$:     .WORD   56777,177777    ;AC
7306  041612  054200  000000          2$:     .WORD   54200,0         ;FSRC
7307  041616  042777  177777          3$:     .WORD   42777,177777    ;RES
7308  041622  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
7309  041624  000000                          0                       ;FPS AFTER EXECUTION.
7310  041626  002000  002000          5$:     .WORD   2000,2000       ;ERROR RES.
7311  041632  104002                  6$:     ERROR   2
7312
7313                                  ;TEST THE DIVF INSTRUCTION:
7314  041634                          CCC5:
7315  041634  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7316  041636  004737  042234                  JSR     PC,@#DIVFSUB
7317  041642  012377  177777          1$:     .WORD   12377,177777    ;AC
7318  041646  012300  000000          2$:     .WORD   12300,0         ;FSRC
7319  041652  040252  125252          3$:     .WORD   40252,125252    ;RES
7320  041656  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
7321  041660  000000                          0                       ;FPS AFTER EXECUTION.
7322  041662  177777  177777          5$:     .WORD   -1,-1           ;ERROR RES.
7323  041666  104002                  6$:     ERROR   2
7324
7325                                  ;TEST DIVIDE ALGORITHM. TEST ROUND CONSTANT.
7326  041670                          CCC6:
7327  041670  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7328  041672  004737  042234                  JSR     PC,@#DIVFSUB
7329  041676  064600  000001          1$:     .WORD   64600,1         ;AC
7330  041702  066600  000000          2$:     .WORD   66600,0         ;FSRC
7331  041706  036200  000001          3$:     .WORD   36200,1         ;RES
7332  041712  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
7333  041714  000000                          0                       ;FPS AFTER EXECUTION.
7334  041716  003000  003000          5$:     .WORD   3000,3000       ;ERROR RES.
7335  041722  104002                  6$:     ERROR   2
7336
7337                                  ;TEST DIVF.
7338  041724                          CCC7:
7339  041724  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
7340  041726  004737  042234              JSR     PC,@#DIVFSUB
7341  041732  034577  177776    1$:       .WORD   34577,177776    ;AC
7342  041736  023400  000000    2$:       .WORD   23400,0         ;FSRC
7343  041742  051377  177776    3$:       .WORD   51377,177776    ;RES
7344  041746  000017            4$:       17                      ;FPS BEFORE EXECUTION.
7345  041750  000000                      0                       ;FPS AFTER EXECUTION.
7346  041752  003400  003400    5$:       .WORD   3400,3400       ;ERROR RES.
7347  041756  104002            6$:       ERROR   2
7348
7349
7350                            ;DIVF TEST.
7351  041760                    CCC8:
7352  041760  104414                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7353  041762  004737  042234              JSR     PC,@#DIVFSUB
7354  041766  067652  125252    1$:       .WORD   67652,125252    ;AC
7355  041772  056500  000000    2$:       .WORD   56500,0         ;FSRC
7356  041776  051343  107070    3$:       .WORD   51343,107070    ;RES
7357  042002  000000            4$:       0                       ;FPS BEFORE EXECUTION.
7358  042004  000000                      0                       ;FPS AFTER EXECUTION.
7359  042006  051543  107070    5$:       .WORD   51543,107070    ;ERROR RES.
7360  042012  104002            6$:       ERROR   2               ;DIDN'T INCREMENT THE EXPONENT
7361                                                              ;AFTER DIVID NORMALIZATION.
7362
7363                            ;DIVF WITH AC NEGATIVE, FSRC NEGATIVE.
7364  042014                    CCC9:
7365  042014  104414                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7366  042016  004737  042234              JSR     PC,@#DIVFSUB
7367  042022  140400  000000    1$:       .WORD   140400,0        ;AC
7368  042026  140500  000000    2$:       .WORD   140500,0        ;FSRC
7369  042032  040052  125253    3$:       .WORD   040052,125253   ;RES
7370  042036  000000            4$:       0                       ;FPS BEFORE EXECUTION.
7371  042040  000000                      0                       ;FPS AFTER EXECUTION.
7372  042042  140052  125253    5$:       .WORD   140052,125253   ;ERROR RES.
7373  042046  104002            6$:       ERROR   2               ;BAD SIGN.
7374
7375                            ;DIVF WITH AC NEGATIVE AND FSRC POSITIVE.
7376  042050                    CCC10:
7377  042050  104414                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7378  042052  004737  042234              JSR     PC,@#DIVFSUB
7379  042056  160077  000000    1$:       .WORD   160077,0        ;AC
7380  042062  040277  000000    2$:       .WORD   40277,0         ;FSRC
7381  042066  160000  000000    3$:       .WORD   160000,0        ;RES
7382  042072  000007            4$:       7                       ;FPS BEFORE EXECUTION.
7383  042074  000010                      10                      ;FPS AFTER EXECUTION.
7384  042076  060000  000000    5$:       .WORD   60000,0         ;ERROR RES.
7385  042102  104002            6$:       ERROR   2               ;BAD SIGN.
7386
7387                            ;DIVF WITH AC POSITIVE AND FSRC NEGATIVE.
7388  042104                    CCC11:
7389  042104  104414                      LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7390  042106  004737  042234              JSR     PC,@#DIVFSUB
7391  042112  040400  000000    1$:       .WORD   40400,0         ;AC
7392  042116  140500  000000    2$:       .WORD   140500,0        ;FSRC
7393  042122  140052  125253    3$:       .WORD   140052,125253   ;RES
7394  042126  000017            4$:       17                      ;FPS BEFORE EXECUTION.
7395  042130  000010                      10                      ;FPS AFTER EXECUTION.
```

```
7396  042132  040052  125253          5$:     .WORD   40052,125253    ;ERROR RES.
7397  042136  104002                  6$:     ERROR   2               ;BAD SIGN.
7398
7399
7400                                  ;TEST DIVF BOTH OPERANDS POSITIVE AND TRUNCATE MODE.
7401  042140                          CCC12:
7402  042140  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7403  042142  004737  042234                  JSR     PC,@#DIVFSUB
7404  042146  060100  000001          1$:     .WORD   60100,1         ;AC
7405  042152  040300  000000          2$:     .WORD   40300,0         ;FSRC
7406  042156  060000  000000          3$:     .WORD   60000,0         ;RES
7407  042162  000052                  4$:     52                      ;FPS BEFORE EXECUTION.
7408  042164  000040                          40                      ;FPS AFTER EXECUTION.
7409  042166  060000  000001          5$:     .WORD   60000,1         ;ERROR RES.
7410  042172  104002                  6$:     ERROR   2               ;TRUNCATION ERROR
7411
7412                                  ;DIVF WITH POSITIVE OPERANDS AND ROUND MODE.
7413  042174                          CCC13:
7414  042174  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7415  042176  004767  000032                  JSR     PC,DIVFSUB
7416  042202  060100  000001          1$:     .WORD   60100,1         ;AC
7417  042206  040300  000000          2$:     .WORD   40300,0         ;FSRC
7418  042212  060000  000001          3$:     .WORD   60000,1         ;RES
7419  042216  000005                  4$:     5                       ;FPS BEFORE EXECUTION.
7420  042220  000000                          0                       ;FPS AFTER EXECUTION.
7421  042222  060000  000000          5$:     .WORD   60000,0         ;ERROR RES.
7422  042226  104002                  6$:     ERROR   2               ;ROUND ERROR.
7423
7424  042230  000137  042460                  JMP     @#CCCDONE       ;GO TO NEXT TEST.
7425
7426                                  ;THIS SUBROUTINE, DIVFSUB, IS CALLED TO SET UP, EXECUTE
7427                                  ;AND CHECK THE RESULT OF A DIVF INSTRUCTION. IT IS CALLED THUS:
7428                                  ;
7429                                  ;               JSR     PC,@#DIVFSUB
7430                                  ;       ACARG:  .WORD   X,X             ;AC OPERAND
7431                                  ;       FSRCARG:.WORD   X,X             ;FSRC OPERAND
7432                                  ;       RES:    .WORD   X,X             ;EXPECTED RESULT
7433                                  ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
7434                                  ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
7435                                  ;       ERRES:  .WORD   X,X             ;ERROR RESULT
7436                                  ;       ERR:    ERROR   X               ;RESULT ERROR
7437                                  ;       CONT:                           ;RETURN ADDRESS
7438                                  ;
7439                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
7440                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, DIVF IS EXECUTED.
7441                                  ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
7442                                  ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
7443                                  ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
7444                                  ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
7445                                  ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
7446                                  ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
7447                                  ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
7448                                  ;THEN THE FAILURE IS REPORTED IN DIVFSUB AND CONTROL IS PASSED TO
7449                                  ;CONT. IF NO ERRORS ARE DETECTED THEN DIVFSUB RETURNS CONTROL
7450                                  ;TO CONT.
7451
```

```
7452  042234  012601                  DIVFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
7453  042236  012700  000200                          MOV     #200,R0         ;SET FD MODE.
7454  042242  170100                                  LDFPS   R0
7455  042244  010100                                  MOV     R1,R0           ;LOAD THE AC OPERAND.
7456  042246  172410                                  LDD     (R0),AC0
7457  042250  016100  000014                          MOV     14(R1),R0       ;LOAD THE FPS
7458  042254  170100                                  LDFPS   R0
7459  042256  012737  042272  001236                  MOV     #1$,@#$TMP2
7460  042264  010100                                  MOV     R1,R0
7461  042266  062700  000004                          ADD     #4,R0           ;ESTABLISH A POINTER TO FSRC.
7462
7463  042272  174410                  1$:             DIVF    (R0),AC0        ;TEST INSTRUCTION.
7464
7465  042274  170204                                  STFPS   R4              ;GET THE FPS.
7466  042276  012700  000200                          MOV     #200,R0         ;SET FD MODE
7467  042302  170100                                  LDFPS   R0
7468
7469  042304  012700  042450                          MOV     #DIVFT,R0       ;GET THE RESULT OF THE DIVF.
7470  042310  174010                                  STD     AC0,(R0)
7471
7472  042312  010102                                  MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
7473  042314  010237  001240                          MOV     R2,@#$TMP3
7474  042320  062702  000004                          ADD     #4,R2
7475  042324  010237  001242                          MOV     R2,@#$TMP4
7476  042330  062702  000004                          ADD     #4,R2
7477  042334  010237  001244                          MOV     R2,@#$TMP5
7478  042340  012737  042450  001246                  MOV     #DIVFT,@#$TMP6
7479  042346  010437  001250                          MOV     R4,@#$TMP7
7480  042352  016137  000016  001252                  MOV     16(R1),@#$TMP10
7481
7482  042360  021061  000010                          CMP     (R0),10(R1)     ;IS THE RESULT CORRECT?
7483  042364  001011                                  BNE     10$             ;IF INCORRECT BRANCH.
7484  042366  026061  000002  000012                  CMP     2(R0),12(R1)
7485  042374  001005                                  BNE     10$
7486
7487  042376  026104  000016                          CMP     16(R1),R4       ;IS FPS CORRECT?
7488  042402  001020                                  BNE     15$             ;IF INCORRECT BRANCH.
7489  042404  000161  000026                          JMP     26(R1)          ;IF NO ERRORS OCCURRED RETURN.
7490
7491  042410  021061  000020          10$:            CMP     (R0),20(R1)     ;DOES THE INCORRECT RESULT
7492  042414  001010                                  BNE     11$             ;MATCH THE ANTICIPATED INCORRECT RESULT.
7493  042416  026061  000002  000022                  CMP     2(R0),22(R1)
7494  042424  001004                                  BNE     11$             ;BRANCH IF NO.
7495
7496  042426  010102                                  MOV     R1,R2           ;IT MATCHED SO RETURN TO THE ERROR
7497                                                                          ;REPORT AT THE CALLING ROUTINE.
7498  042430  062702  000024                          ADD     #24,R2
7499  042434  000112                                  JMP     (R2)
7500
7501  042436                          11$:                                    ;REPORT RESULT INCORRECT.
7502  042436  104002                  12$:            ERROR   2
7503  042440  000161  000026          13$:            JMP     26(R1)
7504
7505  042444                          15$:                                    ;REPORT FPS INCORPECT.
7506  042444  104002                  16$:            ERROR   2
7507  042446  000774                                  BR      13$
```

```
7508
7509  042450  000000  000000  000000  DIVFT:  .WORD   0,0,0,0
7510  042456  000000
7511
7512  042460                          CCCDONE:
7513  042460  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
7514                                                                  ;SEE IF THE USER HAS EXPRESSED
7515                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
7516                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7517                                                                  ;THE USER TYPED CONTROL G?).
7518
7519
7520
7521                                  ;;*****************************************************************
7522                                  ;*TEST 45       DIVD TEST
7523                                  ;*
7524                                  ;*THIS IS A TEST OF THE DIVD INSTRUCTION. NOTE THAT A SUBROUTINE IS
7525                                  ;*USED TO SET UP THE OPERANDS, EXECUTE THE INSTRUCTION AND CHECK THE RESULTS.
7526                                  ;*
7527                                  ;;*****************************************************************
7528  042462  000004                  TST45:  SCOPE
7529
7530                                  ;DIVD TEST WITH POSITIVE OPERANDS AND IN ROUND MODE.
7531  042464                          DDD1:
7532  042464  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7533  042466  004737  043154                  JSR     PC,@#DIVDSUB
7534  042472  034277  000000  000000  1$:     .WORD   34277,0,0,0     ;AC
7535  042500  000000
7536  042502  040277  000000  000000  2$:     .WORD   40277,0,0,0     ;FSRC
7537  042510  000000
7538  042512  034200  000000  000000  3$:     .WORD   34200,0,0,0     ;RES
7539  042520  000000
7540  042522  000200                  4$:     200                     ;FPS BEFORE EXECUTION.
7541  042524  000200                          200                     ;FPS AFTER EXECUTION.
7542  042526  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR RES.
7543  042534  177777
7544  042536  104002                  6$:     ERROR   2
7545
7546                                  ;DIVD WITH AC NEGATIVE AND FSRC POSITIVE IN TRUNCATE MODE.
7547  042540                          DDD2:
7548  042540  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7549  042542  004737  043154                  JSR     PC,@#DIVDSUB
7550  042546  134277  000000  000000  1$:     .WORD   134277,0,0,0    ;AC
7551  042554  000000
7552  042556  040277  000000  000000  2$:     .WORD   40277,0,0,0     ;FSRC
7553  042564  000000
7554  042566  134200  000000  000000  3$:     .WORD   134200,0,0,0    ;RES
7555  042574  000000
7556  042576  000207                  4$:     207                     ;FPS BEFORE EXECUTION.
7557  042600  000210                          210                     ;FPS AFTER EXECUTION.
7558  042602  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR RESULT.
7559  042610  177777
7560  042612  104002                  6$:     ERROR   2
7561
7562                                  ;DIVD TEST WITH OPERANDS BOTH NEGATIVE AND IN TRUNCATE MODE.
7563  042614                          DDD3:
```

```
7564  042614  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7565  042616  004767  000332                  JSR     PC,DIVDSUB
7566  042622  134300  000000  000000  1$:     .WORD   134300,0,0,1    ;AC
7567  042630  000001
7568  042632  140300  000000  000000  2$:     .WORD   140300,0,0,0    ;FSRC
7569  042640  000000
7570  042642  034200  000000  000000  3$:     .WORD   34200,0,0,0     ;RES
7571  042650  000000
7572  042652  000250                  4$:     250                     ;FPS BEFORE EXECUTION.
7573  042654  000240                          240                     ;FPS AFTER EXECUTION.
7574  042656  034200  000000  000000  5$:     .WORD   34200,0,0,1     ;ERROR RES.
7575  042664  000001
7576  042666  104002                  6$:     ERROR   2               ;TRUNCATION ERROR.
7577
7578                                  ;DIVD WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
7579  042670                          DDD4:
7580  042670  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7581  042672  004737  043154                  JSR     PC,@#DIVDSUB
7582  042676  034300  000000  000000  1$:     .WORD   34300,0,0,1     ;AC
7583  042704  000001
7584  042706  140300  000000  000000  2$:     .WORD   140300,0,0,0    ;FSRC
7585  042714  000000
7586  042716  134200  000000  000000  3$:     .WORD   134200,0,0,1    ;RES
7587  042724  000001
7588  042726  000207                  4$:     207                     ;FPS BEFORE EXECUTION.
7589  042730  000210                          210                     ;FPS AFTER EXECUTION.
7590  042732  134200  000000  000000  5$:     .WORD   134200,0,0,0    ;ERROR RES.
7591  042740  000000
7592  042742  104002                  6$:     ERROR   2               ;ROUND ERROR.
7593
7594                                  ;DIVD TEST.
7595  042744                          DDD5:
7596  042744  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7597  042746  004737  043154                  JSR     PC,@#DIVDSUB
7598  042752  100400  000000  000000  1$:     .WORD   100400,0,0,0    ;AC
7599  042760  000000
7600  042762  000500  000000  000000  2$:     .WORD   500,0,0,0       ;FSRC
7601  042770  000000
7602  042772  140052  125252          3$:     .WORD   140052,125252   ;RES
7603  042776  125252  125252                  .WORD   125252,125252
7604  043002  007647                  4$:     7647                    ;FPS BEFORE EXECUTION.
7605  043004  007650                          7650                    ;FPS AFTER EXECUTION.
7606  043006  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR RES.
7607  043014  177777
7608  043016  104002                  6$:     ERROR   2
7609
7610
7611                                  ;DIVD TEST WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
7612  043020                          DDD6:
7613  043020  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7614  043022  004737  043154                  JSR     PC,@#DIVDSUB
7615  043026  000400  000000  000000  1$:     .WORD   400,0,0,0       ;AC
7616  043034  000000
7617  043036  100500  000000  000000  2$:     .WORD   100500,0,0,0    ;FSRC
7618  043044  000000
7619  043046  140052  125252          3$:     .WORD   140052,125252   ;RES
```

```
7620  043052  125252  125253                  .WORD   125252,125253
7621  043056  007707                  4$:     7707                    ;FPS BEFORE EXECUTION.
7622  043060  007710                          7710                    ;FPS AFTER EXECUTION.
7623  043062  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR RES.
7624  043070  177777
7625  043072  104002                  6$:     ERROR   2
7626
7627                                  ;DIVD TEST.
7628  043074                          DDD7:
7629  043074  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7630  043076  004737  043154                  JSR     PC,@#DIVDSUB
7631  043102  170360  170360          1$:     .WORD   170360,170360   ;AC
7632  043106  170360  170360                  .WORD   170360,170360
7633  043112  170360  170360          2$:     .WORD   170360,170360   ;FSRC
7634  043116  170360  170360                  .WORD   170360,170360
7635  043122  040200  000000  000000  3$:     .WORD   40200,0,0,0     ;RES
7636  043130  000000
7637  043132  007717                  4$:     7717                    ;FPS BEFORE EXECUTION.
7638  043134  007700                          7700                    ;FPS AFTER EXECUTION.
7639  043136  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR RES.
7640  043144  177777
7641  043146  104002                  6$:     ERROR   2
7642
7643  043150  000137  043414                  JMP     @#DDDDONE       ;GO TO NEXT TEST.
7644
7645
7646                                  ;THIS SUBROUTINE, DIVDSUB, IS CALLED TO SET UP, EXECUTE
7647                                  ;AND CHECK THE RESULT OF A DIVD INSTRUCTION. IT IS CALLED THUS:
7648                                  ;
7649                                  ;               JSR     PC,@#DIVDSUB
7650                                  ;       ACARG:  .WORD   X,X,X,X                 ;AC OPERAND
7651                                  ;       FSRCARG:.WORD   X,X,X,X                 ;FSRC OPERAND
7652                                  ;       RES:    .WORD   X,X,X,X                 ;EXPECTED RESULT
7653                                  ;       FPSB:   .WORD   X                       ;FPS BEFORE EXECUTION
7654                                  ;       FPSA:   .WORD   X                       ;FPS AFTER EXECUTION
7655                                  ;       ERRES:  .WORD   X,X,X,X                 ;ERROR RESULT
7656                                  ;       ERR:    ERROR   X                       ;RESULT ERROR
7657                                  ;       CONT:                                   ;RETURN ADDRESS
7658                                  ;
7659                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
7660                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, DIVD IS EXECUTED.
7661                                  ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
7662                                  ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
7663                                  ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
7664                                  ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
7665
7666                                  ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
7667                                  ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
7668                                  ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
7669                                  ;THEN THE FAILURE IS REPORTED IN DIVDSUB AND CONTROL IS PASSED TO
7670                                  ;CONT. IF NO ERRORS ARE DETECTED THEN DIVDSUB RETURNS CONTPOL
7671                                  ;TO CONT.
7672
7673  043154  012601                  DIVDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
7674  043156  012700  000200                  MOV     #200,R0         ;SET FD MODE.
7675  043162  170100                          LDFPS   R0
```

```
7676
7677  043164  010100                          MOV     R1,R0           ;SET UP THE AC0 OPERAND.
7678  043166  172410                          LDD     (R0),AC0
7679  043170  016100  000030                 MOV     30(R1),R0       ;LOAD THE FPS.
7680  043174  170100                          LDFPS   R0
7681
7682  043176  012737  043212  001236          MOV     #1$,@#$TMP2
7683  043204  010100                          MOV     R1,R0           ;ESTABLISH A POINTER TO FSRC.
7684  043206  062700  000010                 ADD     #10,R0
7685
7686  043212  174410                  1$:     DIVD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
7687
7688  043214  170204                          STFPS   R4              ;GET THE FPS.
7689  043216  012700  000200                 MOV     #200,R0         ;SET FD MODE.
7690  043222  170100                          LDFPS   R0
7691
7692  043224  012700  043404                 MOV     #DIVDT,R0       ;GET THE RESULT.
7693  043230  174010                          STD     AC0,(R0)
7694
7695  043232  010102                          MOV     R1,R2           ;SAVE DATA IN CASE OF ERROR.
7696  043234  010237  001240                 MOV     R2,@#$TMP3
7697  043240  062702  000010                 ADD     #10,R2
7698  043244  010237  001242                 MOV     R2,@#$TMP4
7699  043250  062702  000010                 ADD     #10,R2
7700  043254  010237  001244                 MOV     R2,@#$TMP5
7701  043260  012737  043404  001246          MOV     #DIVDT,@#$TMP6
7702  043266  010437  001250                 MOV     R4,@#$TMP7
7703  043272  016137  000032  001252          MOV     32(R1),@#$TMP10
7704
7705  043300  010102                          MOV     R1,R2           ;CHECK THE RESULT.
7706  043302  062702  000020                 ADD     #20,R2
7707  043306  012703  043404                 MOV     #DIVDT,R3
7708  043312  012705  000004                 MOV     #4,R5
7709  043316  022223                  2$:     CMP     (R2)+,(R3)+
7710  043320  001006                          BNE     10$             ;BRANCH IF RESULT INCORRECT.
7711  043322  077503                          SOB     R5,2$
7712
7713  043324  026104  000032                 CMP     32(R1),R4       ;IS FPS CORRECT?
7714  043330  001023                          BNE     15$             ;BRANCH IF INCORRECT.
7715  043332  000161  000046                 JMP     46(R1)          ;RETURN.
7716
7717  043336  010102                  10$:    MOV     R1,R2           ;WAS INCORRECT RESULT ANTICIPATED?
7718  043340  062702  000034                 ADD     #34,R2
7719  043344  012703  043404                 MOV     #DIVDT,R3
7720  043350  012705  000004                 MOV     #4,R5
7721  043354  022223                  11$:    CMP     (R2)+,(R3)+
7722  043356  001005                          BNE     12$             ;BRANCH IF NO.
7723  043360  077503                          SOB     R5,11$
7724  043362  010102                          MOV     R1,R2           ;IF THE INCORRECT RESULT WAS
7725  043364  062702  000044                 ADD     #44,R2          ;ANTICIPATED RETURN TO THE
7726                                                                  ;ERROR REPORT IN THE CALLING
7727  043370  000112                          JMP     (R2)            ;ROUTINE.
7728
7729  043372                          12$:                            ;REPORT RESULT INCORRECT.
7730  043372  104002                  13$:    ERROR   2
7731  043374  000161  000046          14$:    JMP     46(R1)
```

```
7732
7733  043400                                15$:                                ;REPORT FPS INCORRECT.
7734  043400  104002                        16$:    ERROR   2
7735  043402  000774                                BR      14$
7736
7737  043404  000000  000000  000000  DIVDT:  .WORD   0,0,0,0
7738  043412  000000
7739
7740  043414                                DDDDONE:
7741  043414  104413                                RSETUP                      ;GO INITIALIZE THE FPS AND STACK; AND
7742                                                                            ;SEE IF THE USER HAS EXPRESSED
7743                                                                            ;THE DESIRE TO CHANGE THE SOFTWARE
7744                                                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7745                                                                            ;THE USER TYPED CONTROL G?).
7746
7747
7748
7749                                ;;**************************************************************************
7750                                ;*TEST 46        MULF TEST
7751                                ;*
7752                                ;*THIS IS A TEST OF THE MULF INSTRUCTION. IT MAKES USE OF A SUBROUTINE
7753                                ;*TO SET UP THE OPERANDS, EXECUTE THE MULF INSTRUCTION AND CHECK THE
7754                                ;*RESULTS.
7755                                ;*
7756                                ;;**************************************************************************
7757  043416  000004                TST46:  SCOPE
7758
7759                                ;MULF WITH (FSRC=AC=0)
7760  043420                        EEE1:
7761  043420  104414                        LPERR                               ;SET UP THE LOOP ON ERROR ADDRESS.
7762  043422  004737  044200                JSR     PC,@#MULFSUB
7763  043426  000000  000000        1$:     .WORD   0,0                         ;AC
7764  043432  000000  000000        2$:     .WORD   0,0                         ;FSRC
7765  043436  000000  000000        3$:     .WORD   0,0                         ;RES
7766  043442  007517                4$:     7517                                ;FPS BEFORE EXECUTION.
7767  043444  007504                        7504                                ;FPS AFTER EXECUTION.
7768  043446  177777  177777        5$:     .WORD   -1,-1
7769  043452  104002                6$:     ERROR   2
7770
7771                                ;MULF WITH (FSRC=0).
7772  043454                        EEE2:
7773  043454  104414                        LPERR                               ;SET UP THE LOOP ON ERROR ADDRESS.
7774  043456  004737  044200                JSR     PC,@#MULFSUB
7775  043462  071625  034435        1$:     .WORD   71625,34435                 ;AC
7776  043466  000000  000000        2$:     .WORD   0,0                         ;FSRC
7777  043472  000000  000000        3$:     .WORD   0,0                         ;RES
7778  043476  000013                4$:     13                                  ;FPS BEFORE EXECUTION.
7779  043500  000004                        4                                   ;FPS AFTER EXECUTION.
7780  043502  177777  177777        5$:     .WORD   -1,-1                       ;ERROR RES.
7781
7782  043506  104002                6$:     ERROR   2
7783
7784                                ;MULF WITH (AC=0)
7785  043510                        EEE3:
7786  043510  104414                        LPERR                               ;SET UP THE LOOP ON ERROR ADDRESS.
7787  043512  004737  044200                JSR     PC,@#MULFSUB
```

```
7788  043516  000000  000000          1$:     .WORD   0,0                     ;AC
7789  043522  071625  153443          2$:     .WORD   071625,153443           ;FSRC
7790  043526  000000  000000          3$:     .WORD   0,0                     ;RES
7791  043532  007500                  4$:     7500                            ;FPS BEFORE EXECUTION.
7792  043534  007504                          7504                            ;FPS AFTER EXECUTION.
7793  043536  177777  177777          5$:     .WORD   -1,-1                   ;ERROR RES.
7794  043542  104002                  6$:     ERROR   2
7795
7796                                  ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN ROUND MODE.
7797  043544                          EEE4:
7798  043544  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
7799  043546  004737  044200                  JSR     PC,@#MULFSUB
7800  043552  040200  000000          1$:     .WORD   40200,0                 ;AC
7801  043556  040177  177777          2$:     .WORD   40177,-1                ;FSRC
7802  043562  040177  177777          3$:     .WORD   40177,-1                ;RES
7803  043566  000017                  4$:     17                              ;FPS BEFORE EXECUTION.
7804  043570  000000                          0                               ;FPS AFTER EXECUTION.
7805  043572  140177  177777          5$:     .WORD   140177,-1               ;ERROR RES.
7806  043576  104002                  6$:     ERROR   2                       ;BAD SIGN.
7807
7808                                  ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN TRUNCATE MODE.
7809  043600                          EEE5:
7810  043600  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
7811  043602  004767  000372                  JSR     PC,MULFSUB
7812  043606  040177  177777          1$:     .WORD   40177,-1                ;AC
7813  043612  040200  000000          2$:     .WORD   40200,0                 ;FSRC
7814  043616  040177  177777          3$:     .WORD   40177,-1                ;RES
7815  043622  000040                  4$:     40                              ;FPS BEFORE EXECUTION.
7816  043624  000040                          40                              ;FPS AFTER EXECUTION.
7817  043626  037777  177777          5$:     .WORD   37777,-1                ;ERROR RES.
7818  043632  104002                  6$:     ERROR   2                       ;ST 252 TO 044 INTO 444 (BUT Y62)
7819                                                                          ;MUL. NORMALIZATION FAILURE.
7820
7821                                  ;MULF WITH BOTH OPERANDS POSITIVE NORMALIZE TEST.
7822  043634                          EEE6:
7823  043634  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
7824  043636  004737  044200                  JSR     PC,@#MULFSUB
7825  043642  040100  000000          1$:     .WORD   40100,0                 ;AC
7826  043646  040100  000000          2$:     .WORD   40100,0                 ;FSRC
7827  043652  040020  000000          3$:     .WORD   40020,0                 ;RES
7828  043656  000012                  4$:     12                              ;FPS BEFORE EXECUTION.
7829  043660  000000                          0                               ;FPS AFTER EXECUTION.
7830  043662  042040  000000          5$:     .WORD   42040,0                 ;ERROR RES.
7831  043666  104002                  6$:     ERROR   2                       ;ST 252 TO 444 INTO 042 (BUT Y62)
7832                                                                          ;MUL. NORMALIZATION FAILURE.
7833
7834                                  ;MULF WITH BOTH OPERANDS POSITIVE IN ROUND MODE.
7835  043670                          EEE7:
7836  043670  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
7837  043672  004737  044200                  JSR     PC,@#MULFSUB
7838  043676  017500  000000          1$:     .WORD   17500,0                 ;AC
7839  043702  023652  125252          2$:     .WORD   23652,125252            ;FSRC
7840  043706  003177  177777          3$:     .WORD   3177,-1                 ;RES
7841  043712  007417                  4$:     7417                            ;FPS BEFORE EXECUTION.
7842  043714  007400                          7400                            ;FPS AFTER EXECUTION.
7843  043716  177777  177777          5$:     .WORD   -1,-1
```

```
7844  043722  104002                  6$:     ERROR   2
7845
7846                                  ;MULF WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
7847  043724                          EEE8:
7848  043724  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7849  043726  004737  044200                  JSR     PC,@#MULFSUB
7850  043732  040342  000000          1$:     .WORD   40342,0         ;AC
7851  043736  176542  000000          2$:     .WORD   176542,0        ;FSRC
7852  043742  176707  102000          3$:     .WORD   176707,102000   ;RES
7853  043746  000007                  4$:     7                       ;FPS BEFORE EXECUTION.
7854  043750  000010                          10                      ;FPS AFTER EXECUTION.
7855  043752  076507  102000          7$:     .WORD   76507,102000    ;ERROR RES.
7856  043756  104002                  6$:     ERROR   2               ;BAD SIGN.
7857
7858                                  ;MULF WITH AC NEGATIVE AND FSRC POSITIVE IN ROUND MODE.
7859  043760                          EEE9:
7860  043760  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7861  043762  004737  044200                  JSR     PC,@#MULFSUB
7862  043766  140200  000000          1$:     .WORD   140200,0        ;AC
7863  043772  007417  007417          2$:     .WORD   7417,7417       ;FSRC
7864  043776  107417  007417          3$:     .WORD   107417,7417     ;RES
7865  044002  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
7866  044004  000010                          10                      ;FPS AFTER EXECUTION.
7867  044006  007417  007417          5$:     .WORD   7417,7417       ;ERROR RES.
7868  044012  104002                  6$:     ERROR   2               ;BAD SIGN.
7869
7870                                  ;MULF WITH BOTH OPERANDS NEGATIVE IN ROUND MODE.
7871  044014                          EEE10:
7872  044014  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7873  044016  004737  044200                  JSR     PC,@#MULFSUB
7874  044022  144600  000000          1$:     .WORD   144600,0        ;AC
7875  044026  154000  000000          2$:     .WORD   154000,0        ;FSRC
7876  044032  060400  000000          3$:     .WORD   60400,0         ;RES
7877  044036  000017                  4$:     17                      ;FPS BEFORE EXECUTION.
7878  044040  000000                          0                       ;FPS AFTER EXECUTION.
7879  044042  160400  000000          5$:     .WORD   160400,0        ;ERROR RES.
7880  044046  104002                  6$:     ERROR   2               ;BAD SIGN.
7881
7882                                  ;MULF BOTH OPERANDS NEGATIVE IN ROUND MODE.
7883  044050                          EEE11:
7884  044050  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7885  044052  004737  044200                  JSR     PC,@#MULFSUB
7886  044056  140300  000000          1$:     .WORD   140300,0        ;AC
7887  044062  160000  000001          2$:     .WORD   160000,1        ;FSRC
7888  044066  060100  000002          3$:     .WORD   60100,2         ;RES
7889  044072  000010                  4$:     10                      ;FPS BEFORE EXECUTION.
7890  044074  000000                          0                       ;FPS AFTER EXECUTION.
7891  044076  060100  000001          5$:     .WORD   60100,1         ;ERROR RES.
7892  044102  104002                  6$:     ERROR   2               ;ROUND FAILURE.
7893
7894                                  ;MULF WITH AC POSITIVE AND FSRC NEGATIVE IN TRUNCATE MODE.
7895  044104                          EEE12:
7896  044104  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7897  044106  004737  044200                  JSR     PC,@#MULFSUB
7898  044112  060000  000001          1$:     .WORD   60000,1         ;AC
7899  044116  140300  000000          2$:     .WORD   140300,0        ;FSRC
```

```
7900  044122  160100  000001          3$:     .WORD   160100,1        ;RES
7901  044126  007547                  4$:     7547                    ;FPS BEFORE EXECUTION.
7902  044130  007550                          7550                    ;FPS AFTER EXECUTION.
7903  044132  160100  000001          5$:     .WORD   160100,1        ;ERROR RES.
7904  044136  104002                  6$:     ERROR   2               ;TRUNCATION ERROR.
7905
7906                                  ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN ROUND MODE.
7907  044140                          EEE13:
7908  044140  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
7909  044142  004737  044200                  JSR     PC,@#MULFSUB
7910  044146  040277  000000          1$:     .WORD   40277,0         ;AC
7911  044152  060000  000001          2$:     .WORD   60000,1         ;FSRC
7912  044156  060077  000001          3$:     .WORD   60077,1         ;RES
7913  044162  000014                  4$:     14                      ;FPS BEFORE EXECUTION.
7914  044164  000000                          0                       ;FPS AFTER EXECUTION.
7915  044166  060077  000002          5$:     .WORD   60077,2         ;ERROR RES.
7916  044172  104002                  6$:     ERROR   2               ;ROUND FAILURE. CONSTANT BAD.
7917
7918  044174  000167  000224                  JMP     EEEDONE         ;GO TO THE NEXT TEST.
7919
7920                                  ;THIS SUBROUTINE, MULFSUB, IS CALLED TO SET UP, EXECUTE
7921                                  ;AND CHECK THE RESULT OF A MULF INSTRUCTION. IT IS CALLED THUS:
7922                                  ;
7923                                  ;               JSR     PC,@#MULFSUB
7924                                  ;       ACARG:  .WORD   X,X             ;AC OPERAND
7925                                  ;       FSRCARG:.WORD   X,X             ;FSRC OPERAND
7926                                  ;       RES:    .WORD   X,X             ;EXPECTED RESULT
7927                                  ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
7928                                  ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
7929                                  ;       ERRES:  .WORD   X,X             ;ERROR RESULT
7930                                  ;       ERR:    ERROR   X               ;RESULT ERROR
7931                                  ;       CONT:                           ;RETURN ADDRESS
7932                                  ;
7933                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
7934                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, MULF IS EXECUTED.
7935                                  ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
7936                                  ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
7937                                  ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
7938                                  ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
7939                                  ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
7940                                  ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
7941                                  ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
7942                                  ;THEN THE FAILURE IS REPORTED IN MULFSUB AND CONTROL IS PASSED TO
7943                                  ;CONT. IF NO ERRORS ARE DETECTED THEN MULFSUB RETURNS CONTROL
7944                                  ;TO CONT.
7945
7946  044200  012601                  MULFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
7947  044202  012700  000200                  MOV     #200,R0         ;SET FD MODE.
7948  044206  170100                          LDFPS   R0
7949  044210  010100                          MOV     R1,R0           ;LOAD THE AC OPERAND.
7950  044212  172410                          LDD     (R0),AC0
7951  044214  016100  000014                  MOV     14(R1),R0       ;LOAD THE FPS
7952  044220  170100                          LDFPS   R0
7953  044222  012737  044236  001236          MOV     #1$,@#$TMP2
7954  044230  010100                          MOV     R1,R0
7955  044232  062700  000004                  ADD     #4,R0           ;ESTABLISH A POINTER TO FSRC.
```

```
7956
7957  044236  171010                  1$:     MULF    (R0),AC0        ;TEST INSTRUCTION.
7958
7959  044240  170204                          STFPS   R4              ;GET THE FPS.
7960  044242  012700  000200                  MOV     #200,R0         ;SET FD MODE
7961  044246  170100                          LDFPS   R0
7962
7963  044250  012700  044414                  MOV     #MULFT,R0       ;GET THE RESULT OF THE MULF.
7964  044254  174010                          STD     AC0,(R0)
7965
7966  044256  010102                          MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
7967  044260  010237  001240                  MOV     R2,@#$TMP3
7968  044264  062702  000004                  ADD     #4,R2
7969  044270  010237  001242                  MOV     R2,@#$TMP4
7970  044274  062702  000004                  ADD     #4,R2
7971  044300  010237  001244                  MOV     R2,@#$TMP5
7972  044304  012737  044414  001246          MOV     #MULFT,@#$TMP6
7973  044312  010437  001250                  MOV     R4,@#$TMP7
7974  044316  016137  000016  001252          MOV     16(R1),@#$TMP10
7975
7976  044324  021061  000010                  CMP     (R0),10(R1)     ;IS THE RESULT CORRECT?
7977  044330  001011                          BNE     10$             ;IF INCORRECT BRANCH.
7978  044332  026061  000002  000012          CMP     2(R0),12(R1)
7979  044340  001005                          BNE     10$
7980
7981  044342  026104  000016                  CMP     16(R1),R4       ;IS FPS CORRECT?
7982  044346  001020                          BNE     15$             ;IF INCORRECT BRANCH.
7983  044350  000161  000026                  JMP     26(R1)          ;IF NO ERRORS OCCURRED RETURN.
7984
7985  044354  021061  000020          10$:    CMP     (R0),20(R1)     ;DOES THE INCORRECT RESULT
7986  044360  001010                          BNE     11$             ;MATCH THE ANTICIPATED INCORRECT RESULT.
7987  044362  026061  000002  000022          CMP     2(R0),22(R1)
7988  044370  001004                          BNE     11$             ;BRANCH IF NO.
7989
7990  044372  010102                          MOV     R1,R2           ;IT MATCHED SO RETURN TO THE ERROR
7991                                                                  ;REPORT AT THE CALL_NG ROUTINE.
7992  044374  062702  000024                  ADD     #24,R2
7993  044400  000112                          JMP     (R2)
7994
7995  044402                          11$:                            ;REPORT RESULT INCORRECT.
7996  044402  104002                  12$:    ERROR   2
7997  044404  000161  000026          13$:    JMP     26(R1)
7998
7999  044410                          15$:                            ;REPORT FPS INCORRECT.
8000  044410  104002                  16$:    ERROR   2
8001  044412  000774                          BR      13$
8002
8003  044414  000000  000000  000000  MULFT:  .WORD   0,0,0,0
8004  044422  000000
8005
8006  044424                          EEEDONE:
8007  044424  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
8008                                                                  ;SEE IF THE USER HAS EXPRESSED
8009                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
8010                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
8011                                                                  ;THE USER TYPED CONTROL G?).
```

```
8012
8013
8014
8015                                    ;;*****************************************************************
8016                                    ;*TEST 47        MULD TEST
8017                                    ;*
8018                                    ;*THIS IS A TEST OF THE MULD INSTRUCTION. NOTE THAT A SUBROUTINE IS
8019                                    ;*USED TO SET UP THE OPERANDS, EXECUTE THE MULD INSTRUCTION AND
8020                                    ;*CHECK THE RESULTS.
8021                                    ;*
8022                                    ;;*****************************************************************
8023  044426  000004                   TST47:  SCOPE
8024
8025                                    ;MULD TEST WITH AC POSITIVE AND FSRC POSITIVE.
8026  044430                            FFF1:
8027  044430  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8028  044432  004737  044714                   JSR     PC,@#MULDSUB
8029  044436  040200  000000  000000   1$:     .WORD   40200,0,0,0     ;AC
8030  044444  000000
8031  044446  023777  177777  177777   2$:     .WORD   23777,-1,-1,-1  ;FSRC
8032  044454  177777
8033  044456  023777  177777  177777   3$:     .WORD   23777,-1,-1,-1  ;RES
8034  044464  177777
8035  044466  000217                    4$:     217                     ;FPS BEFORE EXECUTION.
8036  044470  000200                            200                     ;FPS AFTER ECECUTION.
8037  044472  023777  177777  000000   5$:     .WORD   23777,-1,0,0    ;ERROR RES.
8038  044500  000000
8039  044502  104002                    6$:     ERROR   2               ;BAD CONSTANT USED IN ALGORITHM
8040                                            ;USED 24 INSTEAD OF 56.
8041
8042                                    ;MULD TEST WITH BOTH OPERANDS POSITIVE TRUNCATION TEST.
8043  044504                            FFF2:
8044  044504  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8045  044506  004767  000202                   JSR     PC,MULDSUB
8046  044512  065400  000000  000000   1$:     .WORD   65400,0,0,1     ;AC
8047  044520  000001
8048  044522  037577  177777  177777   2$:     .WORD   37577,-1,-1,-2  ;FSRC
8049  044530  177776
8050  044532  064777  177777  177777   3$:     .WORD   64777,-1,-1,-1  ;RES
8051  044540  177777
8052  044542  000247                    4$:     247                     ;FPS BEFORE EXECUTION.
8053  044544  000240                            240                     ;FPS AFTER EXECUTION.
8054  044546  065000  000000  000000   5$:     .WORD   65000,0,0,0     ;ERROR RES.
8055  044554  000000
8056  044556  104002                    6$:     ERROR   2               ;TRUNCATION ERROR.
8057
8058                                    ;MULD TEST WITH BOTH OPERANDS NEGATIVE IN ROUND MODE.
8059  044560                            FFF3:
8060  044560  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8061  044562  004737  044714                   JSR     PC,@#MULDSUB
8062  044566  137577  177777  177777   1$:     .WORD   137577,-1,-1,-2 ;AC
8063  044574  177776
8064  044576  165400  000000  000000   2$:     .WORD   165400,0,0,1    ;FSRC
8065  044604  000001
8066  044606  065000  000000  000000   3$:     .WORD   65000,0,0,0     ;RES
8067  044614  000000
```

```
8068  044616  007717                    4$:     7717                    ;FPS BEFORE EXECUTION.
8069  044620  007700                            7700                    ;FPS AFTER EXECUTION.
8070  044622  064777  177777  177777    5$:     .WORD   64777,-1,-1,-1  ;ERROR RES.
8071  044630  177777
8072  044632  104002                    6$:     ERROR   2               ;ROUND ERROR.
8073
8074                                    ;MULD   TEST WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
8075  044634                            FFF4:
8076  044634  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8077  044636  004737  044714                    JSR     PC,@#MULDSUB
8078  044642  017500  000000  000000    1$:     .WORD   17500,0,0,0     ;AC
8079  044650  000000
8080  044652  123652  125252            2$:     .WORD   123652,125252   ;FSRC
8081  044656  125252  125252                    .WORD   125252,125252
8082  044662  103177  177777  177777    3$:     .WORD   103177,-1,-1,-1 ;RES
8083  044670  177777
8084  044672  000200                    4$:     200                     ;FPS BEFORE EXECUTION.
8085  044674  000210                            210                     ;FPS AFTER EXECUTION.
8086  044676  103200  000000  000000    5$:     .WORD   103200,0,0,0    ;ERROR RES.
8087  044704  000000
8088  044706  104002                    6$:     ERROR   2               ;ROUND ERROR (BAD CONSTANT).
8089
8090  044710  000167  000240                    JMP     FFFDONE
8091
8092                                    ;THIS SUBROUTINE, MULDSUB, IS CALLED TO SET UP, EXECUTE
8093                                    ;AND CHECK THE RESULT OF A MULD INSTRUCTION. IT IS CALLED THUS:
8094                                    ;
8095                                    ;               JSR     PC,@#MULDSUB
8096                                    ;       ACARG:  .WORD   X,X,X,X         ;AC OPERAND
8097                                    ;       FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
8098                                    ;       RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
8099                                    ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
8100                                    ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
8101                                    ;       ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
8102                                    ;       ERR:    ERROR   X               ;RESULT ERROR
8103                                    ;       CONT:                           ;RETURN ADDRESS
8104                                    ;
8105                                    ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
8106                                    ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, MULD IS EXECUTED.
8107                                    ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
8108                                    ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
8109                                    ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
8110                                    ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
8111                                    ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
8112                                    ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
8113                                    ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
8114                                    ;THEN THE FAILURE IS REPORTED IN MULDSUB AND CONTROL IS PASSED TO
8115                                    ;CONT. IF NO ERRORS ARE DETECTED THEN MULDSUB RETURNS CONTROL
8116                                    ;TO CONT.
8117
8118  044714  012601                    MULDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
8119  044716  012700  000200                    MOV     #200,R0         ;SET FD MODE.
8120  044722  170100                            LDFPS   R0
8121
8122  044724  010100                            MOV     R1,R0           ;SET UP THE AC0 OPERAND.
8123  044726  172410                            LDD     (R0),AC0
```

```
8124  044730  016100  000030          MOV     30(R1),R0       ;LOAD THE FPS.
8125  044734  170100                  LDFPS   R0
8126
8127  044736  012737  044752  001236  MOV     #1$,@#$TMP2
8128  044744  010100                  MOV     R1,R0           ;ESTABLISH A POINTER TO FSRC.
8129  044746  062700  000010          ADD     #10,R0
8130
8131  044752  171010          1$:     MULD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
8132
8133  044754  170204                  STFPS   R4              ;GET THE FPS.
8134  044756  012700  000200          MOV     #200,R0         ;SET FD MODE.
8135  044762  170100                  LDFPS   R0
8136
8137  044764  012700  045144          MOV     #MULDT,R0       ;GET THE RESULT.
8138  044770  174010                  STD     AC0,(R0)
8139
8140  044772  010102                  MOV     R1,R2           ;SAVE DATA IN CASE OF ERROR.
8141  044774  010237  001240          MOV     R2,@#$TMP3
8142  045000  062702  000010          ADD     #10,R2
8143  045004  010237  001242          MOV     R2,@#$TMP4
8144  045010  062702  000010          ADD     #10,R2
8145  045014  010237  001244          MOV     R2,@#$TMP5
8146  045020  012737  045144  001246  MOV     #MULDT,@#$TMP6
8147  045026  010437  001250          MOV     R4,@#$TMP7
8148  045032  016137  000032  001252  MOV     32(R1),@#$TMP10
8149
8150  045040  010102                  MOV     R1,R2           ;CHECK THE RESULT.
8151  045042  062702  000020          ADD     #20,R2
8152  045046  012703  045144          MOV     #MULDT,R3
8153  045052  012705  000004          MOV     #4,R5
8154  045056  022223          2$:     CMP     (R2)+,(R3)+
8155  045060  001006                  BNE     10$             ;BRANCH IF RESULT INCORRECT.
8156  045062  077503                  SOB     R5,2$
8157
8158  045064  026104  000032          CMP     32(R1),R4       ;IS FPS CORRECT?
8159  045070  001023                  BNE     15$             ;BRANCH IF INCORRECT.
8160  045072  000161  000046          JMP     46(R1)          ;RETURN.
8161
8162  045076  010102          10$:    MOV     R1,R2           ;WAS INCORRECT RESULT ANTICIPATED?
8163  045100  062702  000034          ADD     #34,R2
8164  045104  012703  045144          MOV     #MULDT,R3
8165  045110  012705  000004          MOV     #4,R5
8166  045114  022223          11$:    CMP     (R2)+,(R3)+
8167  045116  001005                  BNE     12$             ;BRANCH IF NO.
8168  045120  077503                  SOB     R5,11$
8169  045122  010102                  MOV     R1,R2           ;IF THE INCORRECT RESULT WAS
8170  045124  062702  000044          ADD     #44,R2          ;ANTICIPATED RETURN TO THE
8171                                                          ;ERROR REPORT IN THE CALLING
8172  045130  000112                  JMP     (R2)            ;ROUTINE.
8173
8174  045132                  12$:                            ;REPORT RESULT INCORRECT.
8175  045132  104002          13$:    ERROR   2
8176  045134  000161  000046  *       14$:    JMP     46(R1)
8177
8178  045140                  15$:                            ;REPORT FPS INCORRECT.
8179  045140  104002          16$:    ERROR   2
```

```
8180  045142  000774                          BR      14$
8181
8182  045144  000000  000000  000000  MULDT:  .WORD   0,0,0,0
8183  045152  000000
8184
8185  045154                          FFFDONE:
8186  045154  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
8187                                                                  ;SEE IF THE USER HAS EXPRESSED
8188                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
8189                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
8190                                                                  ;THE USER TYPED CONTROL G?).
8191
8192
8193
8194                                    ;;*****************************************************************
8195                                    ;*TEST 50        UNDER\OVER FLOW, USING MULF WITH TRAPS DISABLED, TEST
8196                                    ;*
8197                                    ;*THIS IS A TEST OF THE OVERFLOW AND UNDERFLOW CONDITIONS USING
8198                                    ;*THE MULF INSTRUCTION WITH TRAPS DISABLED. NOTE THAT A SUBROUTINE
8199                                    ;*IS USED TO SET UP THE OPERANDS, EXECUTE THE MULF INSTRUCTION AND
8200                                    ;*CHECK THE RESULTS.
8201                                    ;*
8202                                    ;;*****************************************************************
8203  045156  000004                    TST50:  SCOPE
8204
8205                                    ;UNDERFLOW, WITH EXPONENT OF RESULT = -129
8206  045160                            1111:
8207  045160  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8208  045162  004737  045404                  JSR     PC,@#OVUNFNT
8209  045166  020200  000000          1$:     .WORD   20200,0         ;AC
8210  045172  020000  000000          2$:     .WORD   20000,0         ;FSRC
8211  045176  000000  000000          3$:     .WORD   0,0             ;RES
8212  045202  177777  177777          4$:     .WORD   -1,-1           ;ERROR RES.
8213  045206  000000                  5$:     0                       ;FPS BEFORE EXECUTION.
8214  045210  000004                          4                       ;FPS AFTER EXECUTION.
8215  045212  000012                  6$:     12                      ;FEC
8216  045214  177777                          -1                      ;FLAG
8217  045216  104002                  7$:     ERROR   2               ;ST 331 TO 155 INTO 115 (BUT FIU)
8218  045220  000401                          BR      8$
8219  045222  104002                          ERROR   2
8220  045224                          8$:
8221
8222                                    ;UNDERFLOW, WITH EXPONENT OF RESULT = -193
8223  045224                            1112:
8224  045224  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8225  045226  004737  045404                  JSR     PC,@#OVUNFNT
8226  045232  010200  000000          1$:     .WORD   10200,0         ;AC
8227  045236  010000  000000          2$:     .WORD   10000,0         ;FSRC
8228  045242  000000  000000          3$:     .WORD   0,0             ;RES
8229  045246  010000  000000          4$:     .WORD   10000,0         ;ERROR RES.
8230  045252  005013                  5$:     5013                    ;FPS BEFORE EXECUTION.
8231  045254  005004                          5004                    ;FPS AFTER EXECUTION.
8232  045256  000012                  6$:     12                      ;FEC
8233  045260  177777                          -1                      ;FLAG
8234  045262  104002                  7$:     ERROR   2               ;SETTING FIUV OR FIV CAUSES TRAP
8235                                                                  ;WITH FIU CLEAR.
```

```
8236  045264  000401                  T50     UNDER\OVER FLOW, USING MULF WITH TRAPS DISABLED, TEST
8237  045266  104002                          BR      8$
8238  045270                                  ERROR   2
8239                                  8$:
8240
8241  045270                          ;OVERFLOW, EXPONENT OF RESULT = 128
8242  045270  104414                  IIII3:
8243  045272  004737  045404                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8244  045276  060200  000000                  JSR     PC,@#OVUNFNT
8245  045302  060000  000000          1$:     .WORD   60200,0         ;AC
8246  045306  000000  000000          2$:     .WORD   60000,0         ;FSRC
8247  045312  060000  000000          3$:     .WORD   0,0             ;RES
8248  045316  000000                  4$:     .WORD   60000,0         ;ERROR RES.
8249  045320  000006                  5$:     0                       ;FPS BEFORE EXECUTION.
8250  045322  000010                          6                       ;FPS AFTER EXECUTION.
8251  045324  000000                  6$:     10                      ;FEC
8252  045326  104002                          0                       ;FLAG
8253  045330  000401                  7$:     ERROR   2               ;ST 333 TO 136 INTO 116 (BUT FIV).
8254  045332  104002                          BR      8$
8255  045334                                  ERROR   2
8256                                  8$:
8257
8258  045334                          ;OVERFLOW, EXPONENT OF RESULT = 130
8259  045334  104414                  IIII4:
8260  045336  004737  045404                  LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8261  045342  060200  000000                  JSR     PC,@#OVUNFNT
8262  045346  060200  000000          1$:     .WORD   60200,0         ;AC
8263  045352  000000  000000          2$:     .WORD   60200,0         ;FSRC
8264  045356  177777  177777          3$:     .WORD   0,0             ;RES
8265  045362  006011                  4$:     .WORD   -1,-1           ;ERROR RES.
8266  045364  006006                  5$:     6011                    ;FPS BEFORE EXECUTION.
8267  045366  000010                          6006                    ;FPS AFTER EXECUTION.
8268  045370  000000                  6$:     10                      ;FEC
8269  045372  104002                          0                       ;FLAG
8270                                  7$:     ERROR   2               ;SETTING FIUV OR FIU WITH
8271  045374  000401                                                  ;FIV CLEAR CAUSES TRAP.
8272  045376  104002                          BR      8$
8273  045400  000167  000410                  ERROR   2
8274                                  8$:     JMP     IIIDONE         ;GO TO NEXT TEST.
8275
8276
8277                                  ;THIS SUBROUTINE, OVUNFNT, IS USED TO SET UP THE OPERANDS, EXECUTE
8278                                  ;THE MULF INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
8279                                  ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
8280                                  ;TO IT IS MADE THUS:
8281                                  ;
8282                                  ;               ACARG:  .WORD   X,X             ;AC OPERAND
8283                                  ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
8284                                  ;               RES:    .WORD   X,X             ;EXPECTED RESULT
8285                                  ;               ERRES:  .WORD   X,X             ;ERROR RESULT
8286                                  ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
8287                                  ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
8288                                  ;               FEC:    .WORD   X               ;EXPECTED FEC
8289                                  ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
8290                                  ;               ERR1:   ERROR   X               ;TRAP ERROR.
8291                                  ;               BR      CONT
```

```
8292                                    ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
8293                                    ;               CONT:                           ;RETURN ADDRESS
8294                                    ;
8295                                    ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
8296                                    ;THE MULF INSTRUCTION IS EXECUTED. IF NO TRAP OCCURS THEN THE
8297                                    ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
8298                                    ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNFNT RETURNS CONTROL
8299                                    ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNFNT
8300                                    ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
8301                                    ;MULF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
8302                                    ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
8303                                    ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNFNT
8304                                    ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
8305                                    ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNFNT WILL
8306                                    ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
8307                                    ;IF A TRAP OCCURS (IT SHOULD NOT) THEN OVUNFNT WILL READ THE FEC.
8308                                    ;SHOULD THE FEC MATCH THE ANTICIPATED FEC OVUNFNT WILL
8309                                    ;STORE ALL DATA AND TRANSFER CONTROL TO THE ERROR CALL AT ERR1. IF THE
8310                                    ;FEC IS NOT THE SAME AS THE ANTICIPATED FEC OVUNFNT WILL REPORT
8311                                    ;THE ERROR AND RETURN TO CONT. NOTE THAT OVUNFNT USES THE FLAG
8312                                    ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
8313                                    ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).
8314
8315  045404  012601                    OVUNFNT:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
8316  045406  012700  000200                    MOV     #200,R0         ;SET FD MODE.
8317  045412  170100                            LDFPS   R0
8318
8319  045414  010100                            MOV     R1,R0           ;LOAD AC0, OPERAND.
8320  045416  172410                            LDD     (R0),AC0
8321
8322  045420  010102                            MOV     R1,R2           ;SAVE THE DATA PATTERNS IN CASE OF
8323  045422  010237  001240                    MOV     R2,@#$TMP3      ;ERROR.
8324  045426  062702  000004                    ADD     #4,R2
8325  045432  010237  001242                    MOV     R2,@#$TMP4
8326  045436  062702  000004                    ADD     #4,R2
8327  045442  010237  001244                    MOV     R2,@#$TMP5
8328  045446  016137  000022  001252            MOV     22(R1),@#$TMP10
8329  045454  012737  046004  001246            MOV     #OVFNTT,@#$TMP6
8330
8331  045462  016100  000020                    MOV     20(R1),R0       ;LOAD THE FPS.
8332  045466  170100                            LDFPS   R0
8333  045470  012737  045512  001236            MOV     #1$,@#$TMP2
8334  045476  012737  045676  000244            MOV     #25$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
8335                                                                    ;OF ERROR.
8336  045504  010100                            MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
8337  045506  062700  000004                    ADD     #4,R0
8338
8339  045512  171010                    1$:     MULF    (R0),AC0        ;TEST INSTRUCTION.
8340
8341  045514  170204                    2$:     STFPS   R4              ;GET FPS.
8342  045516  170305                            STST    R5              ;GET FEC.
8343  045520  012700  000200                    MOV     #200,R0         ;SET FD MODE.
8344  045524  170100                            LDFPS   R0
8345  045526  012700  046004                    MOV     #OVFNTT,R0      ;GET THE RESULT.
8346  045532  174010                            STD     AC0,(R0)
8347  045534  010437  001250                    MOV     R4,@#$TMP7
```

```
 8348  045540  010537  001254                MOV     R5,@#$TMP11
 8349
 8350  045544  012700  046004                MOV     #OVFNTT,R0      ;CHECK THE RESULT.
 8351  045550  010102                        MOV     R1,R2
 8352  045552  062702  000010                ADD     #10,R2
 8353  045556  012703  000002                MOV     #2,R3
 8354  045562  022022              3$.       CMP     (R0)+,(R2)+
 8355  045564  001015                        BNE     15$             ;BRANCH IF INCORRECT.
 8356  045566  077303                        SOB     R3,3$
 8357
 8358  045570  026104  000022                CMP     22(R1),R4       ;WAS FPS CORRECT?
 8359  045574  001002                        BNE     10$             ;BRANCH IF FPS IS INCORRECT.
 8360
 8361  045576  000161  000036        4$:     JMP     36(R1)          ;RETURN, TEST COMPLETED.
 8362
 8363                                ;REPORT INCORRECT FPS.
 8364  045602  005761  000026        10$:    TST     26(R1)          ;WAS THE RESULT OVER OR UNDER FLOW?
 8365  045606  001002                        BNE     12$             ;BRANCH IF UNDERFLOW.
 8366
 8367                                                                ;REPORT FPS BAD AFTER OVERFLOW.
 8368  045610  104002                11$:    ERROR   2
 8369  045612  000771                        BR      4$
 8370
 8371  045614                        12$:                            ;REPORT FPS BAD AFTER UNDERFLOW.
 8372  045614  104002                13$:    ERROR   2
 8373  045616  000767                        BR      4$
 8374
 8375                                ;RESULT INCORRECT.
 8376  045620  012700  046004        15$:    MOV     #OVFNTT,R0      ;SEE IF FAILURE IS ANTICIPATED
 8377  045624  010102                        MOV     R1,R2           ;FAILURE.
 8378  045626  062702  000014                ADD     #14,R2
 8379  045632  012703  000002                MOV     #2,R3
 8380  045636  022022              16$:    CMP     (R0)+,(R2)+
 8381  045640  001007                        BNE     17$             ;BRANCH IF NOT ANTICIPATED.
 8382  045642  077303                        SOB     R3,16$
 8383
 8384  045644  010102                        MOV     R1,R2           ;ERROR WAS ANTICIPATED SO RETURN
 8385  045646  062702  000034                ADD     #34,R2          ;TO THE ERROR REPORT IN THE CALLING
 8386  045652  010237  001236                MOV     R2,@#$TMP2      ;ROUTINE.
 8387  045656  000112                        JMP     (R2)
 8388
 8389  045660  005761  000026        17$:    TST     26(R1)          ;RESULT WAS NOT ANTICIPATED
 8390                                                                ;SO ERROR MUST BE REPORTED HERE.
 8391                                                                ;FIRST SEE IF ARGUMENTS SHOULD
 8392                                                                ;HAVE RESULTED IN OVERFLOW OR UNDER
 8393                                                                ;FLOW BY LOOKING AT THE FLAG.
 8394  045664  001002                        BNE     19$             ;BRANCH IF UNDERFLOW EXPECTED.
 8395
 8396                                                                ;REPORT RESULT INCORRECT, EXPECTING
 8397  045666  104002                18$:    ERROR   2               ;OVERFLOW.
 8398  045670  000742                        BR      4$
 8399
 8400  045672                        19$:                            ;REPORT RESULT INCORRECT, EXPECTING
 8401  045672  104002                20$:    ERROR   2               ;UNDERFLOW.
 8402  045674  000740                        BR      4$
 8403
```

```
8404
8405  045676  011602                 ;IF AN FP TRAP OCCURS COME HERE.
8406  045700  022702  045514  25$:    MOV     (SP),R2         ;GET ADDRESS OF TRAP.
8407  045704  001402                  CMP     #2$,R2          ;WAS THE TRAP DURING THE MULF INSTRUCTION?
8408  045706  000137  062524          BEQ     26$             ;BRANCH IF YES.
8409                                  JMP     @#FPSPUR        ;OTHERWISE GO REPORT A SPURIOUS
8410  045712  022626                                          ;FP TRAP.
8411  045714  010237  001236  26$:    CMP     (SP)+,(SP)+     ;RESET THE STACK.
8412  045720  170204                  MOV     R2,@#$TMP2      ;SAVE DATA FOR ERROR REPORT.
8413  045722  170305                  STFPS   R4              ;GET FPS.
8414  045724  012700  000200          STST    R5              ;GET FEC.
8415  045730  170100                  MOV     #200,R0         ;SET FD MODE.
8416  045732  012700  046004          LDFPS   R0
8417  045736  174010                  MOV     #OVFNTT,R0      ;GET THE RESULT.
8418  045740  010537  001254          STD     AC0,(R0)
8419  045744  020561  000024          MOV     R5,@#$TMP11
8420  045750  001004                  CMP     R5,24(R1)       ;WAS THE FEC ANTICIPATED?
8421                                  BNE     27$             ;BRANCH IF NOT ANTICIPATED.
8422  045752  010102
8423  045754  062702  000030          MOV     R1,R2           ;ERROR WAS ANTICIPATED SO
8424                                  ADD     #30,R2          ,RETURN TO THE ERROR REPORT OF THE
8425  045760  000112                                          ;CALLING ROUTINE.
8426                                  JMP     (R2)
8427  045762  005761  000026  27$:    TST     26(R1)          ;THE ERROR WAS NOT ANTICIPATED SO
8428                                                          ;IT MUST BE REPORTED HERE. FIRST SEE IF EXPECTED
8429                                                          ;OVERFLOW OR UNDER FLOW.
8430  045766  001003                  BNE     29$             ;BRANCH IF EXPECTING UNDERFLOW
8431
8432                                                          ;REPORT TRAPPED ON OVERFLOW WITH FIV=0
8433  045770  104002          28$:    ERROR   2
8434  045772  000161  000036          JMP     36(R1)
8435
8436  045776                  29$:
8437  045776  104002          30$:    ERROR   2               ;REPORT TRAPPED ON UNDER FLOW WITH FIU=0
8438  046000  000161  000036          JMP     36(R1)
8439
```

```
8440  046004  000000  000000  000000  OVFNTT: .WORD   0,0,0,0
8441  046012  000000
8442
8443  046014                          IIIDONE:
8444  046014  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
8445                                                                  ;SEE IF THE USER HAS EXPRESSED
8446                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
8447                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
8448                                                                  ;THE USER TYPED CONTROL G?).
8449
8450
8451
8452
8453                                    ;;*****************************************************************
8454                                    ;*TEST 51        UNDER\OVER FLOW, USING MULD WITH TRAP DISABLED, TEST
8455                                    ;*
8456                                    ;*THIS IS A TEST OF THE OVERFLOW AND UNDERFLOW CONDITIONS THAT CAN
8457                                    ;*ARRISE USING THE MULD INSTRUCTION WITH TRAPS DISABLED. A SUBROUTINE IS
8458                                    ;*USED TO SET UP THE OPERANDS, EXECUTE THE MULD INSTRUCTION AND
8459                                    ;*CHECK THE RESULTS.
8460                                    ;*
8461                                    ;;********* *****************************************************
8462  046016  000004 .                  TST51:  SCOPE
8463
8464                                    ;UNDERFLOW, EXPONENT OF RESULT=-129
8465  046020                            JJJ1:
8466  046020  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8467  046022  004737  046344                    JSR     PC,@#OVUNDNT
8468  046026  020200  000000           1$:     .WORD   20200,0         ;AC
8469  046032  127272  000000                    .WORD   127272,0
8470  046036  020000  000000  000000  2$:     .WORD   20000,0,0,0     ;FSRC
8471  046044  000000
8472  046046  000000  000000  000000  3$:     .WORD   0,0,0,0         ;RES
8473  046054  000000
8474  046056  000000  000000           4$:     .WORD   0,0             ;ERROR RES.
8475  046062  127272  000000                    .WORD   127272,0
8476  046066  000200                    5$:     200                     ;FPS BEFORE EXECUTION.
8477  046070  000204                            204                     ;FPS AFTER EXECUTION.
8478  046072  000012                    6$:     12                      ;FEC
8479  046074  177777                            -1                      ;FLAG
8480  046076  104002                    7$:     ERROR   2               ;ST 331 TO 155 INTO 115 (BUT FIU)
8481  046100  000401                            BR      8$
8482  046102  104002                            ERROR   2               ;ST 115 (BUT FD)
8483  046104                            8$:
8484
8485                                    ;UNDERFLOW, EXPONENT OF RESULT = -193
8486  046104                            JJJ2:
8487  046104  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8488  046106  004737  046344                    JSR     PC,@#OVUNDNT
8489  046112  010200  000000           1$:     .WORD   10200,0         ;AC
8490  046116  123456  000000                    .WORD   123456,0
8491  046122  010000  000000  000000  2$:     .WORD   10000,0,0,0     ;FSRC
8492  046130  000000
8493  046132  000000  000000  000000  3$:     .WORD   0,0,0,0         ;RES
8494  046140  000000
8495  046142  000000  000000  123456  4$:     .WORD   0,0,123456,0    ;ERROR RES
```

```
8496  046150  000000
8497  046152  005213                  5$:     5213                    ;FPS BEFORE EXECUTION.
8498  046154  005204                          5204                    ;FPS AFTER EXECUTION.
8499  046156  000012                  6$:     12                      ;FEC
8500  046160  177777                          -1                      ;FLAG
8501  046162  104002                  7$:     ERROR   2               ;SETTING FIUV OR FIV BAD.
8502  046164  000401                          BR      8$
8503  046166  104002                          ERROR   2               ;ST 115 (BUT FD)
8504  046170                          8$:
8505
8506                                  ;OVERFLOW, EXPONENT OF RESULT = 128
8507  046170                          JJJ3:
8508  046170  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8509  046172  004737  046344                JSR     PC,@#OVUNDNT
8510  046176  060200  000000          1$:     .WORD   60200,0         ;AC
8511  046202  065432  000000                  .WORD   65432,0
8512  046206  060000  000000  000000  2$:     .WORD   60000,0,0,0     ;FSRC
8513  046214  000000
8514  046216  000000  000000  000000  3$:     .WORD   0,0,0,0         ;RES
8515  046224  000000
8516  046226  000000  000000  065432  4$:     .WORD   0,0,65432,0     ;ERROR RES.
8517  046234  000000
8518  046236  000200                  5$:     200                     ;FPS BEFORE EXECUTION.
8519  046240  000206                          206                     ;FPS AFTER EXECUTION.
8520  046242  000010                  6$:     10                      ;FEC
8521  046244  000000                          0                       ;FLAG
8522  046246  104002                  7$:     ERROR   2               ;ST 333 TO 136 INTO 116 (BUT FIV)
8523  046250  000401                          BR      8$
8524  046252  104002                          ERROR   2               ;ST 116 (BUT FD)
8525  046254                          8$:
8526
8527
8528                                  ;OVERFLOW, EXPONENT OF RESULT = 130
8529  046254                          JJJ4:
8530  046254  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8531  046256  004737  046344                JSR     PC,@#OVUNDNT
8532  046262  060200  000000          1$:     .WORD   60200,0         ;AC
8533  046266  125252  000000                  .WORD   125252,0
8534  046272  060200  000000  000000  2$:     .WORD   60200,0,0,0     ;FSRC
8535  046300  000000
8536  046302  000000  000000  000000  3$:     .WORD   0,0,0,0         ;RES
8537  046310  000000
8538  046312  000000  000000  125252  4$:     .WORD   0,0,125252,0    ;ERROR RES.
8539  046320  000000
8540  046322  006211                  5$:     6211                    ;FPS BEFORE EXECUTION.
8541  046324  006206                          6206                    ;FPS AFTER EXECUTION.
8542  046326  000010                  6$:     10                      ;FEC
8543  046330  000000                          0                       ;FLAG
8544  046332  104002                  7$:     ERROR   2               ;SETTING FIUV OR FIV BAD.
8545  046334  000401                          BR      8$
8546  046336  104002                          ERROR   2               ;ST 116 (BUT FD)
8547  046340  000137  046754          8$:     JMP     @#JJJDONE       ;GO TO NEXT TEST.
8548
8549                                  ;THIS SUBROUTINE, OVUNDNT, IS USED TO SET UP THE OPERANDS, EXECUTE
8550                                  ;THE MULD INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
8551                                  ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
```

```
8552                                    ;TO IT IS MADE THUS:
8553                                    ;
8554                                    ;           ACARG:  .WORD   X,X,X,X         ;AC OPERAND
8555                                    ;           FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
8556                                    ;           RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
8557                                    ;           ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
8558                                    ;           FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
8559                                    ;           FPSA:   .WORD   X               ;FPS AFTER EXECUTION
8560                                    ;           FEC:    .WORD   X               ;EXPECTED FEC
8561                                    ;           FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
8562                                    ;           ERR1:   ERROR   X               ;TRAP ERROR.
8563                                    ;           BR      CONT
8564                                    ;           ERR2:   ERROR   X               ;DATA, RESULT ERROR
8565                                    ;           CONT:                           ;RETURN ADDRESS
8566                                    ;
8567                                    ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
8568                                    ;THE MULD INSTRUCTION IS EXECUTED. IF NO TRAP OCCURS THEN THE
8569                                    ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
8570                                    ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNDNT RETURNS CONTROL
8571                                    ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNDNT
8572                                    ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
8573                                    ;MULD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
8574                                    ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
8575                                    ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNDNT
8576                                    ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
8577                                    ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNDNT WILL
8578                                    ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
8579                                    ;IF A TRAP OCCURS (IT SHOULD NOT) THEN OVUNDNT WILL READ THE FEC.
8580                                    ;SHOULD THE FEC MATCH THE ANTICIPATED FEC OVUNDNT WILL
8581                                    ;STORE ALL DATA AND TRANSFER CONTROL TO THE ERROR CALL AT ERR1. IF THE
8582                                    ;FEC IS NOT THE SAME AS THE ANTICIPATED FEC OVUNDNT WILL REPORT
8583                                    ;THE ERROR AND RETURN TO CONT. NOTE THAT OVUNDNT USES THE FLAG
8584                                    ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
8585                                    ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).
8586
8587  046344  012601                    OVUNDNT:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
8588  046346  012700  000200                    MOV     #200,R0         ;SET FD MODE.
8589  046352  170100                     LDFPS   R0
8590
8591  046354  010100                     MOV     R1,R0           ;LOAD AC0, OPERAND.
8592  046356  172410                     LDD     (R0),AC0
8593
8594  046360  010102                     MOV     R1,R2           ;SAVE THE DATA PATTERNS IN CASE OF
8595  046362  010237  001240             MOV     R2,@#$TMP3      ;ERROR.
8596  046366  062702  000010             ADD     #10,R2
8597  046372  010237  001242             MOV     R2,@#$TMP4
8598  046376  062702  000010             ADD     #10,R2
8599  046402  010237  001244             MOV     R2,@#$TMP5
8600  046406  016137  000042  001252     MOV     42(R1),@#$TMP10
8601  046414  012737  046744  001246     MOV     #OVDNTT,@#$TMP6
8602
8603  046422  016100  000040             MOV     40(R1),R0       ;LOAD THE FPS.
8604  046426  170100                     LDFPS   R0
8605  046430  012737  046452  001236     MOV     #1$,@#$TMP2
8606  046436  012737  046636  000244     MOV     #25$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
8607                                                              ;OF ERROR.
```

```
8608  046444  010100                        MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
8609  046446  062700  000010                ADD     #10,R0
8610
8611  046452  171010                1$:     MULD    (R0),AC0        ;TEST INSTRUCTION.
8612
8613  046454  170204                2$:     STFPS   R4              ;GET FPS.
8614  046456  170305                        STST    R5              ;GET FEC.
8615  046460  012700  000200                MOV     #200,R0         ;SET FD MODE.
8616  046464  170100                        LDFPS   R0
8617  046466  012700  046744                MOV     #OVDNTT,R0      ;GET THE RESULT.
8618  046472  174010                        STD     AC0,(R0)
8619  046474  010437  001250                MOV     R4,@#$TMP7
8620  046500  010537  001254                MOV     R5,@#$TMP11
8621
8622  046504  012700  046744                MOV     #OVDNTT,R0      ;CHECK THE RESULT.
8623  046510  010102                        MOV     R1,R2
8624  046512  062702  000020                ADD     #20,R2
8625  046516  012703  000004                MOV     #4,R3
8626  046522  022022                3$:     CMP     (R0)+,(R2)+
8627  046524  001015                        BNE     15$             ;BRANCH IF INCORRECT.
8628  046526  077303                        SOB     R3,3$
8629
8630  046530  026104  000042                CMP     42(R1),R4       ;WAS FPS CORRECT?
8631  046534  001002                        BNE     10$             ;BRANCH IF FPS IS INCORRECT.
8632
8633  046536  000161  000056        4$:     JMP     56(R1)          ;RETURN, TEST COMPLETED.
8634
8635                                ;REPORT INCORRECT FPS.
8636  046542  005761  000046        10$:    TST     46(R1)          ;WAS THE RESULT OVER OR UNDER FLOW?
8637  046546  001002                        BNE     12$             ;BRANCH IF UNDERFLOW.
8638
8639                                                                ;REPORT FPS BAD AFTER OVERFLOW.
8640  046550  104002                11$:    ERROR   2
8641  046552  000771                        BR      4$
8642
8643  046554                        12$:                            ;REPORT FPS BAD AFTER UNDERFLOW.
8644  046554  104002                13$:    ERROR   2
8645  046556  000767                        BR      4$
8646
8647                                ;RESULT INCORRECT.
8648  046560  012700  046744        15$:    MOV     #OVDNTT,R0      ;SEE IF FAILURE IS ANTICIPATED
8649  046564  010102                        MOV     R1,R2           ;FAILURE.
8650  046566  062702  000030                ADD     #30,R2
8651  046572  012703  000004                MOV     #4,R3
8652  046576  022022                16$:    CMP     (R0)+,(R2)+
8653  046600  001007                        BNE     17$             ;BRANCH IF NOT ANTICIPATED.
8654  046602  077303                        SOB     R3,16$
8655
8656  046604  010102                        MOV     R1,R2           ;ERROR WAS ANTICIPATED SO RETURN
8657  046606  062702  000054                ADD     #54,R2          ;TO THE ERROR REPORT IN THE CALLING
8658  046612  010237  001236                MOV     R2,@#$TMP2      ;ROUTINE.
8659  046616  000112                        JMP     (R2)
8660
8661  046620  005761  000046        17$:    TST     46(R1)          ;RESULT WAS NOT ANTICIPATED
8662                                                                ;SO ERROR MUST BE REPORTED HERE.
8663                                                                ;FIRST SEE IF ARGUMENTS SHOULD
```

```
8664
8665                                                    ;HAVE RESULTED IN OVERFLOW OR UNDER
8666  046624  001002                    BNE     19$     ;FLOW BY LOOKING AT THE FLAG.
8667                                                    ;BRANCH IF UNDERFLOW EXPECTED.
8668
8669  046626  104002            18$:    ERROR   2       ;REPORT RESULT INCORRECT, EXPECTING
8670  046630  000742                    BR      4$      ;OVERFLOW.
8671
8672  046632                    19$:
8673  046632  104002            20$:    ERROR   2       ;REPORT RESULT INCORRECT, EXPECTING
8674  046634  000740                    BR      4$      ;UNDERFLOW.
8675
8676                            ;IF AN FP TRAP OCCURS COME HERE.
8677  046636  011602            25$:    MOV     (SP),R2 ;GET ADDRESS OF TRAP.
8678  046640  022702  046454            CMP     #2$,R2  ;WAS THE TRAP DURING THE MULF INSTRUCTION?
8679  046644  001402                    BEQ     26$     ;BRANCH IF YES.
8680  046646  000137  062524            JMP     @#FPSPUR ;OTHERWISE GO REPORT A SPURIOUS
8681                                                    ;FP TRAP.
8682  046652  022626            26$:    CMP     (SP)+,(SP)+ ;RESET THE STACK.
8683  046654  010237  001236            MOV     R2,@#$TMP2 ;SAVE DATA FOR ERROR REPORT.
8684  046660  170204                    STFPS   R4      ;GET FPS.
8685  046662  170305                    STST    R5      ;GET FEC.
8686  046664  012700  000200            MOV     #200,R0 ;SET FD MODE.
8687  046670  170100                    LDFPS   R0
8688  046672  012700  046744            MOV     #OVDNTT,R0 ;GET THE RESULT.
8689  046676  174010                    STD     AC0,(R0)
8690  046700  010537  001254            MOV     R5,@#$TMP11
8691  046704  020561  000044            CMP     R5,44(R1) ;WAS THE FEC ANTICIPATED?
8692  046710  001004                    BNE     27$     ;BRANCH IF NOT ANTICIPATED.
8693
8694  046712  010102                    MOV     R1,R2   ;ERROR WAS ANTICIPATED SO
8695  046714  062702  000050            ADD     #50,R2  ;RETURN TO THE ERROR REPORT OF THE
8696                                                    ;CALLING ROUTINE.
8697  046720  000112                    JMP     (R2)
8698
8699  046722  005761  000026    27$:    TST     26(R1)  ;THE ERROR WAS NOT ANTICIPATED SO
8700                                                    ;IT MUST BE REPORTED HERE. FIRST SEE IF EXPECTED
8701                                                    ;OVERFLOW OR UNDER FLOW.
8702  046726  001003                    BNE     29$     ;BRANCH IF EXPECTING UNDERFLOW
8703
8704                                                    ;REPORT TRAPPED ON OVERFLOW WITH FIV=0
8705  046730  104002            28$:    ERROR   2
8706  046732  000161  000056            JMP     56(R1)
8707
8708  046736                    29$:                  ;REPORT TRAPPED ON UNDER FLOW WITH FIU-0
8709  046736  104002            30$:    ERROR   2
8710  046740  000161  000056            JMP     56(R1)
8711
8712  046744  000000  000000  000000 OVDNTT: .WORD   0,0,0,0
8713  046752  000000
8714
8715  046754                    JJJDONE:
8716  046754  104413                    RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
8717                                                    ;SEE IF THE USER HAS EXPRESSED
8718                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
8719                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
```

```
8720                                                    ;THE USER TYPED CONTROL G?).
8721
8722
8723
8724
8725                                    ;;*****************************************************************
8726                                    ;*TEST 52        UNDER\OVER FLOW, USING MULF WITH TRAPS ENABLED, TEST
8727                                    ;*
8728                                    ;*THIS IS A TEST OF THE UNDERFLOW AND OVERFLOW
8729                                    ;*CONDITIONS THAT CAN OCCUR USING THE MULF INSTRUCTION.
8730                                    ;*A SUBROUTINE IS CALLED TO SET UP THE OPERANDS,
8731                                    ;* EXECUTE THE MULF INSTRUCTION AND CHECK
8732                                    ;*THE RESULTS. HERE THE PARTICULAR INTERRUPT,
8733                                    ;*EITHER OVERFLOW OR UNDERFLOW, IS ENABLED SO A TRAP SHOULD
8734                                    ;*OCCUR.
8735                                    ;*
8736                                    ;;*****************************************************************
8737  046756  000004                    TST52:  SCOPE
8738
8739                                    ;UNDERFLOW, EXPONENT OF RESULT = -129
8740  046760                            KKK1:
8741  046760  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8742  046762  004737  047204                    JSR     PC,@#OVUNFT
8743  046766  020123  045676           1$:     .WORD   20123,45676     ;AC
8744  046772  020200  000000           2$:     .WORD   20200,0         ;FSRC
8745  046776  000123  045676           3$:     .WORD   123,45676       ;RES
8746  047002  177777  177777           4$:     .WORD   -1,-1           ;ERROR RES.
8747  047006  002000                    5$:     2000                    ;FPS BEFORE EXECUTION.
8748  047010  102004                            102004                  ;FPS AFTER EXECUTION.
8749  047012  000012                    6$:     12                      ;FEC
8750  047014  177777                            -1                      ;FLAG
8751  047016  104002                    7$:     ERROR   2               ;ST 331 (BUT FIU) NO TRAP.
8752
8753  047020  000401                            BR      8$
8754  047022  104002                            ERROR   2
8755  047024                            8$:
8756
8757                                    ;UNDERFLOW, EXPONENT OF THE RESULT = -193
8758  047024                            KKK3:
8759  047024  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
8760  047026  004737  047204                    JSR     PC,@#OVUNFT
8761  047032  010127  127272           1$:     .WORD   10127,127272    ;AC
8762  047036  010200  000000           2$:     .WORD   10200,0         ;FSRC
8763  047042  060127  127272           3$:     .WORD   60127,127272    ;RES
8764  047046  177777  177777           4$:     .WORD   -1,-1           ;ERROR RES.
8765  047052  007017                    5$:     7017                    ;FPS BEFORE EXECUTION.
8766  047054  107000                            107000                  ;FPS AFTER EXECUTION.
8767  047056  000012                    6$:     12                      ;FEC
8768  047060  177777                            -1
8769  047062  104002                    7$:     ERROR   2               ;ST 331 (BUT FIU) NO TRAP.
8770  047064  000401                            BR      8$
8771  047066  104002                            ERROR   2
8772  047070                            8$:
8773
8774                                    ;OVERFLOW, EXPONENT OF THE RESULT = 128
8775  047070                            KKK4:
```

```
 8776  047070  104414                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 8777  047072  004737  047204                JSR     PC,@#OVUNFT
 8778  047076  060252  125252          1$:   .WORD   60252,125252            ;AC
 8779  047102  060000  000000          2$:   .WORD   60000,0                 ;FSRC
 8780  047106  000052  125252          3$:   .WORD   000052,125252           ;RES
 8781  047112  177777  177777          4$:   .WORD   -1,-1                   ;ERROR RES.
 8782  047116  001000                  5$:   1000                            ;FPS BEFORE EXECUTION.
 8783  047120  101006                        101006                          ;FPS AFTER EXECUTION.
 8784  047122  000010                  6$:   10                              ;FEC
 8785  047124  000000                        0                               ;FLAG
 8786  047126  104002                  7$:   ERROR   2                       ;ST 333 (BUT FIV) NO TRAP
 8787  047130  000401                        BR      8$
 8788  047132  104002                        ERROR   2
 3789  047134                          8$:
 8790
 8791                                  ;OVERFLOW, EXPONENT OF RESULT = 130
 8792  047134                          KKK5:
 8793  047134  104414                        LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
 8794  047136  004737  047204                JSR     PC,@#OVUNFT
 8795  047142  060345  067654          1$:   .WORD   60345,67654             ;AC
 8796  047146  060200  000000          2$:   .WORD   60200,0                 ;FSRC
 8797  047152  000345  067654          3$:   .WORD   345,67654               ;RES
 8798  047156  177777  177777          4$:   .WORD   -1,-1                   ;ERROR RES.
 8799  047162  007015                  5$:   7015                            ;FPS BEFORE EXECUTION.
 8800  047164  107002                        107002                          ;FPS AFTER EXECUTION.
 8801  047166  000010                  6$:   10                              ;FEC
 8802  047170  000000                        0                               ;FLAG
 8803  047172  104002                  7$:   ERROR   2                       ;ST 133 (BUT FIV) NO TRAP
 8804  047174  000401                        BR      8$
 8805  047176  104002                        ERROR   2
 8806  047200  000167  000412        8$:   JMP     KKKDONE
 8807
 8808                                  ;THIS SUBROUTINE, OVUNFT, IS USED TO SET UP THE OPERANDS, EXECUTE
 8809                                  ;THE MULF INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
 8810                                  ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
 8811                                  ;TO IT IS MADE THUS:
 8812                                  ;
 8813                                  ;               ACARG:  .WORD   X,X             ;AC OPERAND
 8814                                  ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
 8815                                  ;               RES:    .WORD   X,X             ;EXPECTED RESULT
 8816                                  ;               ERRES:  .WORD   X,X             ;ERROR RESULT
 8817                                  ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
 8818                                  ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
 8819                                  ;               FEC:    .WORD   X               ;EXPECTED FEC
 8820                                  ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
 8821                                  ;               ERR1:   ERROR   X               ;TRAP ERROR.
 8822                                  ;               BR      CONT
 8823                                  ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
 8824                                  ;               CONT:                           ;RETURN ADDRESS
 8825                                  ;
 8826                                  ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
 8827                                  ;THE MULF INSTRUCTION IS EXECUTED. IF THE TRAP OCCURS THEN THE
 8828                                  ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
 8829                                  ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNFT RETURNS CONTROL
 8830                                  ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNFT
 8831                                  ;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).

047204  012601                  OVUNFT: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
047206  012700  000200                  MOV     #200,R0         ;SET FD MODE.
047212  170100                          LDFPS   R0

047214  010100                          MOV     R1,R0           ;LOAD AC0, OPERAND.
047216  172410                          LDD     (R0),AC0

047220  010102                          MOV     R1,R2           ;SAVE THE DATA PATTERNS IN CASE OF
047222  010237  001240                  MOV     R2,@#$TMP3      ;ERROR.
047226  062702  000004                  ADD     #4,R2
047232  010237  001242                  MOV     R2,@#$TMP4
047236  062702  000004                  ADD     #4,R2
047242  010237  001244                  MOV     R2,@#$TMP5
047246  016137  000022  001252          MOV     22(R1),@#$TMP10
047254  012737  047606  001246          MOV     #OVFTT,@#$TMP6

047262  016100  000020                  MOV     20(R1),R0       ;LOAD THE FPS.
047266  170100                          LDFPS   R0
047270  012737  047312  001236          MOV     #1$,@#$TMP2

047276  012737  047322  000244          MOV     #50$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
                                                                ;OF ERROR.
047304  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
047306  062700  000004                  ADD     #4,R0

047312  171010                  1$:     MULF    (R0),AC0        ;TEST INSTRUCTION. SHOULD CAUSE TRAP.
047314  170000                  2$:     CFCC

047316  000137  047546                  JMP     @#25$           ;FAILURE, NO TRAP.

047322  011602                  50$:    MOV     (SP),R2         ;TRAP TO HERE AND SEE IF THE PC OF THE
047324  020227  047314                  CMP     R2,#2$          ;TRAP WAS THAT OF THE MULF INSTRUCTION.
047330  001402                          BEQ     51$             ;BRANCH IF YES.
047332  000137  062524                  JMP     @#FPSPUR        ;OTHERWISE REPORT SPURIOUS FP ERROR.

047336  022626                  51$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
047340  170204                          STFPS   R4              ;GET FPS.
047342  170305                          STST    R5              ;GET FEC.
047344  012700  000200                  MOV     #200,R0         ;SET FD MODE.
047350  170100                          LDFPS   R0
047352  012700  047606                  MOV     #OVFTT,R0       ;GET THE RESULT.
047356  174010                          STD     AC0,(R0)
047360  010437  001250                  MOV     R4,@#$TMP7
047364  010537  001254                  MOV     R5,@#$TMP11
```

```
8888
8889  047370  012700  047606          MOV     #OVFTT,R0       ;CHECK THE RESULT.
8890  047374  010102                  MOV     R1,R2
8891  047376  062702  000010          ADD     #10,R2
8892  047402  012703  000002          MOV     #2,R3
8893  047406  022022          3$:     CMP     (R0)+,(R2)+
8894  047410  001027                  BNE     15$             ;BRANCH IF INCORRECT.
8895  047412  077303                  SOB     R3,3$
8896
8897  047414  026104  000022          CMP     22(R1),R4       ;WAS FPS CORRECT?
8898  047420  001014                  BNE     10$             ;BRANCH IF FPS IS INCORRECT.
8899
8900  047422  026105  000024          CMP     24(R1),R5       ;IS FEC CORRECT?
8901  047426  001002                  BNE     5$              ;IF INCORRECT BRANCH.
8902  047430  000161  000036  4$:     JMP     36(R1)          ;RETURN, TEST COMPLETED
8903
8904                          ;REPORT INCORRECT FEC.
8905  047434  005761  000026  5$:     TST     26(R1)          ;WAS THE RESULT OVERFLOW OR UNDERFLOW?
8906  047440  001002                  BNE     7$              ;BRANCH IF UNDERFLOW.
8907
8908                                                          ;RFPORT BAD FEC ON EXPECTED OVERFLOW.
8909  047442  104002          6$:     ERROR   2
8910  047444  000771                  BR      4$
8911
8912  047446          7$:                                     ;REPORT BAD FEC ON EXPECTED UNDERFLOW.
8913  047446  104002          8$:     ERROR   2
8914  047450  000767                  BR      4$
8915
8916                          ;REPORT INCORRECT FPS.
8917  047452  005761  000026  10$:    TST     26(R1)          ;WAS THE RESULT OVER OR UNDER FLOW?
8918  047456  001002                  BNE     12$             ;BRANCH IF UNDERFLOW.
8919
8920                                                          ;REPORT FPS BAD AFTER OVERFLOW.
8921  047460  104002          11$:    ERROR   2
8922  047462  000762                  BR      4$
8923
8924  047464          12$:                                    ;REPORT FPS BAD AFTER UNDERFLOW.
8925  047464  104002          13$:    ERROR   2
8926  047466  000760                  BR      4$
8927
8928                          ;RESULT INCORRECT.
8929  047470  012700  047606  15$:    MOV     #OVFTT,R0       ;SEE IF FAILURE IS ANTICIPATED
8930  047474  010102                  MOV     R1,R2           ;FAILURE.
8931  047476  062702  000014          ADD     #14,R2
8932  047502  012703  000002          MOV     #2,R3
8933  047506  022022          16$:    CMP     (R0)+,(R2)+
8934  047510  001007                  BNE     17$             ;BRANCH IF NOT ANTICIPATED.
8935  047512  077303                  SOB     R3,16$
8936
8937  047514  010102                  MOV     R1,R2           ;ERROR WAS ANTICIPATED SO RETURN
8938  047516  062702  000034          ADD     #34,R2          ;TO THE ERROR REPORT IN THE CALLING
8939  047522  010237  001236          MOV     R2,@#$TMP2      ;ROUTINE.
8940  047526  000112                  JMP     (R2)
8941
8942  047530  005761  000026  17$:    TST     26(R1)          ;RESULT WAS N   ANTICIPATED
8943                                                          ;SO ERROR MUST BE REPORTED HERE.
```

```
8944                                                    ;FIRST SEE IF ARGUMENTS SHOULD
8945                                                    ;HAVE RESULTED IN OVERFLOW OR UNDER
8946                                                    ;FLOW BY LOOKING AT THE FLAG.
8947  047534  001002                    BNE     19$     ;BRANCH IF UNDERFLOW EXPECTED.
8948
8949                                                    ;REPORT RESULT INCORRECT, EXPECTING
8950  047536  104002              18$:  ERROR   2       ;OVERFLOW.
8951  047540  000733                    BR      4$
8952
8953  047542                      19$:                  ;REPORT RESULT INCORRECT, EXPECTING
8954  047542  104002              20$:  ERROR   2       ;UNDERFLOW.
8955  047544  000731                    BR      4$
8956
8957                              ;IF NO FP TRAP OCCURS COME HERE.
8958  047546  170204              25$:  STFPS   R4      ;GET FPS.
8959  047550  170305                    STST    R5      ;GET FEC.
8960  047552  012700  000200            MOV     #200,R0 ;SET FD MODE.
8961  047556  170100                    LDFPS   R0
8962  047560  012700  047606            MOV     #OVFTT,R0       ;GET THE RESULT.
8963  047564  174010                    STD     AC0,(R0)
8964  047566  010437  001250            MOV     R4,@#$TMP7
8965  047572  010537  001254            MOV     R5,@#$TMP11
8966  047576  010102                    MOV     R1,R2   ;ERROR WAS ANTICIPATED SO
8967  047600  062702  000030            ADD     #30,R2  ;RETURN TO THE ERROR REPORT OF THE
8968                                                    ;CALLING ROUTINE.
8969  047604  000112                    JMP     (R2)
8970
8971  047606  000000  000000  000000  OVFTT:  .WORD   0,0,0,0
8972  047614  000000
8973
8974  047616                      KKKDONE:
8975  047616  104413                    RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
8976                                                    ;SEE IF THE USER HAS EXPRESSED
8977                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
8978                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
8979                                                    ;THE USER TYPED CONTROL G?).
8980
8981
8982
8983
8984                              ;;*****************************************************************
8985                              ;*TEST 53       UNDER\OVER FLOW, USING MULD WITH TRAPS ENABLED, TEST
8986                              ;*
8987                              ;*THIS IS A TEST OF THE OVER FLOW AND UNDER FLOW CONDITIONS USING THE
8988                              ;*MULD INSTRUCTION WITH TRAPS ENABLED. A SUBROUTINE IS USED TO SET UP
8989                              ;*THE OPERANDS, EXECUTE THE MULD INSTRUCTION AND CHECK THE RESULTS.
8990                              ;*
8991                              ;;*****************************************************************
8992  047620  000004              TST53:  SCOPE
8993
8994                              ;UNDERFLOW, EXPONENT OF RESULT = -129
8995  047622                      LLL1:
8996  047622  104414                    LPERR           ;SET UP THE LOOP ON ERROR ADDRESS.
8997  047624  004737  050146            JSR     PC,@#OVUNDT
8998  047630  020052  125252      1$:   .WORD   20052,125252    ;AC
8999  047634  125252  125252            .WORD   125252,125252
```

```
 9000  047640  020300  000000  000000  2$:     .WORD   20300,0,0,0      ;FSRC
 9001  047646  000000
 9002  047650  000177  177777  177777  3$:     .WORD   177,-1,-1,-1     ;RES
 9003  047656  177777
 9004  047660  000177  177777          4$:     .WORD   177,-1           ;ERROR RES.
 9005  047664  125252  125252                  .WORD   125252,125252
 9006  047670  002200                  5$:     2200                     ;FPS BEFORE EXECUTION.
 9007  047672  102204                          102204                   ;FPS AFTER EXECUTION.
 9008  047674  000012                  6$:     12                       ;FEC
 9009  047676  177777                          -1                       ;FLAG
 9010  047700  104002                  7$:     ERROR   2                ;ST 331 (BUT FIU) NO TRAP.
 9011  047702  000401                          BR      8$
 9012  047704  104002                          ERROR   2                ;ST 155 (BUT FD)
 9013  047706                          8$:
 9014
 9015                                  ;UNDERFLOW, EXPONENT OF THE RESULT = -193
 9016  047706                          LLL2:
 9017  047706  104414                          LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
 9018  047710  004737  050146                  JSR     PC,@#OVUNDT
 9019  047714  010327  127272          1$:     .WORD   10327,127272     ;AC
 9020  047720  036363  045454                  .WORD   36363,45454
 9021  047724  010000  000000  000000  2$:     .WORD   10000,0,0,0      ;FSRC
 9022  047732  000000
 9023  047734  060127  127272          3$:     .WORD   60127,127272     ;RES
 9024  047740  036363  045454                  .WORD   36363,45454
 9025  047744  177777  177777  177777  4$:     .WORD   -1,-1,-1,-1      ;ERROR RES.
 9026  047752  177777
 9027  047754  007217                  5$:     7217                     ;FPS BEFORE EXECUTION.
 9028  047756  107200                          107200                   ;FPS AFTER EXECUTION.
 9029  047760  000012                  6$:     12                       ;FEC
 9030  047762  177777                          -1                       ;FLAG
 9031  047764  104002                  7$:     ERROR   2                ;ST 137 (BUT FIU) NO TRAP.
 9032  047766  000401                          BR      8$
 9033  047770  104002                          ERROR   2
 9034  047772                          8$:
 9035
 9036                                  ;OVERFLOW, EXPONENT OF THE RESULT = 128
 9037  047772                          LLL3:
 9038  047772  104414                          LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
 9039  047774  004737  050146                  JSR     PC,@#OVUNDT
 9040  050000  060252  125252          1$:     .WORD   60252,125252     ;AC
 9041  050004  125252  125252                  .WORD   125252,125252    ;FSRC
 9042  050010  160100  000000  000000  2$:     .WORD   160100,0,0,0     ;FSRC
 9043  050016  000000
 9044  050020  100177  177777  177777  3$:     .WORD   100177,-1,-1,-1  ;RES
 9045  050026  177777
 9046  050030  100177  177777          4$:     .WORD   100177,-1        ;ERROR RES.
 9047  050034  125252  125252                  .WORD   125252,125252
 9048  050040  001200                  5$:     1200                     ;FPS BEFORE EXECUTION.
 9049  050042  101216                          101216                   ;FPS AFTER EXECUTION.
 9050  050044  000010                  6$:     10                       ;FEC
 9051  050046  000000                          0                        ;FLAG
 9052  050050  104002                  7$:     ERROR   2                ;ST 333 (BUT FIV) NO TRAP.
 9053  050052  000401                          BR      8$
 9054  050054  104002                          ERROR   2                ;ST 700 (BUT FD).
 9055  050056                          8$:
```

```
 9056
 9057                                   ;OVERFLOW, EXPONENT OF THE RESULT = 130
 9058  050056                           LLL4:
 9059  050056  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9060  050060  004737  050146                   JSR     PC,@#OVUNDT
 9061  050064  060345  067654           1$:     .WORD   60345,67654     ;AC
 9062  050070  056765  045676                   .WORD   56765,45676
 9063  050074  060200  000000  000000 2$:      .WORD   60200,0,0,0     ;FSRC
 9064  050102  000000
 9065  050104  000345  067654           3$:     .WORD   345,67654       ;RES
 9066  050110  056765  045676                   .WORD   56765,45676
 9067  050114  177777  177777  177777 4$:      .WORD   -1,-1,-1,-1     ;ERROR RES.
 9068  050122  177777
 9069  050124  007215                     5$:   7215                    ;FPS BEFORE EXECUTION.
 9070  050126  107202                           107202                  ;FPS AFTER EXECUTION.
 9071  050130  000010                     6$:   10                      ;FEC
 9072  050132  000000                           0                       ;FLAG
 9073  050134  104002                     7$:   ERROR   2               ;ST 133 (BUT FIV) NO TRAP
 9074  050136  000401                           BR      8$
 9075  050140  104002                           ERROR   2
 9076  050142  000137  050560           8$:     JMP     @#LLLDONE
 9077
 9078                                   ;THIS SUBROUTINE, OVUNDT, IS USED TO SET UP THE OPERANDS, EXECUTE
 9079                                   ;THE MULD INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
 9080                                   ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
 9081                                   ;TO IT IS MADE THUS:
 9082                                   ;
 9083                                   ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
 9084                                   ;               FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
 9085                                   ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
 9086                                   ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
 9087                                   ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
 9088                                   ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
 9089                                   ;               FEC:    .WORD   X               ;EXPECTED FEC
 9090                                   ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
 9091                                   ;               ERR1:   ERROR   X               ;TRAP ERROR.
 9092                                   ;               BR      CONT
 9093                                   ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
 9094                                   ;               CONT:                           ;RETURN ADDRESS
 9095                                   ;
 9096                                   ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
 9097                                   ;THE MULD INSTRUCTION IS EXECUTED. IF THE TRAP OCCURS THEN THE
 9098                                   ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
 9099                                   ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNDT RETURNS CONTROL
 9100                                   ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNDT
 9101                                   ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. THE FEC IS TREATED
 9102                                   ;IN THE SAME WAY. IF THE RESULT OF THE
 9103                                   ;MULF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
 9104                                   ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
 9105                                   ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNDT
 9106                                   ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
 9107                                   ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNDT WILL
 9108                                   ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
 9109                                   ;IF NO TRAP OCCURS CONTROL IS PASSED TO ERR1.
 9110                                   ;NOTE THAT OVUNDNT USES THE FLAG
 9111                                   ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
```

```
 9112                                   ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).
 9113
 9114  050146  012601                   OVUNDT: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
 9115  050150  012700  000200                   MOV     #200,R0         ;SET FD MODE.
 9116  050154  170100                           LDFPS   R0
 9117
 9118  050156  010100                           MOV     R1,R0           ;LOAD AC0, OPERAND.
 9119  050160  172410                           LDD     (R0),AC0
 9120
 9121  050162  010102                           MOV     R1,R2           ;SAVE THE DATA PATTERNS IN CASE OF
 9122  050164  010237  001240                   MOV     R2,@#$TMP3      ;ERROR.
 9123  050170  062702  000010                   ADD     #10,R2
 9124  050174  010237  001242                   MOV     R2,@#$TMP4
 9125  050200  062702  000010                   ADD     #10,R2
 9126  050204  010237  001244                   MOV     R2,@#$TMP5
 9127  050210  016137  000042  001252           MOV     42(R1),@#$TMP10
 9128  050216  012737  050550  001246           MOV     #OVDTT,@#$TMP6
 9129
 9130  050224  016100  000040                   MOV     40(R1),R0       ;LOAD THE FPS.
 9131  050230  170100                           LDFPS   R0
 9132  050232  012737  050254  001236           MOV     #1$,@#$TMP2
 9133  050240  012737  050264  000244           MOV     #50$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
 9134                                           ;OF ERROR.
 9135  050246  010100                           MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
 9136  050250  062700  000010                   ADD     #10,R0
 9137
 9138  050254  171010                   1$:     MULD    (R0),AC0        ;TEST INSTRUCTION. SHOULD CAUSE TRAP.
 9139  050256  170000                   2$:     CFCC
 9140
 9141  050260  000137  050510                   JMP     @#25$           ;FAILURE, NO TRAP.
 9142
 9143  050264  011602                   50$:    MOV     (SP),R2         ;TRAP TO HERE AND SEE IF THE PC OF THE
 9144  050266  020227  050256                   CMP     R2,#2$          ;TRAP WAS THAT OF THE MULF INSTRUCTION.
 9145  050272  001402                           BEQ     51$             ;BRANCH IF YES.
 9146  050274  000137  062524                   JMP     @#FPSPUR        ;OTHERWISE REPORT SPURIOUS FP ERROR.
 9147
 9148  050300  022626                   51$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
 9149  050302  170204                           STFPS   R4              ;GET FPS.
 9150  050304  170305                           STST    R5              ;GET FEC.
 9151  050306  012700  000200                   MOV     #200,R0         ;SET FD MODE.
 9152  050312  170100                           LDFPS   R0
 9153  050314  012700  050550                   MOV     #OVDTT,R0       ;GET THE RESULT.
 9154  050320  174010                           STD     AC0,(R0)
 9155  050322  010437  001250                   MOV     R4,@#$TMP7
 9156  050326  010537  001254                   MOV     R5,@#$TMP11
 9157
 9158  050332  012700  050550                   MOV     #OVDTT,R0       ;CHECK THE RESULT.
 9159  050336  010102                           MOV     R1,R2
 9160  050340  062702  000020                   ADD     #20,R2
 9161  050344  012703  000004                   MOV     #4,R3
 9162  050350  022022                   3$:     CMP     (R0)+,(R2)+
 9163  050352  001027                           BNE     15$             ;BRANCH IF INCORRECT.
 9164  050354  077303                           SOB     R3,3$
 9165
 9166  050356  026104  000042                   CMP     42(R1),R4       ;WAS FPS CORRECT?
 9167  050362  001014                           BNE     10$             ;BRANCH IF FPS IS INCORRECT.
```

```
9168
9169  050364  026105  000044          CMP     44(R1),R5       ;IS FEC CORRECT?
9170  050370  001002                  BNE     5$              ;IF INCORRECT BRANCH.
9171  050372  000161  000056  4$:     JMP     56(R1)          ;RETURN, TEST COMPLETED.
9172
9173                                  ;REPORT INCORRECT FEC.
9174  050376  005761  000046  5$:     TST     46(R1)          ;WAS THE RESULT OVERFLOW OR UNDERFLOW?
9175  050402  001002                  BNE     7$              ;BRANCH IF UNDERFLOW.
9176
9177                                                          ;REPORT BAD FEC ON EXPECTED OVERFLOW.
9178  050404  104002          6$:     ERROR   2
9179  050406  000771                  BR      4$
9180
9181  050410                  7$:                             ;REPORT BAD FEC ON EXPECTED UNDERFLOW.
9182  050410  104002          8$:     ERROR   2
9183  050412  000767                  BR      4$
9184
9185                                  ;REPORT INCORRECT FPS.
9186  050414  005761  000046  10$:    TST     46(R1)          ;WAS THE RESULT OVER OR UNDER FLOW?
9187  050420  001002                  BNE     12$             ;BRANCH IF UNDERFLOW.
9188
9189                                                          ;REPORT FPS BAD AFTER OVERFLOW.
9190  050422  104002          11$:    ERROR   2
9191  050424  000762                  BR      4$
9192
9193  050426                  12$:                            ;REPORT FPS BAD AFTER UNDERFLOW.
9194  050426  104002          13$:    ERROR   2
9195  050430  000760                  BR      4$
9196
9197                                  ;RESULT INCORRECT.
9198  050432  012700  050550  15$:    MOV     #OVDTT,R0       ;SEE IF FAILURE IS ANTICIPATED
9199  050436  010102                  MOV     R1,R2           ;FAILURE.
9200  050440  062702  000030          ADD     #30,R2
9201  050444  012703  000004          MOV     #4,R3
9202  050450  022022          16$:    CMP     (R0)+,(R2)+
9203  050452  001007                  BNE     17$             ;BRANCH IF NOT ANTICIPATED.
9204  050454  077303                  SOB     R3,16$
9205
9206  050456  010102                  MOV     R1,R2           ;ERROR WAS ANTICIPATED SO RETURN
9207  050460  062702  000054          ADD     #54,R2          ;TO THE ERROR REPORT IN THE CALLING
9208  050464  010237  001236          MOV     R2,@#$TMP2      ;ROUTINE.
9209  050470  000112                  JMP     (R2)
9210
9211  050472  005761  000046  17$:    TST     46(R1)          ;RESULT WAS NOT ANTICIPATED
9212                                                          ;SO ERROR MUST BE REPORTED HERE.
9213                                                          ;FIRST SEE IF ARGUMENTS SHOULD
9214                                                          ;HAVE RESULTED IN OVERFLOW OR UNDER
9215                                                          ;FLOW BY LOOKING AT THE FLAG.
9216  050476  001002                  BNE     19$             ;BRANCH IF UNDERFLOW EXPECTED.
9217
9218                                                          ;REPORT RESULT INCORRECT, EXPECTING
9219  050500  104002          18$:    ERROR   2               ;OVERFLOW.
9220  050502  000733                  BR      4$
9221
9222  050504                  19$:                            ;REPORT RESULT INCORRECT, EXPECTING
9223  050504  104002          20$:    ERROR   2               ;UNDERFLOW.
```

```
9224  050506  000731                          BR      4$
9225
9226                                  ;IF NO FP TRAP OCCURS COME HERE.
9227  050510  170204                  25$:    STFPS   R4              ;GET FPS.
9228  050512  170305                          STST    R5              ;GET FEC.
9229  050514  012700  000200                  MOV     #200,R0         ;SET FD MODE.
9230  050520  170100                          LDFPS   R0
9231  050522  012700  050550                  MOV     #OVDTT,R0       ;GET THE RESULT.
9232  050526  174010                          STD     AC0,(R0)
9233  050530  010437  001250                  MOV     R4,@#$TMP7
9234  050534  010537  001254                  MOV     R5,@#$TMP11
9235  050540  010102                          MOV     R1,R2           ;ERROR WAS ANTICIPATED SO
9236  050542  062702  000050                  ADD     #50,R2          ;RETURN TO THE ERROR REPORT OF THE
9237                                                                  ;CALLING ROUTINE.
9238  050546  000112                          JMP     (R2)
9239
9240  050550  000000  000000  000000  OVDTT:  .WORD   0,0,0,0
9241  050556  000000
9242
9243  050560                          LLLDONE:
9244  050560  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
9245                                                                  ;SEE IF THE USER HAS EXPRESSED
9246                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
9247                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
9248                                                                  ;THE USER TYPED CONTROL G?).
9249
9250
9251
9252
9253
9254                                  ;;*****************************************************************
9255                                  ;*TEST 54        MODF TEST
9256                                  ;*
9257                                  ;*THIS IS A TEST OF THE MODF INSTRUCTION, WHICH MAKES USE OF
9258                                  ;*A SUBROUTINE TO SET UP THE OPERANDS, EXECUTE THE MODF INSTRUCTION
9259                                  ;*AND CHECK THE RESULTS.
9260                                  ;*
9261                                  ;;*****************************************************************
9262  050562  000004                  TST54:  SCOPE
9263
9264                                  ;MODF WITH (FSRC=AC=0)
9265  050564                          GGG1:
9266  050564  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9267  050566  004737  051650                  JSR     PC,@#MODFSUB
9268  050572  000000  000000          1$:     .WORD   0,0             ;AC
9269  050576  000000  000000          2$:     .WORD   0,0             ;FSRC
9270  050602  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
9271  050606  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
9272  050612  177777  177777          5$:     .WORD   -1,-1           ;ERROR FRACTIONAL RES.
9273  050616  177777  177777          6$:     .WORD   -1,-1           ;ERROR INGETER RES.
9274  050622  000013                  7$:     13                      ;FPS BEFORE EXECUTION.
9275  050624  000004                          4                       ;FPS AFTER EXECUTION.
9276  050626  104002                  8$:     ERROR   2               ;STORE SINGLE ZERO BAD.
9277  050630  000401                          BR      9$
9278  050632  104002                          ERROR   2               ;AC V 1 <= ZERO FAILED.
9279  050634                          9$:
```

```
9280
9281                                    ;MODF TEST, WITH (FSRC=0)
9282  050634                            GGG2:
9283  050634  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9284  050636  004737  051650                    JSR     PC,@#MODFSUB
9285  050642  123456  076543            1$:     .WORD   123456,76543    ;AC
9286  050646  000000  000000            2$:     .WORD   0,0             ;FSRC
9287  050652  000000  000000            3$:     .WORD   0,0             ;FRACTIONAL RES.
9288  050656  000000  000000            4$:     .WORD   0,0             ;INTEGER RESULT.
9289  050662  123456  076543            5$:     .WORD   123456,76543    ;ERROR FRACTIONAL RES.
9290  050666  177777  177777            6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
9291  050672  000000                    7$:     0                       ;FPS BEFORE EXECUTION.
9292  050674  000004                            4                       ;FPS AFTER EXECUTION.
9293  050676  104002                    8$:     ERROR   2               ;STORE ZERO FAILURE.
9294  050700  000401                            BR      9$
9295  050702  104002                            ERROR   2
9296  050704                            9$:
9297
9298                                    ;MODF TEST WITH (AC=0)
9299  050704                            GGG3:
9300  050704  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9301  050706  004737  051650                    JSR     PC,@#MODFSUB
9302  050712  000000  000000            1$:     .WORD   0,0             ;AC
9303  050716  076543  021234            2$:     .WORD   76543,21234     ;FSRC
9304  050722  000000  000000            3$:     .WORD   0,0             ;FRACTIONAL RES.
9305  050726  000000  000000            4$:     .WORD   0,0             ;INTEGER RES.
9306  050732  000000  000000            5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
9307  050736  177777  177777            6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
9308  050742  000003                    7$:     3                       ;FPS BEFORE EXECUTION.
9309  050744  000004                            4                       ;FPS AFTER EXECUTION.
9310  050746  104002                    8$:     ERROR   2               ;RES.BAD
9311  050750  000401                            BR      9$
9312  050752  104002                            ERROR   2
9313  050754                            9$:
9314
9315                                    ;MODF TEST WITH EXPONENT OF THE RESULT = 25
9316  050754                            GGG4:
9317  050754  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9318  050756  004737  051650                    JSR     PC,@#MODFSUB
9319  050762  046252  125252            1$:     .WORD   46252,125252    ;AC
9320  050766  040300  000000            2$:     .WORD   40300,0         ;FSRC
9321  050772  000000  000000            3$:     .WORD   0,0             ;FRACTIONAL RES.
9322  050776  046377  177777            4$:     .WORD   46377,-1        ;INTEGER RES.
9323  051002  046252  125252            5$:     .WORD   46252,125252    ;ERROR FRACTIONAL RES.
9324  051006  040300  000000            6$:     .WORD   40300,0         ;ERROR INTEGER RES.
9325  051012  000013                    7$:     13                      ;FPS BEFORE EXECUTION.
9326  051014  000004                            4                       ;FPS AFTER EXECUTION.
9327  051016  104002                    8$:     ERROR   2               ;ST 134
9328  051020  000401                            BR      9$
9329  051022  104002                            ERROR   2
9330  051024                            9$:
9331
9332                                    ;MODF TEST WITH EXPONENT OF THE RESULT = 127
9333  051024                            GGG5:
9334  051024  104414                            LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9335  051026  004737  051650                    JSR     PC,@#MODFSUB
```

```
 9336  051032  077652  125252          1$:     .WORD   77652,125252    ;AC
 9337  051036  040300  000000          2$:     .WORD   40300,0         ;FSRC
 9338  051042  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
 9339  051046  077777  177777          4$:     .WORD   77777,-1        ;INTEGER RES.
 9340  051052  077652  125252          5$:     .WORD   77652,125252    ;ERROR FRACTIONAL RES.
 9341  051056  040300  000000          6$:     .WORD   40300,0         ;ERROR INTEGER RES.
 9342  051062  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
 9343  051064  000004                          4                       ;FPS AFTER EXECUTION.
 9344  051066  104002                  8$:     ERROR   2
 9345  051070  000401                          BR      9$
 9346  051072  104002                          ERROR   2
 9347  051074                          9$:
 9348
 9349                                  ;MODF TEST WITH EXPONENT OF RESULT   25
 9350  051074                          GGG6:
 9351  051074  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9352  051076  004737  051650                  JSR     PC,@#MODFSUB
 9353  051102  046200  000001          1$:     .WORD   46200,1         ;AC
 9354  051106  040340  000000          2$:     .WORD   40340,0         ;FSRC
 9355  051112  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
 9356  051116  046340  000001          4$:     .WORD   46340,1         ;INTEGER RES.
 9357  051122  040000  000000          5$:     .WORD   40000,0         ;ERROR FRACTIONAL RES.
 9358  051126  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
 9359  051132  000013                  7$:     13                      ;FPS BEFORE EXECUTION.
 9360  051134  000004                          4                       ;FPS AFTER EXECUTION.
 9361  051136  104002                  8$:     ERROR   2               ;BAD CONSTANT (NOT 24),
 9362                                                                  ;OR ST 525 TO 050 INTO 150.
 9363  051140  000401                          BR      9$
 9364  051142  104002                          ERROR   2
 9365  051144                          9$:
 9366
 9367                                  ;MODF   TEST WITH EXPONENT OF THE RESULT - 24
 9368  051144                          GGG7:
 9369  051144  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9370  051146  004737  051650                  JSR     PC,@#MODFSUB
 9371  051152  046000  000001          1$:     .WORD   46000,1         ;AC
 9372  051156  040340  000000          2$:     .WORD   40340,0         ;FSRC
 9373  051162  040100  000000          3$:     .WORD   40100,0         ;FRACTIONAL RES.
 9374  051166  046140  000001          4$:     .WORD   46140,1         ;INTEGER RESULT.
 9375  051172  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
 9376  051176  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
 9377  051202  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
 9378  051204  000000                          0                       ;FPS AFTER EXECUTION.
 9379  051206  104002                  8$:     ERROR   2               ;BAD CONSTANT USED (NOT 24)
 9380                                                                  ;OR ST 525 TO 150 INTO 050
 9381  051210  000401                          BR      9$
 9382  051212  104002                          ERROR   2
 9383  051214                          9$:
 9384
 9385                                  ;MODF TEST WITH EXPONENT OF THE RESULT = 10
 9386  051214                          GGG8:
 9387  051214  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9388  051216  004737  051650                  JSR     PC,@#MODFSUB
 9389  051222  042577  177777          1$:     .WORD   42577,-1        ;AC
 9390  051226  040200  000000          2$:     .WORD   40200,0         ;FSRC
 9391  051232  040177  176000          3$:     .WORD   40177,176000    ;FRACTIONAL RES.
```

```
 9392  051236  042577  140000          4$:     .WORD   42577,140000    ;INTEGER RES.
 9393  051242  177777  177777          5$:     .WORD   -1,-1           ;ERROR FRACTIONAL RES.
 9394  051246  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
 9395  051252  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
 9396  051254  000000                          0                       ;FPS AFTER EXECUTION.
 9397  051256  104002                  8$:     ERROR   2
 9398  051260  000401                          BR      9$
 9399  051262  104002                          ERROR   2
 9400  051264                          9$:     ;
 9401
 9402                                  ;MODF TEST WITH THE EXPONENT OF THE RESULT = 10
 9403  051264                          GGG9:
 9404  051264  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9405  051266  004737  051650                  JSR     PC,@#MODFSUB
 9406  051272  042577  140001          1$:     .WORD   42577,140001    ;AC
 9407  051276  040200  000000          2$:     .WORD   40200,0         ;FSRC
 9408  051302  034600  000000          3$:     .WORD   34600,0         ;FRACTIONAL RES.
 9409  051306  042577  140000          4$:     .WORD   42577,140000    ;INTEGER RES.
 9410  051312  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
 9411  051316  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
 9412  051322  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
 9413  051324  000000                          0                       ;FPS AFTER EXECUTION.
 9414  051326  104002                  8$:     ERROR   2               ;ST 532 TO 122 INTO NORMALIZE.
 9415  051330  000401                          BR      9$
 9416  051332  104002                          ERROR   2
 9417  051334                          9$:
 9418
 9419                                  ;MODF TEST WITH EXPONENT OF THE RESULT = 9
 9420  051334                          GGG10:
 9421  051334  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9422  051336  004737  051650                  JSR     PC,@#MODFSUB
 9423  051342  042377  100000          1$:     .WORD   42377,100000    ;AC
 9424  051346  040200  000000          2$:     .WORD   40200,0         ;FSRC
 9425  051352  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
 9426  051356  042377  100000          4$:     .WORD   42377,100000    ;INTEGER RES.
 9427  051362  177777  177777          5$:     .WORD   -1,-1           ;ERROR FRACTIONAL RES.
 9428  051366  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
 9429  051372  000013                  7$:     13                      ;FPS BEFORE EXECUTION.
 9430  051374  000004                          4                       ;FPS AFTER EXECUTION.
 9431  051376  104002                  8$:     ERROR   2
 9432  051400  000401                          BR      9$
 9433  051402  104002                          ERROR   2
 9434  051404                          9$:
 9435
 9436                                  ;MODF TEST WITH EXPONENT OF THE RESULT = 0
 9437  051404                          GGG11:
 9438  051404  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9439  051406  004737  051650                  JSR     PC,@#MODFSUB
 9440  051412  040177  177777          1$:     .WORD   40177,-1        ;AC
 9441  051416  040200  000000          2$:     .WORD   40200,0         ;FSRC
 9442  051422  040177  177777          3$:     .WORD   40177,-1        ;FRACTIONAL RES.
 9443  051426  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
 9444  051432  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
 9445  051436  040177  177777          6$:     .WORD   40177,-1        ;ERROR INTEGER RES.
 9446  051442  000017                  7$:     17                      ;FPS BEFORE EXECUTION.
 9447  051444  000000                          0                       ;FPS AFTER EXECUTION.
```

```
 9448  051446  104002                  8$:     ERROR   2               ;ST 041 TO 046 INTO 246.
 9449  051450  000401                          BR      9$
 9450  051452  104002                          ERROR   2
 9451  051454                          9$:
 9452
 9453                                  ;MODF TEST WITH EXPONENT OF THE RESULT = -15
 9454  051454                          GGG12:
 9455  051454  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9456  051456  004737  051650                  JSR     PC,@#MODFSUB
 9457  051462  034377  177777          1$:     .WORD   34377,-1        ;AC
 9458  051466  040200  000000          2$:     .WORD   40200,0         ;FSRC
 9459  051472  034377  177777          3$:     .WORD   34377,-1        ;FRACTIONAL RES.
 9460  051476  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
 9461  051502  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
 9462  051506  034377  177777          6$:     .WORD   34377,-1        ;ERROR INTEGER RES.
 9463  051512  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
 9464  051514  000000                          0                       ;FPS AFTER EXECUTION.
 9465  051516  104002                  8$:     ERROR   2
 9466  051520  000401                          BR      9$
 9467  051522  104002                          ERROR   2
 9468  051524                          9$:
 9469
 9470                                  ;MODF TEST WITH EXPONENT OF RESULT = -64, IN ROUND MODE
 9471  051524                          GGG13:
 9472  051524  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9473  051526  004737  051650                  JSR     PC,@#MODFSUB
 9474  051532  020000  000001          1$:     .WORD   20000,1         ;AC
 9475  051536  040300  000000          2$:     .WORD   40300,0         ;FSRC
 9476  051542  020100  000002          3$:     .WORD   20100,2         ;FRACTIONAL RES.
 9477  051546  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
 9478  051552  020100  000001          5$.     .WORD   20100,1         ;ERROR FRACTIONAL RES.
 9479  051556  000000  000000          6$:     .WORD   0,0             ;ERROR INTEGER RES.
 9480  051562  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
 9481  051564  000000                          0                       ;FPS AFTER EXECUTION.
 9482  051566  104002                  8$:     ERROR   2               ;ROUND TRUNK, ST 126 INTO ROUND.
 9483  051570  000401                          BR      9$
 9484  051572  104002                          ERROR   2
 9485  051574                          9$:
 9486
 9487                                  ;MODF TEST WITH EXPONENT OF RESULT = 11
 9488  051574                          GGG14:
 9489  051574  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9490  051576  004737  051650                  JSR     PC,@#MODFSUB
 9491  051602  142777  170000          1$:     .WORD   142777,170000   ;AC
 9492  051606  040200  000000          2$:     .WORD   40200,0         ;FSRC
 9493  051612  140000  000000          3$:     .WORD   140000,0        ;FRACTIONAL RES.
 9494  051616  142777  160000          4$:     .WORD   142777,160000   ;INTEGER RES.
 9495  051622  040000  000000          5$:     .WORD   40000,0         ;ERROR FRACTIONAL RES.
 9496  051626  042777  160000          6$:     .WORD   42777,160000    ;ERROR INTEGER RES.
 9497  051632  000007                  7$:     7                       ;FPS BEFORE EXECUTION.
 9498  051634  000010                          10                      ;FPS AFTER EXECUTION.
 9499  051636  104002                  8$:     ERROR   2               ;SIGN OF FRACTION.
 9500  051640  000401                          BR      9$
 9501  051642  104002                          ERROR   2               ;SIGN OF INTEGER.
 9502  051644  000167  000366          9$:     JMP     GGGDONE         ;GO TO NEXT TEST.
 9503
```

```
9504
9505                            ;THIS SUBROUTINE, MODFSUB, IS CALLED TO SETUP THE
9506                            ;OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK THE RESULTS.
9507                            ;IT IS CALLED THUS:
9508                            ;
9509                            ;               ACARG:  .WORD   X,X             ;AC OPERAND
9510                            ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
9511                            ;               FRES:   .WORD   X,X             ;FRACTIONAL RESULT
```

```
9512                                    ;              INTRES: .WORD   X,X             ;INTEGER RESULT
9513                                    ;              ERFRES: .WORD   X,X             ;ERROR FRACTION RESULT
9514                                    ;              ERINTRES:.WORD  X,X             ;ERROR INTEGER RESULT
9515                                    ;              FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
9516                                    ;              FPSA:   .WORD   X               ;FPS AFTER EXECUTION
9517                                    ;              ERR1:   ERROR   X               ;FRACTION ERROR
9518                                    ;                      BR      CONT
9519                                    ;              ERR2:   ERROR   X               ;INTEGER ERROR
9520                                    ;              CONT:                           ;RETURN ADDRESS
9521                                    ;
9522                                    ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODF
9523                                    ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
9524                                    ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
9525                                    ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
9526                                    ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
9527                                    ;THEN MODFSUB WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
9528                                    ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
9529                                    ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
9530                                    ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
9531                                    ;FAILURE MATCHES THE TRUE RESULT THEN  MODFSUB PASSES CONTROL TO THE
9532                                    ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
9533                                    ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
9534                                    ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
9535                                    ;IF A MATCH IS MADE HOWEVER, MODFSUB WILL RETURN CONTROL TO THE ERROR
9536                                    ;CALL AT ERR2.
9537
9538  051650  012601                    MODFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
9539  051652  012700  000200                    MOV     #200,R0         ;SET FD MODE.
9540  051656  170100                     LDFPS   R0
9541  051660  010100                     MOV     R1,R0           ;SET UP AC0
9542  051662  172410                     LDD     (R0),AC0
9543  051664  012700  052226                    MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
9544  051670  172510                     LDD     (R0),AC1
9545  051672  016100  000030                    MOV     30(R1),R0       ;SET UP THE FPS.
9546  051676  170100                     LDFPS   R0
9547  051700  012737  051714  001236             MOV     #1$,@#$TMP2
9548  051706  010100                     MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
9549  051710  062700  000004                    ADD     #4,R0
9550
9551  051714  171410                    1$:     MODF    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
9552
9553  051716  170204                     STFPS   R4              ;GET THE FPS.
9554  051720  012700  000200                    MOV     #200,R0         ;SET FD MODE.
9555  051724  170100                     LDFPS   R0
9556  051726  012700  052206                    MOV     #MODFT0,R0      ;GET THE FRACTIONAL RESULT.
9557  051732  174010                     STD     AC0,(R0)
9558  051734  012700  052216                    MOV     #MODFT1,R0      ;GET THE INTEGER RESULT.
9559  051740  174110                     STD     AC1,(R0)
9560
9561  051742  010102                     MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
9562  051744  010237  001240                    MOV     R2,@#$TMP3
9563  051750  062702  000004                    ADD     #4,R2
9564  051754  010237  001242                    MOV     R2,@#$TMP4
9565  051760  062702  000004                    ADD     #4,R2
9566  051764  010237  001244                    MOV     R2,@#$TMP5
9567  051770  062702  000004                    ADD     #4,R2
```

```
 9568  051774  010237  001246                    MOV     R2,@#$TMP6
 9569  052000  012737  052206  001250            MOV     #MODFT0,@#$TMP7
 9570  052006  012737  052216  001252            MOV     #MODFT1,@#$TMP10
 9571  052014  010437  001254                    MOV     R4,@#$TMP11
 9572  052020  016137  000032  001256            MOV     32(R1),@#$TMP12
 9573
 9574  052026  012702  052206                    MOV     #MODFT0,R2      ;CHECK THE FRACTIONAL RESULT.
 9575  052032  026112  000010                    CMP     10(R1),(R2)
 9576  052036  001022                            BNE     10$             ;BRANCH IF INCORRECT.
 9577  052040  026162  000012  000002            CMP     12(R1),2(R2)
 9578  052046  001016                            BNE     10$
 9579
 9580  052050  012702  052216                    MOV     #MODFT1,R2      ;CHECK THE INTEGER RESULT.
 9581  052054  026112  000014                    CMP     14(R1),(R2)
 9582  052060  001026                            BNE     15$             ;BRANCH IF INCORRECT.
 9583  052062  026162  000016  000002            CMP     16(R1),2(R2)
 9584  052070  001022                            BNE     15$
 9585
 9586  052072  026104  000032                    CMP     32(R1),R4       ;CHECK THE FPS.
 9587  052076  001034                            BNE     20$             ;BRANCH IF INCORRECT.
 9588
 9589  052100  000161  000042          9$:       JMP     42(R1)          ;RETURN.
 9590
 9591                                    ;FRACTIONAL ERROR.
 9592  052104  026112  000020          10$:      CMP     20(R1),(R2)     ;WAS THE ERROR ANTICIPATED?
 9593  052110  001010                            BNE     11$             ;BRANCH IF NOT ANTICIPATED.
 9594  052112  026162  000022  000002            CMP     22(R1),2(R2)
 9595  052120  001004                            BNE     11$
 9596  052122  010102                            MOV     R1,R2           ;THE ERROR WAS ANTICIPATED SO
 9597  052124  062702  000034                    ADD     #34,R2          ;RETURN TO THE ERROR REPORT AT THE
 9598                                                                    ;CALLING ROUTINE.
 9599  052130  000112                            JMP     (R2)
 9600
 9601  052132                          11$:                              ;THE ERROR WAS NOT ANTICIPATED SO
 9602  052132  104002                  12$:      ERROR   2               ;REPORT THE INCORRECT FRACTION HERE.
 9603  052134  000761                            BR      9$
 9604
 9605                                    ;INTEGER ERROR.
 9606  052136  026112  000024          15$:      CMP     24(R1),(R2)     ;WAS THIS ERROR ANTICIPATED?
 9607  052142  001010                            BNE     16$             ;BRANCH IF NOT.
 9608  052144  026162  000026  000002            CMP     26(R1),2(R2)
 9609  052152  001004                            BNE     16$
 9610  052154  010102                            MOV     R1,R2           ;THE ERROR WAS ANTICIAPTED SO RETURN
 9611  052156  062702  000040                    ADD     #40,R2          ;TO THE ERROR REPORT IN THE CALLING
 9612                                                                    ;ROUTINE.
 9613  052162  000112                            JMP     (R2)
 9614
 9615  052164                          16$:                              ;THE ERROR WAS NOT ANTICIPATED SO REPORT
 9616  052164  104002                  17$:      ERROR   2               ;THE INTEGER FAILURE HERE.
 9617  052166  000744                            BR      9$
 9618
 9619                                    ;FPS INCORRECT.
 9620  052170  010437  001254          20$:      MOV     R4,@#$TMP11     ;REPORT INCORRECT FPS.
 9621  052174  016137  000032  001256            MOV     32(R1),@#$TMP12
 9622  052202  104002                  21$:      ERROR   2
 9623  052204  000735                            BR      9$
```

```
 9624
 9625  052206  000000  000000  000000  MODFT0: .WORD   0,0,0,0
 9626  052214  000000
 9627
 9628  052216  000000  000000  000000  MODFT1: .WORD   0,0,0,0
 9629  052224  000000
 9630
 9631  052226  177777  177777  177777  MODP1:  .WORD   -1,-1,-1,-1
 9632  052234  177777
 9633
 9634  052236                          GGGDONE:
 9635  052236  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
 9636                                                                  ;SEE IF THE USER HAS EXPRESSED
 9637                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
 9638                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 9639                                                                  ;THE USER TYPED CONTROL G?).
 9640
 9641
 9642
 9643
 9644                                  ;;**********************************************************************
 9645                                  ;*TEST 55        MODD TEST
 9646                                  ;*
 9647                                  ;*THIS IS A TEST OF THE MODD INSTRUCTION. IT MAKES USE OF A SUBROUTINE
 9648                                  ;*TO SET UP THE ARGUMENTS, EXECUTE THE INSTRUCTION AND CHECK THE
 9649                                  ;*RESULTS.
 9650                                  ;*
 9651                                  ;;**********************************************************************
 9652  052240  000004                  TST55:  SCOPE
 9653
 9654                                  ;MODD WITH (FS C=AC=0)
 9655  052242                          HHH1:
 9656  052242  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9657  052244  004737  053746                  JSR     PC,@#MODDSUB
 9658  052250  000000  000000  000000  1$:     .WORD   0,0,0,0         ;AC
 9659  052256  000000
 9660  052260  000000  000000  000000  2$:     .WORD   0,0,0,0         ;FSRC
 9661  052266  000000
 9662  052270  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
 9663  052276  000000
 9664  052300  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
 9665  052306  000000
 9666  052310  000000  000000  000000  5$:     .WORD   0,0,0,0         ;ERROR FRACTIONAL RES.
 9667  052316  000000
 9668  052320  000000  000000  177777  6$:     .WORD   0,0,-1,-1       ;ERROR INTEGER RES.
 9669  052326  177777
 9670  052330  000200                  7$:     200                     ;FPS BEFORE EXECUTION.
 9671  052332  000204                          204                     ;FPS AFTER EXECUTION.
 9672  052334  104002                  8$:     ERROR   2
 9673  052336  000401                          BR      9$
 9674  052340  104002                          ERROR   2               ;ST 231 TO 142 INTO 143
 9675  052342                          9$:
 9676
 9677                                  ;MODD TEST WITH FSRC=0
 9678  052342                          HHH2:
 9679  052342  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
```

```
 9680  052344  004737  053746                    JSR     PC,@#MODDSUB
 9681  052350  012345  067012          1$:       .WORD   012345,67012    ;AC
 9682  052354  034567  012345                    .WORD   34567,012345
 9683  052360  000000  000000  000000  2$:       .WORD   0,0,0,0         ;FSRC
 9684  052366  000000
 9685  052370  000000  000000  000000  3$:       .WORD   0,0,0,0         ;FRACTIONAL RES.
 9686  052376  000000
 9687  052400  000000  000000  000000  4$:       .WORD   0,0,0,0         ;INTEGER RES.
 9688  052406  000000
 9689  052410  012345  067012          5$:       .WORD   012345,67012    ;ERROR FRACTIONAL RES.
 9690  052414  034567  012345                    .WORD   34567,012345
 9691  052420  177777  177777  177777  6$:       .WORD   -1,-1,-1,-1     ;ERROR INTEGER RES.
 9692  052426  177777
 9693  052430  000213                    7$:       213                     ;FPS BEFORE EXECUTION.
 9694  052432  000204                              204                     ;FPS AFTER EXECUTION.
 9695  052434  104002                    8$:       ERROR   2               ;STORE DOUBLE ZERO
 9696  052436  000401                              BR      9$
 9697  052440  104002                              ERROR   2               ;AC V 1 <= ZERO ST 143
 9698  052442                            9$:
 9699
 9700                                    ;MODD TEST WITH (AC=0)
 9701  052442                            HHH3:
 9702  052442  104414                              LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9703  052444  004737  053746                    JSR     PC,@#MODDSUB
 9704  052450  000000  000000  000000  1$:       .WORD   0,0,0,0         ;AC
 9705  052456  000000
 9706  052460  072727  127272          2$:       .WORD   72727,127272    ;FSRC
 9707  052464  072727  127272                    .WORD   72727,127272
 9708  052470  000000  000000  000000  3$:       .WORD   0,0,0,0         ;FRACTIONAL RES.
 9709  052476  000000
 9710  052500  000000  000000  000000  4$:       .WORD   0,0,0,0         ;INTEGER RES.
 9711  052506  000000
 9712  052510  177777  177777  177777  5$:       .WORD   -1,-1,-1,-1     ;ERROR FRACTIONAL RES.
 9713  052516  177777
 9714  052520  177777  177777  177777  6$:       .WORD   -1,-1,-1,-1     ;ERROR INTEGER RES.
 9715  052526  177777
 9716  052530  000213                    7$:       213                     ;FPS BEFORE EXECUTION.
 9717  052532  000204                              204                     ;FPS AFTER EXECUTION.
 9718  052534  104002                    8$:       ERROR   2
 9719  052536  000401                              BR      9$
 9720  052540  104002                              ERROR   2
 9721  052542                            9$:
 9722
 9723                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 57
 9724  052542                            HHH4:
 9725  052542  104414                              LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9726  052544  004737  053746                    JSR     PC,@#MODDSUB
 9727  052550  056252  125252          1$:       .WORD   56252,125252    ;AC
 9728  052554  125252  125250                    .WORD   125252,125250
 9729  052560  040300  000000  000000  2$:       .WORD   40300,0,0,0     ;FSRC
 9730  052566  000000
 9731  052570  000000  000000  000000  3$:       .WORD   0,0,0,0         ;FRACTIONAL RES.
 9732  052576  000000
 9733  052600  056377  177777  177777  4$:       .WORD   56377,-1,-1,-4  ;INTEGER RES.
 9734  052606  177774
 9735  052610  000000  000000          5$:       .WORD   0,0             ;ERROR FRACTIONAL RES.
```

```
 9736  052614  125252  125252                 .WORD   125252,125252
 9737  052620  056377  177777  177777  6$:    .WORD   56377,-1,-1,-1  ;ERROR INTEGER RES.
 9738  052626  177777
 9739  052630  000213                 7$:    213                     ;FPS BEFORE EXECUTION.
 9740  052632  000204                        204                     ;FPS AFTER EXECUTION.
 9741  052634  104002                 8$:    ERROR   2               ;ST 526 TO 134 INTO 135
 9742  052636  000401                        BR      9$
 9743  052640  104002                        ERROR   2
 9744  052642                         9$:
 9745
 9746                                 ;MODD TEST WITH EXPONENT OF THE RESULT = 79
 9747  052642                         HHH5:
 9748  052642  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9749  052644  004737  053746                 JSR     PC,@#MODDSUB
 9750  052650  140240  000000  000000  1$:    .WORD   140240,0,0,0    ;AC
 9751  052656  000000
 9752  052660  063714  146314          2$:    .WORD   63714,146314    ;FSRC
 9753  052664  133572  167737                 .WORD   133572,167737
 9754  052670  000000  000000  000000  3$:    .WORD   0,0,0,0         ;FRACTIONAL RES.
 9755  052676  000000
 9756  052700  163777  177777          4$:    .WORD   163777,-1       ;INTEGER RES.
 9757  052704  162531  125726                 .WORD   162531,125726
 9758  052710  177777  177777  177777  5$:    .WORD   -1,-1,-1,-1     ;ERROR FRACTIONAL RES.
 9759  052716  177777
 9760  052720  063777  177777          6$:    .WORD   63777,-1        ;ERROR INTEGER RES.
 9761  052724  162531  125726                 .WORD   162531,125726
 9762  052730  000210                 7$:    210                     ;FPS BEFORE EXECUTION.
 9763  052732  000204                        204                     ;FPS AFTER EXECUTION.
 9764  052734  104002                 8$:    ERROR   2
 9765  052736  000401                        BR      9$
 9766  052740  104002                        ERROR   2               ;ST 526 BAD SIGN
 9767  052742                         9$:
 9768
 9769                                 ;MODD TEST WITH EXPONENT OF THE RESULT = 57
 9770  052742                         HHH6:
 9771  052742  104414                        LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
 9772  052744  004737  053746                 JSR     PC,@#MODDSUB
 9773  052750  056200  000000  000000  1$:    .WORD   56200,0,0,1     ;AC
 9774  052756  000001
 9775  052760  040340  000000  000000  2$:    .WORD   40340,0,0,0     ;FSRC
 9776  052766  000000
 9777  052770  000000  000000  000000  3$:    .WORD   0,0,0,0         ;FRACTIONAL RES.
 9778  052776  000000
 9779  053000  056340  000000  000000  4$:    .WORD   56340,0,0,1     ;INTEGER RES.
 9780  053006  000001
 9781  053010  040000  000000  000000  5$:    .WORD   40000,0,0,0     ;ERROR FRACTIONAL RES.
 9782  053016  000000
 9783  053020  056340  000000  000000  6$:    .WORD   56340,0,0,1     ;ERROR INTEGER RES.
 9784  053026  000001
 9785  053030  000213                 7$:    213                     ;FPS BEFORE EXECUTION.
 9786  053032  000204                        204                     ;FPS AFTER EXECUTION.
 9787  053034  104002                 8$:    ERROR   2               ;CONSTANT BAD (NOT 56)
 9788                                                                 ;OR ST 525 TO 050 INTO 150
 9789  053036  000401                        BR      9$
 9790  053040  104002                        ERROR   2
 9791  053042                         9$:
```

```
 9792
 9793                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 56
 9794  053042                                HHH7:
 9795  053042  104414                               LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
 9796  053044  004737  053746                       JSR     PC,@#MODDSUB
 9797  053050  056000  000000  000000  1$:         .WORD   56000,0,0,1         ;AC
 9798  053056  000001
 9799  053060  040340  000000  000000  2$:         .WORD   40340,0,0,0         ;FSRC
 9800  053066  000000
 9801  053070  040100  000000  000000  3$:         .WORD   40100,0,0,0         ;FRACTIONAL RES.
 9802  053076  000000
 9803  053100  056140  000000  000000  4$:         .WORD   56140,0,0,1         ;INTEGER RES.
 9804  053106  000001
 9805  053110  000000  000000  000000  5$:         .WORD   0,0,0,0             ;ERROR FRACTIONAL RES.
 9806  053116  000000
 9807  053120  056140  000000  000000  6$:         .WORD   56140,0,0,1         ;ERROR INTEGER RES.
 9808  053126  000001
 9809  053130  000213                    7$:         213                         ;FPS BEFORE EXECUTION.
 9810  053132  000200                                200                         ;FPS AFTER EXECUTION.
 9811  053134  104002                    8$:         ERROR   2                   ;BAD CONSTANT (NOT 56) OR
 9812                                                                            ;ST 525 TO 150 INTO 050
 9813  053136  000401                                BR      9$
 9814  053140  104002                                ERROR   2
 9815  053142                            9$:
 9816
 9817                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 36
 9818  053142                                HHH8:
 9819  053142  104414                               LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
 9820  053144  004737  053746                       JSR     PC,@#MODDSUB
 9821  053150  051177  177777  177777  1$:         .WORD   51177,-1,-1,-1      ;AC
 9822  053156  177777
 9823  053160  040200  000000  000000  2$:         .WORD   40200,0,0,0         ;FSRC
 9824  053166  000000
 9825  053170  040177  177760  000000  3$:         .WORD   40177,-20,0,0       ;FRACTIONAL RES.
 9826  053176  000000
 9827  053200  051177  177777  177760  4$:         .WORD   51177,-1,-20,0      ;INTEGER RES.
 9828  053206  000000
 9829  053210  177777  177777  177777  5$:         .WORD   -1,-1,-1,-1         ;ERROR FRACTIONAL RES.
 9830  053216  177777
 9831  053220  177777  177777  177777  6$:         .WORD   -1,-1,-1,-1         ;ERROR INTEGER RES.
 9832  053226  177777
 9833  053230  000217                    7$:         217                         ;FPS BEFORE EXECUTION.
 9834  053232  000200                                200                         ;FPS AFTER EXECUTION.
 9835  053234  104002                    8$:         ERROR   2
 9836  053236  000401                                BR      9$
 9837  053240  104002                                ERROR   2
 9838  053242                            9$:
 9839
 9840                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 30
 9841  053242                                HHH9:
 9842  053242  104414                               LPERR                       ;SET UP THE LOOP ON ERROR ADDRESS.
 9843  053244  004737  053746                       JSR     PC,@#MODDSUB
 9844  053250  040200  000000  000000  1$:         .WORD   40200,0,0,0         ;AC
 9845  053256  000000
 9846  053260  047577  177777            2$:         .WORD   47577,-1            ;FSRC
 9847  053264  176000  000001                      .WORD   176000,1
```

```
9848  053270  031600  000000  000000  3$:     .WORD   31600,0,0,0     ;FRACTIONAL RES.
9849  053276  000000
9850  053300  047577  177777          4$:     .WORD   47577,-1        ;INTEGER RES.
9851  053304  176000  000000                  .WORD   176000,0
9852  053310  000000  000000  000000  5$:     .WORD   0,0,0,0         ;ERROR FRACTIONAL RES.
9853  053316  000000
9854  053320  047577  177777  177777  6$:     .WORD   47577,-1,-1,-1  ;ERROR INTEGER RES.
9855  053326  177777
9856  053330  000200                  7$:     200                     ;FPS BEFORE EXECUTION.
9857
9858  053332  000200                          200                     ;FPS AFTER EXECUTION.
9859  053334  104002                  8$:     ERROR   2               ;(NORMALIZE) ST 532 TO 122
9860                                                                  ;INTO NORM.
9861  053336  000401                          BR      9$
9862  053340  104002                          ERROR   2               ;AC V 1 <= X14
9863                                                                  ;OR ST 733 TO 156 INTO 157.
9864  053342                          9$:
9865
9866                                  ;MODD TEST WITH EXPONENT OF THE RESULT = 31
9867  053342                          HHH10:
9868  053342  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9869  053344  004737  053746                  JSR     PC,@#MODDSUB
9870  053350  047777  177777          1$:     .WORD   47777,-1        ;AC
9871  053354  177000  000000                  .WORD   177000,0
9872  053360  040200  000000  000000  2$:     .WORD   40200,0,0,0     ;FSRC
9873  053366  000000
9874  053370  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
9875  053376  000000
9876  053400  047777  177777          4$:     .WORD   47777,-1        ;INTEGER RES.
9877  053404  177000  000000                  .WORD   177000,0
9878  053410  000000  000000  177000  5$:     .WORD   0,0,177000,0    ;ERROR FRACTIONAL RES.
9879  053416  000000
9880  053420  177777  177777  177777  6$:     .WORD   -1,-1,-1,-1     ;ERROR INTEGER RES.
9881  053426  177777
9882  053430  000213                  7$:     213                     ;FPS BEFORE EXECUTION.
9883  053432  000204                          204                     ;FPS AFTER EXECUTION.
9884  053434  104002                  8$:     ERROR   2               ;(BUT FD) STORE X10
9885  053436  000401                          BR      9$
9886  053440  104002                          ERROR   2
9887  053442                          9$:
9888
9889                                  ;MODD TEST WITH EXPONENT OF THE RESULT = 0
9890  053442                          HHH11:
9891  053442  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9892  053444  004737  053746                  JSR     PC,@#MODDSUB
9893  053450  040200  000000  000000  1$:     .WORD   40200,0,0,0     ;AC
9894  053456  000000
9895  053460  040177  072727          2$:     .WORD   40177,72727     ;FSRC
9896  053464  127272  072727                  .WORD   127272,72727
9897  053470  040177  072727          3$:     .WORD   40177,72727     ;FRACTIONAL RES.
9898  053474  127272  072727                  .WORD   127272,72727
9899  053500  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
9900  053506  000000
9901  053510  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR FRACTIONAL RES.
9902  053516  177777
9903  053520  000000  000000  177777  6$:     .WORD   0,0,-1,-1       ;ERROR INTEGER RES.
```

```
9904  053526  177777
9905  053530  000200                  7$:     200                     ;FPS BEFORE EXECUTION.
9906  053532  000200                          200                     ;FPS AFTER EXECUTION.
9907  053534  104002                  8$:     ERROR   2
9908  053536  000401                          BR      9$
9909  053540  104002                          ERROR   2               ;ST 246 TO 126 INTO 127 (BUT FD)
9910  053542                          9$:
9911
9912                                  ;MODD TEST WITH EXPONENT OF THE RESULT = -115
9913  053542                          HHH12:
9914  053542  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9915  053544  004737  053746                  JSR     PC,@#MODDSUB
9916  053550  003377  177777          1$:     .WORD   3377,-1         ;AC
9917  053554  177777  052525                  .WORD   -1,52525
9918  053560  040200  000000  000000  2$:     .WORD   40200,0,0,0     ;FSRC
9919  053566  000000
9920  053570  003377  177777          3$:     .WORD   3377,-1         ;FRACTIONAL RES.
9921  053574  177777  052525                  .WORD   -1,52525
9922  053600  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
9923  053606  000000
9924  053610  177777  177777  177777  5$:     .WORD   -1,-1,-1,-1     ;ERROR FRACTIONAL RES.
9925  053616  177777
9926  053620  000000  000000  177777  6$:     .WORD   0,0,-1,-1       ;ERROR INTEGER RES.
9927  053626  177777
9928  053630  000200                  7$:     200                     ;FPS BEFORE EXECUTION.
9929  053632  000200                          200                     ;FPS AFTER EXECUTION.
9930  053634  104002                  8$:     ERROR   2
9931  053636  000401                          BR      9$
9932  053640  104002                          ERROR   2               ;ST 446 TO 126 INTO 127 (BUT FD)
9933  053642                          9$:
9934
9935                                  ;MODD TEST WITH EXPONENT OF THE RESULT = -63, IN ROUND MODE.
9936  053642                          HHH13:
9937  053642  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
9938  053644  004737  053746                  JSR     PC,@#MODDSUB
9939  053650  040300  000000  000000  1$:     .WORD   40300,0,0,0     ;AC
9940  053656  000000
9941  053660  020200  000000  000000  2$:     .WORD   20200,0,0,1     ;FSRC
9942  053666  000001
9943  053670  020300  000000  000000  3$:     .WORD   20300,0,0,2     ;FRACTIONAL RES.
9944  053676  000002
9945  053700  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
9946  053706  000000
9947  053710  000000  000000  177777  5$:     .WORD   0,0,-1,-1       ;ERROR FRACTIONAL RES.
9948  053716  177777
9949  053720  177777  177777  177777  6$:     .WORD   -1,-1,-1,-1     ;ERROR INTEGER RES.
9950  053726  177777
9951  053730  000200                  7$:     200                     ;FPS BEFORE EXECUTION.
9952  053732  000200                          200                     ;FPS AFTER EXECUTION.
9953  053734  104002                  8$:     ERROR   2               ;ST 127 INTO RND/TR
9954  053736  000401                          BR      9$
9955  053740  104002                          ERROR   2
9956  053742  000137  054344          9$:     JMP     @#HHHDONE       ;GO TO THE NEXT TEST.
9957
9958                                  ;THIS SUBROUTINE, MODDSUB, IS CALLED TO SETUP THE
9959                                  ;OPERANDS, EXECUTE THE MODD INSTRUCTION AND CHECK THE RESULTS.
```

```
 9960                                   ;IT IS CALLED THUS:
 9961                                   ;
 9962                                   ;       ACARG:  .WORD   X,X,X,X         ;AC OPERAND
 9963                                   ;       FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
 9964                                   ;       FRES:   .WORD   X,X,X,X         ;FRACTIONAL RESULT
 9965                                   ;       INTRES: .WORD   X,X,X,X         ;INTEGER RESULT
 9966                                   ;       ERFRES: .WORD   X,X,X,X         ;ERROR FRACTION RESULT
 9967                                   ;       ERINTRES:.WORD  X,X,X,X         ;ERROR INTEGER RESULT
 9968                                   ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
 9969                                   ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
 9970                                   ;       ERR1:   ERROR   X               ;FRACTION ERROR
 9971                                   ;               BR      CONT
 9972                                   ;       ERR2:   ERROR   X               ;INTEGER ERROR
 9973                                   ;       CONT:                           ;RETURN ADDRESS
 9974                                   ;
 9975                                   ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODD
 9976                                   ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
 9977                                   ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
 9978                                   ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
 9979                                   ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
 9980                                   ;THEN MODDSUB WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
 9981                                   ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
 9982                                   ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
 9983                                   ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
 9984                                   ;FAILURE MATCHES THE TRUE RESULT THEN  MODDSUB PASSES CONTROL TO THE
 9985                                   ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
 9986                                   ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
 9987                                   ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
 9988
 9989                                   ;IF A MATCH IS MADE HOWEVER, MODDSUB WILL RETURN CONTROL TO THE ERROR
 9990                                   ;CALL AT ERR2.
 9991
 9992  053746  012601                   MODDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
 9993  053750  012700  000200                   MOV     #200,R0         ;SET FD MODE.
 9994  053754  170100                   LDFPS   R0
 9995  053756  010100                   MOV     R1,R0           ;SET UP AC0
 9996  053760  172410                   LDD     (R0),AC0
 9997  053762  012700  052226           MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
 9998  053766  172510                   LDD     (R0),AC1
 9999  053770  016100  000060           MOV     60(R1),R0       ;SET UP THE FPS.
10000  053774  170100                   LDFPS   R0
10001  053776  012737  054012  001236   MOV     #1$,@#$TMP2
10002  054004  010100                   MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
10003  054006  062700  000010           ADD     #10,R0
10004
10005  054012  171410                   1$:     MODD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
10006
10007  054014  170204                   STFPS   R4              ;GET THE FPS.
10008  054016  012700  000200           MOV     #200,R0         ;SET FD MODE.
10009  054022  170100                   LDFPS   R0
10010  054024  012700  054324           MOV     #MODDT0,R0      ;GET THE FRACTIONAL RESULT.
10011  054030  174010                   STD     AC0,(R0)
10012  054032  012700  054334           MOV     #MODDT1,R0      ;GET THE INTEGER RESULT.
10013  054036  174110                   STD     AC1,(R0)
10014
10015  054040  010102                   MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
```

```
10016  054042  010237  001240              MOV     R2,@#$TMP3
10017  054046  062702  000010              ADD     #10,R2
10018  054052  010237  001242              MOV     R2,@#$TMP4
10019  054056  062702  000010              ADD     #10,R2
10020  054062  010237  001244              MOV     R2,@#$TMP5
10021  054066  062702  000010              ADD     #10,R2
10022  054072  010237  001246              MOV     R2,@#$TMP6
10023  054076  012737  054324  001250      MOV     #MODDT0,@#$TMP7
10024  054104  012737  054334  001252      MOV     #MODDT1,@#$TMP10
10025  054112  016137  000062  001256      MOV     62(R1),@#$TMP12
10026  054120  010437  001254              MOV     R4,@#$TMP11
10027
10028  054124  012702  054324              MOV     #MODDT0,R2      ;CHECK THE FRACTIONAL RESULT.
10029  054130  010103                      MOV     R1,R3
10030  054132  062703  000020              ADD     #20,R3
10031  054136  012705  000004              MOV     #4,R5
10032  054142  022223          2$:         CMP     (R2)+,(R3)+
10033  054144  001020                      BNE     10$             ;BRANCH IF INCORRECT.
10034  054146  077503                      SOB     R5,2$
10035
10036  054150  012702  054334              MOV     #MODDT1,R2      ;CHECK THE INTEGER RESULT.
10037  054154  010103                      MOV     R1,R3
10038  054156  062703  000030              ADD     #30,R3
10039  054162  012705  000004              MOV     #4,R5
10040  054166  022223          3$:         CMP     (R2)+,(R3)+
10041  054170  001026                      BNE     15$             ;BRANCH IF INCORRECT.
10042  054172  077503                      SOB     R5,3$
10043
10044
10045  054174  026104  000062              CMP     62(R1),R4       ;CHECK THE FPS.
10046  054200  001042                      BNE     20$             ;BRANCH IF INCORRECT.
10047
10048  054202  000161  000072          9$:     JMP     72(R1)          ;RETURN.
10049
10050                                  ;FRACTIONAL ERROR.
10051  054206  012702  054324          10$:    MOV     #MODDT0,R2      ;WAS THE FRACTIONAL ERROR ANTICIPATED?
10052  054212  010103                          MOV     R1,R3
10053  054214  062703  000040                  ADD     #40,R3
10054  054220  012705  000004                  MOV     #4,R5
10055  054224  022223          50$:    CMP     (R2)+,(R3)+
10056  054226  001005                          BNE     11$             ;BRANCH IF NOT ANTICIPATED.
10057  054230  077503                          SOB     R5,50$
10058  054232  010102                          MOV     R1,R2           ;THE ERROR WAS ANTICIPATED SO
10059  054234  062702  000064                  ADD     #64,R2          ;RETURN TO THE ERROR REPORT AT THE
10060                                                                  ;CALLING ROUTINE.
10061  054240  000112                          JMP     (R2)
10062
10063  054242                          11$:                            ;THE ERROR WAS NOT ANTICIPATED SO
10064  054242  104002                  12$:    ERROR   2               ;REPORT THE INCORRECT FRACTION HERE.
10065  054244  000756                          BR      9$
10066
10067                                  ;INTEGER ERROR.
10068  054246  012702  054334          15$:    MOV     #MODDT1,R2      ;WAS THE INTEGER ERROR ANTICIPATED?
10069  054252  010103                          MOV     R1,R3
10070  054254  062703  000050                  ADD     #50,R3
10071  054260  012705  000004                  MOV     #4,R5
```

```
10072  054264  022223                  60$:    CMP     (R2)+,(R3)+
10073  054266  001005                          BNE     17$                     ;BRANCH IF NOT ANTICIPATED.
10074  054270  077503                          SOB     R5,60$
10075  054272  010102                          MOV     R1,R2                   ;THE ERROR WAS ANTICIAPTED SO RETURN
10076  054274  062702  000070                  ADD     #70,R2                  ;TO THE ERROR REPORT IN THE CALLING
10077                                                                          ;ROUTINE.
10078  054300  000112                          JMP     (R2)
10079
10080  054302                          16$:                                    ;THE ERROR WAS NOT ANTICIPATED SO REPORT
10081  054302  104002                  17$:    ERROR   2                       ;THE INTEGER FAILURE HERE.
10082  054304  000736                          BR      9$
10083
10084                                  ;FPS INCORRECT.
10085  054306  010437  001254          20$:    MOV     R4,@#$TMP11             ;REPORT INCORRECT FPS.
10086  054312  016137  000062  001256          MOV     62(R1),@#$TMP12
10087  054320  104002                  21$:    ERROR   2
10088  054322  000727                          BR      9$
10089
10090  054324  000000  000000  000000  MODDT0: .WORD   0,0,0,0
10091  054332  000000
10092
10093  054334  000000  000000  000000  MODDT1: .WORD   0,0,0,0
10094  054342  000000
10095
10096  054344                          HHHDONE:
10097  054344  104413                          RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
10098                                                                          ;SEE IF THE USER HAS EXPRESSED
10099                                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
10100                                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
10101                                                                          ;THE USER TYPED CONTROL G?).
10102
10103
10104
10105
10106                                  ;;*****************************************************************
10107                                  ;*TEST 56        UNDER\OVER FLOW, USING MODF WITH TRAPS DISABLED, TEST
10108                                  ;*
10109                                  ;*THIS IS A TEST OF THE MODF OVERFLOW AND UNDERFLOW CONDITIONS. IT MAKES
10110                                  ;*USE OF A SUBROUTINE TO SETUP THE OPERANDS, EXECUTE THE MODF INSTRUCTION
10111                                  ;*AND CHECK THE RESULTS. TRAPS ARE DISABLED DURING THIS TEST.
10112                                  ;*
10113                                  ;;*****************************************************************
10114  054346  000004                  TST56:  SCOPE
10115
10116                                  ;UNDERFLOW TEST, WITH EXPONENT OF THE RESULT = -129, FIU = 1, FID = 1
10117  054350                          MMM1:
10118  054350  104414                          LPERR                           ;SET UP THE LOOP ON ERROR ADDRESS.
10119  054352  004767  000320                  JSR     PC,MODFOV
10120  054356  020123  045676          1$:     .WORD   20123,45676             ;AC
10121  054362  020200  000000          2$:     .WORD   20200,0                 ;FSRC
10122  054366  000123  045676          3$:     .WORD   123,45676               ;FRACTIONAL RES.
10123  054372  000000  000000          4$:     .WORD   0,0                     ;INTEGER RES.
10124  054376  177777  177777          5$:     .WORD   -1,-1                   ;ERROR FRACTIONAL RES.
10125  054402  177777  177777          6$:     .WORD   -1,-1                   ;ERROR INTEGER RES.
10126  054406  042000                  7$:     42000                           ;FPS BEFORE EXECUTION.
10127  054410  142004                          142004                          ;FPS AFTER EXECUTION.
```

```
10128  054412  000012                          12                      ;FEC
10129  054414  104002                  8$:     ERROR   2               ;FEC INCORRECT, UNDERFLOW.
10130  054416  000401                          BR      9$
10131  054420  104002                          ERROR   2               ;AC V 1 (2,3) <= ZERO, ST 126.
10132  054422                          9$:
10133
10134                                  ;UNDERFLOW EXP OF RESULT = -193, FIU = 0, FID = 1
10135  054422                          MMM2:
10136  054422  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
10137  054424  004737  054676                  JSR     PC,@#MODFOV
10138  054430  010200  000000          1$:     .WORD   10200,0         ;AC
10139  054434  010000  000000          2$:     .WORD   10000,0         ;FSRC
10140  054440  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
10141  054444  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
10142  054450  177777  177777          5$.     .WORD   -1,-1           ;ERROR FRACTIONAL RES.
10143  054454  177777  177777          6$:     .WORD   -1,-1           ;ERROR INTEGER RES.
10144  054460  005013                  7$:     5013                    ;FPS BEFORE EXECUTION.
10145  054462  005004                          5004                    ;FPS AFTER EXECUTION.
10146  054464  000012                          12                      ;FEC
10147  054466  000240                  8$:     NOP
10148  054470  000401                          BR      9$
10149  054472  104002                          ERROR   2
10150  054474                          9$:
10151
10152                                  ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, FIV = 1, FID   1
10153  054474                          MMM3:
10154  054474  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
10155  054476  004737  054676                  JSR     PC,@#MODFOV
10156  054502  060052  125252          1$:     .WORD   60052,125252    ;AC
10157  054506  060200  000000          2$:     .WORD   60200,0         ;FSRC
10158  054512  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
10159  054516  000052  125252          4$:     .WORD   52,125252       ;INTEGER RES.
10160  054522  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
10161  054526  000000  000000          6$:     .WORD   0,0             ;ERROR INTEGER RES.
10162  054532  041000                  7$:     41000                   ;FPS BEFORE EXECUTION.
10163  054534  141006                          141006                  ;FPS AFTER EXECUTION.
10164  054536  000010                          10                      ;FEC
10165  054540  104002                  8$:     ERROR   2               ;BAD FEC ON OVERFLOW.
10166  054542  000401                          BR      9$
10167  054544  104002                          ERROR   2               ;ST 520 TO STORE ZERO TWICE
10168                                                                  ;INTO 162
10169  054546                          9$:
10170
10171                                  ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 130, FIV = 0, FID = 1
10172  054546                          MMM4:
10173  054546  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
10174  054550  004737  054676                  JSR     PC,@#MODFOV
10175  054554  060345  067654          1$:     .WORD   60345,67654     ;AC
10176  054560  060200  000000          2$:     .WORD   60200,0         ;FSRC
10177  054564  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
10178  054570  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
10179  054574  000000  000000          5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
10180  054600  000345  067654          6$:     .WORD   345,67654       ;ERROR INTEGER RES.
10181  054604  006011                  7$:     6011                    ;FPS BEFORE EXECUTION.
10182  054606  006006                          6006                    ;FPS AFTER EXECUTION.
10183  054610  000010                          10                      ;FEC
```

```
                                        T56     UNDER\OVER FLOW, USING MODF WITH TRAPS DISABLED, TEST

10184  054612  000240                   8$:     NOP
10185  054614  000401                           BR      9$
10186  054616  104002                           ERROR   2               ;ST 520 TO 162 INTO STORE ZERO TWICE.
10187  054620                           9$:
10188
10189                                   ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, RESULT NEGATIVE
10190                                   ;AND FIV = 1, FID = 1
10191  054620                           MMM5:
10192  054620  104414                           LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
10193  054622  004737  054676                   JSR     PC,@#MODFOV
10194  054626  160252  125252           1$:     .WORD   160252,125252   ;AC
10195  054632  060000  000000           2$:     .WORD   60000,0         ;FSRC
10196  054636  000000  000000           3$:     .WORD   0,0             ;FRACTIONAL RES.
10197  054642  100052  125252           4$:     .WORD   100052,125252   ;INTEGER RES.
10198  054646  000000  000000           5$:     .WORD   0,0             ;ERROR FRACTIONAL RES.
10199  054652  000052  125252           6$:     .WORD   52,125252       ;ERROR INTEGER RES.
10200  054656  041000                   7$:     41000                   ;FPS BEFORE EXECUTION.
10201  054660  141006                           141006                  ;FPS AFTER EXECUTION.
10202  054662  000010                           10                      ;FEC
10203  054664  104002                   8$:     ERROR   2
10204  054666  000401                           BR      9$
10205  054670  104002                           ERROR   2               ;ST 517, BAD SIGN.
10206  054672  000137  055272           9$:     JMP     @#MMMMDONE      ;GO TO THE NEXT TEST.
10207
10208                                   ;THIS SUBROUTINE, MODFOV, IS CALLED TO SETUP THE
10209                                   ;OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK THE RESULTS.
10210                                   ;IT IS CALLED THUS:
10211                                   ;
10212                                   ;               ACARG:  .WORD   X,X             ;AC OPERAND
10213                                   ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
10214                                   ;               FRES:   .WORD   X,X             ;FRACTIONAL RESULT
10215                                   ;               INTRES: .WORD   X,X             ;INTEGER RESULT
10216                                   ;               ERFRES: .WORD   X,X             ;ERROR FRACTION RESULT
10217                                   ;               ERINTRES:.WORD  X,X             ;ERROR INTEGER RESULT
10218                                   ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
10219                                   ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
10220                                   ;               FEC:    .WORD   X               ;FEC
10221                                   ;               ERR1:   ERROR   X               ;FEC ERROR
10222                                   ;                       BR      CONT
10223                                   ;               ERR2:   ERROR   X               ;INTEGER ERROR
10224                                   ;               CONT:                           ;RETURN ADDRESS
10225                                   ;
10226                                   ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODF
10227                                   ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
10228                                   ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
10229                                   ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
10230                                   ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
10231                                   ;THEN MODFOV WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
10232                                   ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
10233                                   ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
10234                                   ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
10235                                   ;FAILURE MATCHES THE TRUE RESULT THEN  MODFOV PASSES CONTROL TO THE
10236                                   ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
10237                                   ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
10238                                   ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
10239                                   ;IF A MATCH IS MADE HOWEVER, MODFOV WILL RETURN CONTROL TO THE ERROR
```

```
10240                                   ;CALL AT ERR2.
10241
10242  054676  012601                   MODF0\  MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
10243  054700  012700  000200                   MOV     #200,R0         ;SET FD MODE.
10244  054704  170100                           LDFPS   R0
10245  054706  010100                           MOV     R1,R0           ;SET UP AC0
10246  054710  172410                           LDD     (R0),AC0
10247  054712  012700  052226                   MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
10248  054716  172510                           LDD     (R0),AC1
10249  054720  016100  000030                   MOV     30(R1),R0       ;SET UP THE FPS.
10250  054724  170100                           LDFPS   R0
10251  054726  012737  054742  001236           MOV     #1$,@#$TMP2
10252  054734  010100                           MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
10253  054736  062700  000004                   ADD     #4,R0
10254
10255  054742  171410                   1$:     MODF    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
10256
10257  054744  170204                           STFPS   R4              ;GET THE FPS.
10258  054746  170305                           STST    R5              ;GET FEC.
10259  054750  012700  000200                   MOV     #200,R0         ;SET FD MODE.
10260  054754  170100                           LDFPS   R0
10261  054756  012700  055252                   MOV     #MODFD0,R0      ;GET THE FRACTIONAL RESULT.
10262  054762  174010                           STD     AC0,(R0)
10263  054764  012700  055262                   MOV     #MODFD1,R0      ;GET THE INTEGER RESULT.
10264  054770  174110                           STD     AC1,(R0)
10265
10266  054772  010102                           MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
10267  054774  010237  001240                   MOV     R2,@#$TMP3
10268  055000  062702  000004                   ADD     #4,R2
10269  055004  010237  001242                   MOV     R2,@#$TMP4
10270  055010  062702  000004                   ADD     #4,R2
10271  055014  010237  001244                   MOV     R2,@#$TMP5
10272  055020  062702  000004                   ADD     #4,R2
10273  055024  010237  001246                   MOV     R2,@#$TMP6
10274  055030  012737  055252  001250           MOV     #MODFD0,@#$TMP7
10275  055036  012737  055262  001252           MOV     #MODFD1,@#$TMP10
10276  055044  010437  001254                   MOV     R4,@#$TMP11
10277  055050  016137  000032  001256           MOV     32(R1),@#$TMP12
10278  055056  010537  001260                   MOV     R5,@#$TMP13
10279  055062  016137  000034  001262           MOV     34(R1),@#$TMP14
10280
10281  055070  012702  055252                   MOV     #MODFD0,R2      ;CHECK THE FRACTIONAL RESULT.
10282  055074  026112  000010                   CMP     10(R1),(R2)
10283  055100  001025                           BNE'    10$             ;BRANCH IF INCORRECT.
10284  055102  026162  000012  000002           CMP     12(R1),2(R2)
10285  055110  001021                           BNE     10$
10286
10287  055112  012702  055262                   MOV     #MODFD1,R2      ;CHECK THE INTEGER RESULT.
10288  055116  026112  000014                   CMP     14(R1),(R2)
10289  055122  001016                           BNE     15$             ;BRANCH IF INCORRECT.
10290  055124  026162  000016  000002           CMP     16(R1),2(R2)
10291  055132  001012                           BNE     15$
10292
10293  055134  026104  000032                   CMP     32(R1),R4       ;CHECK THE FPS.
10294  055140  001024                           BNE     20$             ;BRANCH IF INCORRECT.
10295
```

```
                                T56     UNDER\OVER FLOW, USING MODF WITH TRAPS DISABLED, TEST                    SEQ 0204

10296  055142  026105  000034            CMP     34(R1),R5       ;CHECK THE FEC.
10297  055146  001030                    BNE     25$             ;BRANCH IF INCORRECT.
10298
10299  055150  000161  000044    9$:     JMP     44(R1)          ;RETURN.
10300
10301                                   ;FRACTIONAL ERROR.
10302  055154                    10$:                            ;THE ERROR WAS NOT ANTICIPATED SO
10303  055154  104002            12$:    ERROR   2               ;REPORT THE INCORRECT FRACTION HERE.
10304  055156  000774                    BR      9$
10305
10306                                   ;INTEGER ERROR.
10307  055160  026112  000024    15$:    CMP     24(R1),(R2)     ;WAS THIS ERROR ANTICIPATED?
10308  055164  001010                    BNE     16$             ;BRANCH IF NOT.
10309  055166  026162  000026  000002      CMP     26(R1),2(R2)
10310  055174  001004                    BNE     16$
10311  055176  010102                    MOV     R1,R2           ;THE ERROR WAS ANTICIAPTED SO RETURN
10312  055200  062702  000042            ADD     #42,R2          ;TO THE ERROR REPORT IN THE CALLING
10313                                                            ;ROUTINE.
10314  055204  000112                    JMP     (R2)
10315
10316  055206                    16$:                            ;THE ERROR WAS NOT ANTICIPATED SO REPORT
10317  055206  104002            17$:    ERROR   2               ;THE INTEGER FAILURE HERE.
10318  055210  000757                    BR      9$
10319
10320                                   ;FPS INCORRECT.
10321  055212  010437  001254    20$:    MOV     R4,@#$TMP11     ;REPORT INCORRECT FPS.
10322  055216  016137  000032  001256      MOV     32(R1),@#$TMP12
10323  055224  104002            21$:    ERROR   2
10324  055226  000750                    BR      9$
10325
10326                                   ;REPORT FEC ERROR.
10327  055230  010537  001260    25$:    MOV     R5,@#$TMP13
10328  055234  016137  000034  001262      MOV     34(R1),@#$TMP14
10329  055242  010102                    MOV     R1,R2
10330  055244  062702  000036            ADD     #36,R2
10331  055250  000112                    JMP     (R2)
10332
10333  055252  000000  000000  000000  MODFD0: .WORD   0,0,0,0
10334  055260  000000
10335
10336  055262  000000  000000  000000  MODFD1: .WORD   0,0,0,0
10337  055270  000000
10338
10339  055272                    MMMDONE:
10340  055272  104413                    RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
10341                                                            ;SEE IF THE USER HAS EXPRESSED
10342                                                            ;THE DESIRE TO CHANGE THE SOFTWARE
10343                                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
10344                                                            ;THE USER TYPED CONTROL G?).
10345
10346
10347
10348
10349                                   ;;*****************************************************************
10350                                   ;*TEST 57        UNDER\OVER FLOW, USING MODD WITH TRAPS DISABLED, TEST
10351                                   ;*
```

```
10352                                  ;*THIS IS A TEST OF THE MODD INSTRUCTION'S OVER FLOW AND UNDER FLOW
10353                                  ;*CONDITIONS. A SUBROUTINE IS USED TO SET UP THE OPERANDS, EXECUTE THE
10354                                  ;*MODD INSTUCTION AND CHECK THE RESULTS.
10355                                  ;*
10356                                  ;;*****************************************************************
10357  055274  000004                 TST57:  SCOPE
10358
10359                                  ;UNDERFLOW TEST WITH EXPONENT OF THE RESULT = -129, FIU = 1, FID = 1
10360  055276                         NNN1:
10361  055276  104414                         LPERR                  ;SET UP THE LOOP ON ERROR ADDRESS.
10362  055300  004737  055712                JSR     PC,@#MODDOV
10363  055304  020252  125252        1$:     .WORD   20252,125252   ;AC
10364  055310  125252  125252                .WORD   125252,125252
10365  055314  020100  000000  000000  2$:     .WORD   20100,0,0,0    ;FSRC
10366  055322  000000
10367  055324  000177  177777  177777  3$:     .WORD   177,-1,-1,-1   ;FRACTIONAL RES.
10368  055332  177777
10369  055334  000000  000000  000000  4$:     .WORD   0,0,0,0        ;INTEGER RES.
10370  055342  000000
10371  055344  020252  125252          5$:     .WORD   20252,125252   ;ERROR FRACTIONAL RES.
10372  055350  125252  125252                .WORD   125252,125252
10373  055354  000000  000000  177777  6$:     .WORD   0,0,-1,-1      ;ERROR INTEGER RES.
10374  055362  177777
10375  055364  042200                 7$:     42200                  ;FPS BEFORE EXECUTION.
10376  055366  142204                         142204                 ;FPS AFTER EXECUTION.
10377  055370  000012                         12                     ;FEC
10378  055372  104002                 8$:     ERROR   2              ;FEC INCORRECT ON UNDERFLOW.
10379  055374  000401                         BR      9$
10380  055376  104002                         ERROR   2              ;ST 155 (BUT FD)
10381  055400                         9$:
10382
10383                                  ;UNDERFLOW TEST WITH EXPONENT OF THE RESULT = -193, FIU = 0, FID = 1
10384  055400                         NNN2:
10385  055400  104414                         LPERR                  ;SET UP THE LOOP ON ERROR ADDRESS.
10386  055402  004737  055712                JSR     PC,@#MODDOV
10387  055406  010000  000000          1$:     .WORD   10000,0        ;AC
10388  055412  123456  000000                .WORD   123456,0
10389  055416  010200  000000  000000  2$:     .WORD   10200,0,0,0    ;FSRC
10390  055424  000000
10391  055426  000000  000000  000000  3$:     .WORD   0,0,0,0        ;FRACTIONAL RES.
10392  055434  000000
10393  055436  000000  000000  000000  4$:     .WORD   0,0,0,0        ;INTEGER RES.
10394  055444  000000
10395  055446  000000  000000  000000  5$:     .WORD   0,0,0,0        ;ERROR FRACTIONAL RES.
10396  055454  000000
10397  055456  000000  000000          6$:     .WORD   0,0            ;ERROR INTEGER RES.
10398  055462  123456  000000                .WORD   123456,0
10399  055466  005213                 7$:     5213                   ;FPS BEFORE EXECUTION.
10400  055470  005204                         5204                   ;FPS AFTER EXECUTION.
10401  055472  000012                         12
10402  055474  000240                 8$:     NOP
10403  055476  000401                         BR      9$
10404  055500  104002                         ERROR   2              ;ST 047 (BUT FD).
10405  055502                         9$:
10406
10407                                  ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, FIV = 1, FID = 1
```

```
10408  055502                          NNN3:
10409  055502  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
10410  055504  004737  055712                  JSR     PC,@#MODDOV
10411  055510  060252  125252          1$:     .WORD   60252,125252    ;AC
10412  055514  125252  125252                  .WORD   125252,125252
10413  055520  060100  000000  000000  2$:     .WORD   60100,0,0,0     ;FSRC
10414  055526  000000
10415  055530  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
10416  055536  000000
10417  055540  000177  177777  177777  4$:     .WORD   177,-1,-1,-1    ;INTEGER RES.
10418  055546  177777
10419  055550  000000  000000  000000  5$:     .WORD   0,0,0,0         ;ERROR FRACTIONAL RES.
10420  055556  000000
10421  055560  000177  177777          6$:     .WORD   177,-1          ;ERROR INTEGER RES.
10422  055564  125252  125252                  .WORD   125252,125252
10423  055570  041200                  7$:     41200                   ;FPS BEFORE EXECUTION.
10424  055572  141206                          141206                  ;FPS AFTER EXECUTION.
10425  055574  000010                          10                      ;FEC
10426  055576  104002                  8$:     ERROR   2               ;FEC BAD ON OVERFLOW.
10427  055600  000401                          BR      9$
10428  055602  104002                          ERROR   2               ;ST 520 TO 162 INTO 163 (BUT FD).
10429  055604                          9$:
10430
10431                                  ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 130, FIV - 0, FID = 1
10432  055604                          NNN4:
10433  055604  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
10434  055606  004737  055712                  JSR     PC,@#MODDOV
10435  055612  060200  000000          1$:     .WORD   60200,0         ;AC
10436  055616  125252  000000                  .WORD   125252,0
10437  055622  060200  000000  000000  2$:     .WORD   60200,0,0,0     ;FSRC
10438  055630  000000
10439  055632  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
10440  055640  000000
10441  055642  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
10442  055650  000000
10443  055652  000000  000000  000000  5$:     .WORD   0,0,0,0         ;ERROR FRACTIONAL RES.
10444  055660  000000
10445  055662  000400  000000          6$:     .WORD   400,0           ;ERROR INTEGER RES.
10446  055666  125252  000000                  .WORD   125252,0
10447  055672  006211                  7$:     6211                    ;FPS BEFORE EXECUTION.
10448  055674  006206                          6206                    ;FPS AFTER EXECUTION.
10449  055676  000010                          10                      ;FEC
10450  055700  000240                  8$:     NOP
10451  055702  000401                          BR      9$
10452  055704  104002                          ERROR   2               ;ST 520 TO 162 INTO STORE ZERO TWICE.
10453  055706  000137  056320          9$:     JMP     @#NNNDONE       ;GO TO NEXT TEST.
10454
10455                                  ;THIS SUBROUTINE, MODDOV, IS CALLED TO SETUP THE
10456                                  ;OPERANDS, EXECUTE THE MODD INSTRUCTION AND CHECK THE RESULTS.
10457                                  ;IT IS CALLED THUS:
10458                                  ;
10459                                  ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
10460                                  ;               FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
10461                                  ;               FRES:   .WORD   X,X,X,X         ;FRACTIONAL RESULT
10462                                  ;               INTRES: .WORD   X,X,X,X         ;INTEGER RESULT
10463                                  ;               ERFRES: .WORD   X,X,X,X         ;ERROR FRACTION RESULT
```

```
10464                                   ;               ERINTRES:.WORD  X,X,X,X          ;ERROR INTEGER RESULT
10465                                   ;               FPSB:   .WORD   X                ;FPS BEFORE EXECUTION
10466                                   ;               FPSA:   .WORD   X                ;FPS AFTER EXECUTION
10467                                   ;               ERR1:   ERROR   X                ;FRACTION ERROR
10468                                   ;                       BR      CONT
10469                                   ;               ERR2:   ERROR   X                ;INTEGER ERROR
10470                                   ;               CONT:                            ;RETURN ADDRESS
10471                                   ;
10472                                   ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODD
10473                                   ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
10474                                   ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
10475                                   ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
10476                                   ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
10477                                   ;THEN MODDOV WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
10478                                   ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
10479                                   ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
10480                                   ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
10481                                   ;FAILURE MATCHES THE TRUE RESULT THEN  MODDOV PASSES CONTROL TO THE
10482                                   ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
10483                                   ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
10484                                   ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
10485                                   ;IF A MATCH IS MADE HOWEVER, MODDOV WILL RETURN CONTROL TO THE ERROR
10486                                   ;CALL AT ERR2.
10487
10488  055712  012601                   MODDOV: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
10489  055714  012700  000200                   MOV     #200,R0         ;SET FD MODE.
10490  055720  170100                           LDFPS   R0
10491  055722  010100                           MOV     R1,R0           ;SET UP AC0
10492  055724  172410                           LDD     (R0),AC0
10493  055726  012700  052226                   MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
10494  055732  172510                           LDD     (R0),AC1
10495  055734  016100  000060                   MOV     60(R1),R0       ;SET UP THE FPS.
10496  055740  170100                           LDFPS   R0
10497  055742  012737  055756  001236           MOV     #1$,@#$TMP2
10498  055750  010100                           MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
10499  055752  062700  000010                   ADD     #10,R0
10500
10501  055756  171410                   1$:     MODD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
10502
10503  055760  170305                           STST    R5              ;GET THE FPS.
10504  055762  170204                           STFPS   R4              ;GET THE FPS.
10505  055764  012700  000200                   MOV     #200,R0         ;SET FD MODE.
10506  055770  170100                           LDFPS   R0
10507  055772  012700  056300                   MOV     #MODDD0,R0      ;GET THE FRACTIONAL RESULT.
10508  055776  174010                           STD     AC0,(R0)
10509  056000  012700  056310                   MOV     #MODDD1,R0      ;GET THE INTEGER RESULT.
10510  056004  174110                           STD     AC1,(R0)
10511
10512  056006  010102                           MOV     R1,R2           ;SAVE THE DATA IN CASE OF ERROR.
10513  056010  010237  001240                   MOV     R2,@#$TMP3
10514  056014  062702  000010                   ADD     #10,R2
10515  056020  010237  001242                   MOV     R2,@#$TMP4
10516  056024  062702  000010                   ADD     #10,R2
10517  056030  010237  001244                   MOV     R2,@#$TMP5
10518  056034  062702  000010                   ADD     #10,R2
10519  056040  010237  001246                   MOV     R2,@#$TMP6
```

```
10520  056044  012737  056300  001250          MOV     #MODDD0,@#$TMP7
10521  056052  012737  056310  001252          MOV     #MODDD1,@#$TMP10
10522  056060  010437  001254                  MOV     R4,@#$TMP11
10523  056064  016137  000062  001256          MOV     62(R1),@#$TMP12
10524  056072  010537  001260                  MOV     R5,@#$TMP13
10525  056076  016137  000064  001262          MOV     64(R1),@#$TMP14
10526
10527  056104  012702  056300                  MOV     #MODDD0,R2      ;CHECK THE FRACTIONAL RESULT.
10528  056110  010103                          MOV     R1,R3
10529  056112  062703  000020                  ADD     #20,R3
10530  056116  012700  000004                  MOV     #4,R0
10531  056122  022223                  2$:     CMP     (R2)+,(R3)+
10532  056124  001023                          BNE     10$             ;BRANCH IF INCORRECT.
10533  056126  077003                          SOB     R0,2$
10534
10535  056130  012702  056310                  MOV     #MODDD1,R2      ;CHECK THE INTEGER RESULT.
10536  056134  010103                          MOV     R1,R3
10537  056136  062703  000030                  ADD     #30,R3
10538  056142  012700  000004                  MOV     #4,R0
10539  056146  022223                  3$:     CMP     (R2)+,(R3)+
10540  056150  001013                          BNE     15$             ,BRANCH IF INCORRECT.
10541  056152  077003                          SOB     R0,3$
10542
10543
10544  056154  026104  000062                  CMP     62(R1),R4       ;CHECK THE FPS.
10545  056160  001027                          BNE     20$             ;BRANCH IF INCORRECT.
10546
10547  056162  026105  000064                  CMP     64(R1),R5       ;CHECK THE FEC.
10548  056166  001033                          BNE     25$
10549
10550  056170  000161  000074          9$:     JMP     74(R1)          ;RETURN.
10551
10552                                  ;FRACTIONAL ERROR.
10553  056174                          10$:                            ;THE ERROR WAS NOT ANTICIPATED SO
10554  056174  104002                  12$:    ERROR   2               ;REPORT THE INCORRECT FRACTION HERE.
10555  056176  000774                          BR      9$
10556
10557                                  ;INTEGER ERROR.
10558  056200  012702  056310          15$:    MOV     #MODDD1,R2      ;WAS THE INTEGER ERROR ANTICIPATED?
10559  056204  010103                          MOV     R1,R3
10560  056206  062703  000050                  ADD     #50,R3
10561  056212  012705  000004                  MOV     #4,R5
10562  056216  022223                  60$:    CMP     (R2)+,(R3)+
10563  056220  001005                          BNE     17$             ;BRANCH IF NOT ANTICIPATED.
10564  056222  077503                          SOB     R5,60$
10565  056224  010102                          MOV     R1,R2           ;THE ERROR WAS ANTICIAPTED SO RETURN
10566  056226  062702  000072                  ADD     #72,R2          ;TO THE ERROR REPORT IN THE CALLING
10567                                                                  ;ROUTINE.
10568  056232  000112                          JMP     (R2)
10569
10570  056234                          16$:                            ;THE ERROR WAS NOT ANTICIPATED SO REPORT
10571  056234  104002                  17$:    ERROR   2               ;THE INTEGER FAILURE HERE.
10572  056236  000754                          BR      9$
10573
10574                                  ;FPS INCORRECT.
10575  056240  010437  001254          20$:    MOV     R4,@#$TMP11     ;REPORT INCORRECT FPS.
```

```
10576  056244  016137  000062  001256          MOV     62(R1),@#$TMP12
10577  056252  104002                  21$:    ERROR   2
10578  056254  000745                          BR      9$
10579
10580                                  ;REPORT FEC ERROR.
10581  056256  010537  001260          25$:    MOV     R5,@#$TMP13
10582  056262  016137  000064  001262          MOV     64(R1),@#$TMP14
10583  056270  010102                          MOV     R1,R2
10584  056272  062702  000066                  ADD     #66,R2
10585  056276  000112                          JMP     (R2)
10586
10587  056300  000000  000000  000000  MODDD0: .WORD   0,0,0,0
10588  056306  000000
10589
10590  056310  000000  000000  00000C  MODDD1: .WORD   0,0,0,0
10591  056316  000000
10592
10593  056320                          NNNDONE:
10594  056320  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
10595                                                                  ;SEE IF THE USER HAS EXPRESSED
10596                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
10597                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
10598                                                                  ;THE USER TYPED CONTROL G?).
10599
10600
10601                                  ;;*****************************************************************
10602                                  ;*TEST 60        MORE MICROCODES COVERAGE
10603                                  ;*THIS TEST WILL PROVIDE ADDITIONAL MICRO-CODE LOCATIONS COVERAGE
10604                                  ;*IN FPP1 AND FPP2.
10605                                  ;*
10606                                  ;;*****************************************************************
10607  056322  000004                  TST60:  SCOPE
10608  056324                          XX1:
10609  056324  104414                          LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
10610  056326  012737  056342  000244  XT1:    MOV     #XT1A,@#244
10611  056334  170227  000000                  STFPS   #0
10612  056340  000401                          BR      XT2
10613  056342  104001                  XT1A:   ERROR   1               ;SHOULD NOT TRAP
10614
10615  056344  012700  177777          XT2:    MOV     #-1,R0
10616  056350  170127  000000                  LDFPS   #0
10617  056354  170200                          STFPS   R0
10618  056356  005700                          TST     R0
10619  056360  001401                          BEQ     XT2A
10620  056362  104001                          ERROR   1               ;FPS IS NOT ZEROED
10621  056364  012700  057124          XT2A:   MOV     #XPAT0,R0
10622  056370  172440                          LDF     -(R0),AC0
10623  056372  022700  057120                  CMP     #XPAT0-4,R0
10624  056376  001401                          BEQ     XT2B
10625  056400  104001                          ERROR   1               ;R0 WAS NOT DECR BY 4
10626  056402  170200                  XT2B:   STFPS   R0
10627  056404  022700  000004                  CMP     #4,R0           ;CHECK IF FZ IS SET?
10628  056410  001401                          BEQ     XT3
10629  056412  104001                          ERROR   1
10630
10631  056414  170127  000000          XT3:    LDFPS   #0
```

```
10632  056420  012700  057124                MOV     #XPAT0,R0
10633  056424  174040                      STF     AC0,-(R0)
10634  056426  022700  057120                CMP     #XPAT0-4,R0
10635  056432  001401                      BEQ     XT3A            ;R0 WAS NOT DECR BY 4
10636  056434  104001                      ERROR   1
10637  056436  170200              XT3A:   STFPS   R0
10638  056440  005700                      TST     R0
10639  056442  001401                      BEQ     XT4
10640  056444  104001                      ERROR   1
10641
10642  056446  170127  000000        XT4:    LDFPS   #0
10643  056452  012737  056476  000244        MOV     #XT4A,@#244
10644  056460  170127  004000                LDFPS   #04000          ;INTRPT ON UNDEFINED VARIABLE
```

```
10645  056464  172437  057124                 LDF     @#XPAT0,AC0
10646  056470  174437  057154                 DIVF    @#XPAT3,AC0     ;GET UNDEFINED VARIABLE, _0
10647  056474  104001                         ERROR   1               ;MISSING INTERRUPT TO 244
10648  056476  170200                 XT4A:   STFPS   R0
10649  056500  022700  104004                 CMP     #104004,R0      ;CHECK: FER,FIUV,FZ ARE SET?
10650  056504  001401                         BEQ     XT4B
10651  056506  104001                         ERROR   1
10652  056510  012700  057114         XT4B:   MOV     #XBUF,R0
10653  056514  174010                         STF     AC0,(R0)
10654  056516  005737  057114                 TST     @#XBUF
10655  056522  001401                         BEQ     XT5
10656  056524  104001                         ERROR   1               ;AC0 SHOULD REMAIN ZEROED
10657
10658  056526  012737  056546  000244 XT5:    MOV     #XT5A,@#244
10659  056534  170127  004000                 LDFPS   #04000          ;INTRPT ON UNDEFINED VARIBALE
10660  056540  177437  057154                 LDCDF   @#XPAT3,AC0     ;GET UNDEFINED VARIABLE, _0
10661  056544  104001                         ERROR   1               ;MISSING INTERRUPT TO 244
10662  056546  170200                 XT5A:   STFPS   R0
10663  056550  022700  104014                 CMP     #104014,R0      ;CHECK: FER,FIUV,FN,FZ ARE SET?
10664  056554  001401                         BEQ     XT5B
10665  056556  104001                         ERROR   1
10666  056560  012700  057114         XT5B:   MOV     #XBUF,R0
10667  056564  174010                         STF     AC0,(R0)
10668  056566  005737  057114                 TST     @#XBUF
10669  056572  001401                         BEQ     XT6
10670  056574  104001                         ERROR   1               ;AC0 SHOULD BE ZEROED
10671
10672  056576                         XT6:
10673  056576  104414                         LPERR                   ;SET UP THE LOOP ON ERROR ADDRESS.
10674  056600  012737  056624  000244         MOV     #XT6A,@#244
10675  056606  170127  004000                 LDFPS   #04000          ;INTRPT ON UNDEFINED VARIBALE
10676  056612  172437  057124                 LDF     @#XPAT0,AC0
10677  056616  172037  057154                 ADDF    @#XPAT3,AC0
10678  056622  104002                         ERROR   2               ;MISSING INTERRUPT TO 244
10679  056624  170200                 XT6A:   STFPS   R0
10680  056626  022700  104004                 CMP     #104004,R0      ;CHECK: FER,FIUV,FZ ARE SET?
10681  056632  001401                         BEQ     XT6B
10682  056634  104002                         ERROR   2
10683  056636  012700  057114         XT6B:   MOV     #XBUF,R0
10684  056642  174010                         STF     AC0,(R0)
10685  056644  005737  057114                 TST     @#XBUF
10686  056650  001401                         BEQ     XT7
10687  056652  104002                         ERROR   2               ;AC0 SHOULD BE ZEROED
10688
10689  056654  170127  000000         XT7:    LDFPS   #0
10690  056660  172437  057164                 LDF     @#XPAT4,AC0
10691  056664  175437  057214                 STCFI   AC0,@#XPAT0
10692  056670  022737  000002  057214         CMP     #2,@#XPAT0      ;CHECK DATA
10693  056676  001401                         BEQ     XT8
10694  056700  104001                         ERROR   1
10695
10696  056702  170127  000100         XT8:    LDFPS   #100            ;SET FL
10697  056706  172437  057164                 LDF     @#XPAT4,AC0
10698  056712  175467  000276                 STCFI   AC0,XPAT0
10699  056716  022737  000002  057216         CMP     #2,@#XPAT0+2
10700  056724  001401                         BEQ     XT9
```

```
10701  056726  104001                          ERROR   1
10702
10703                                  ;START OF FPP2
10704  056730  170127  000000          XT9:    LDFPS   #0
10705  056734  172437  057124                  LDF     @#XPAT0,AC0
10706  056740  172037  057164                  ADDF    @#XPAT4,AC0
10707  056744  170200                          STFPS   R0
10708  056746  005700                          TST     R0
10709  056750  001401                          BEQ     XT10
10710  056752  104002                          ERROR   2
10711
10712  056754  170127  000000          XT10:   LDFPS   #0
10713  056760  172437  057164                  LDF     @#XPAT4,AC0
10714  056764  173037  057164                  SUBF    @#XPAT4,AC0
10715  056770  170200                          STFPS   R0
10716  056772  022700  000004                  CMP     #4,R0
10717  056776  001401                          BEQ     XT11
10718  057000  104002                          ERROR   2
10719
10720  057002  170127  000000          XT11:   LDFPS   #0
10721  057006  172437  057164                  LDF     @#XPAT4,AC0
10722  057012  173437  057164                  CMPF    @#XPAT4,AC0
10723  057016  170200                          STFPS   R0
10724  057020  022700  000004                  CMP     #4,R0           ;CHECK IF FZ IS SET?
10725  057024  001401                          BEQ     XT12
10726  057026  104002                          ERROR   2
10727
10728  057030  170127  000000          XT12:   LDFPS   #0
10729  057034  172437  057164                  LDF     @#XPAT4,AC0
10730  057040  174437  057144                  DIVF    @#XPAT2,AC0
10731  057044  012700  057114                  MOV     #XBUF,R0
10732  057050  174010                          STF     AC0,(R0)
10733  057052  022737  040176  057114          CMP     #040176,@#XBUF  ;CHECK DATA
10734  057060  001401                          BEQ     XT13
10735  057062  104002                          ERROR   2
10736
10737  057064  170127  000000          XT13:   LDFPS   #0
10738  057070  172437  057174                  LDF     @#XPAT5,AC0
10739  057074  174437  057204                  DIVF    @#XPAT6,AC0
10740  057100  170200                          STFPS   R0
10741  057102  022700  000004                  CMP     #4,R0
10742  057106  001401                          BEQ     XT13B
10743  057110  104002                          ERROR   2
10744  057112  000444                  XT13B:  BR      XNEXT
10745
10746
10747  057114  000000  000000  000000  XBUF:   .WORD   0,0,0,0
10748  057122  000000
10749  057124  000000  000000  000000  XPAT0:  .WORD   0,0,0,0
10750  057132  000000
10751  057134  000001  000001  000001  XPAT1:  .WORD   1,1,1,1
10752  057142  000001
10753  057144  040401  000000  000000  XPAT2:  .WORD   40401,0,0,0
10754  057152  000000
10755  057154  100000  000000  000000  XPAT3:  .WORD   100000,0,0,0
10756  057162  000000
```

```
10757  057164  040400  000000  000000  XPAT4:  .WORD   040400,0,0,0
10758  057172  000000
10759  057174  000207  000000  000000  XPAT5:  .WORD   207,0,0,0
10760  057202  000000
10761  057204  077007  000000  000000  XPAT6:  .WORD   77007,0,0,0
10762  057212  000000
10763  057214  000000  000000  000000  XPAT0:  .WORD   0,0,0,0
10764  057222  000000
10765
10766  057224                          XNEXT:
```

```
                                        .SBTTL END OF PASS ROUTINE
                                        ;;*****************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*INDICATE END-OF-PROGRAM AFTER 1 PASS THRU THE PROGRAM
                                        ;*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
                                        ;*IF SW12=1 INHIBIT TRACE TRAP
                                        ;*IF THERE IS A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO LOOP
                                        ;;*****************************************************************
057224  000004                  $EOP:   SCOPE
057226  005067  121650                  CLR     $TSTNM          ;ZERO THE TEST NUMBER
057232  005067  122044                  CLR     $TIMES          ;ZERO THE NUMBER OF ITERATIONS
057236  005267  122062                  INC     $PASS           ;INCREMENT THE PASS NUMBER
057242  042767  100000  122054          BIC     #100000,$PASS   ;DON'T ALLOW A NEGATIVE PASS NUMBER
057250  005327                          DEC     (PC)+           ;LOOP?
057252  000001          $EOPCT: .WORD   1                       ;1 PASS FIRST TIME (QV)!!!
057254  003045                          BGT     $DOAGN          ;YES
057256  012737                          MOV     (PC)+,@(PC)+    ;RESTORE COUNTER
057260  000001          $ENDCT: .WORD   1
057262  057252                          $EOPCT
057264  104401  057437                  TYPE    ,$ENDMG         ;TYPE 'END PASS #''
057270  016746  122030                  MOV     $PASS,-(SP)     ;SAVE PASS COUNT FOR TYPEOUT
057274  104405                          TYPDS                   ;TYPE PASS COUNT IN DECIMAL
057276  104401  057434                  TYPE    ,$ENULL         ;TYPE A NULL CHARACTER STRING
057302  013700  000042          $GET42: MOV     @#42,R0         ;GET MONITOR ADDRESS
057306  001411                          BEQ     DOAGIN          ;BRANCH IF NO MONITOR
057310  005046                          CLR     -(SP)           ;INSURE THE 'T'' BIT IS CLEAR
057312  012746  057320                  MOV     #$CLR.T,-(SP)   ;SETUP FOR AN RTI OR RTT
057316  000442                          BR      $RTRN           ;GO DO AN RTI OR RTT TO LOAD THE PSW
                                                                ;WITH A CLEARED 'T'' BIT
057320  000005          $CLR.T: RESET                           ;CLEAR THE WORLD
057322  004710          $ENDAD: JSR     PC,(R0)                 ;GO TO THE  MONITOR
057324  000240                  NOP                             ;SAVE ROOM
057326  000240                  NOP                             ;FOR
057330  000240                  NOP                             ;ACT11
057332  013737  000004  001232  DOAGIN: MOV     @#4,@#$TMP0     ;SAVE CONTENTS OF LOCATION 4
057340  012737  057356  000004          MOV     #1$,@#4         ;SET UP INCASE OF TRAP
057346  012737  000001  164000          MOV     #1,@#164000     ;NOTIFY MULTI-TESTER
057354  000402                          BR      2$              ;NO TRAP SO DON'T RESET STACK
057356  062706  000004          1$:     ADD     #4,SP           ;RESET STACK AFTER TRAP
057362  013737  001232  000004  2$:     MOV     @#$TMP0,@#4     ;RESTORE CONTENTS OF LOCATION 4
057370  104400          $DOAGN: TRAP                            ;PUSH OLD PSW AND PC ON STACK
057372  042716  000020                  BIC     #20,(SP)        ;CLEAR THE 'T'' BIT
057376  032777  010000  121534          BIT     #BIT12,@SWR     ;RUN WITH TRACE TRAP?
057404  001005                          BNE     1$              ;BRANCH IF NO
057406  005167  000020                  COM     $TBIT           ;IS IT TIME FOR TRACE TRAP
057412  100402                          BMI     1$              ;BRANCH IF NO
057414  052716  000020                  BIS     #20,(SP)        ;SET TRACE TRAP
057420  012746  057426          1$:     MOV     #$LOOP,-(SP)    ;RETURN HERE FROM RTI
057424  000002          $RTRN:  RTI                             ;RETURN--THIS IS CHANGED TO AN 'RTT''
                                                                ;IF IT IS A LEGAL INSTRUCTION
057426  000137          $LOOP:  JMP     @(PC)+                  ;RETURN TO TESTING
057430  002160          $RTNAD: .WORD   LOOP
057432  000000          $TBIT:  .WORD   0                       ;'T'' BIT STATE INDICATOR
057434     377     377  000     $ENULL: .BYTE   -1,-1,0          ;NULL CHARACTER STRING
```

```
10823  057437  015     042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
10824  057444  050040  051501  020123
10825  057452  000043
10826                                  .EVEN
10827                                  .SBTTL  SCOPE HANDLER ROUTINE
10828
10829                                  ;;*****************************************************************
10830                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
10831                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
10832                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
10833                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10834                                  ;*SW14=1        LOOP ON TEST
10835                                  ;*SW11=1        INHIBIT ITERATIONS
10836                                  ;*SW09=1        LOOP ON ERROR
10837                                  ;*SW08=1        LOOP ON TEST IN SWR<7:0>
10838                                  ;*CALL
10839                                  ;*      SCOPF           ;;SCOPE=IOT
10840
10841  057454                          $SCOPE:
10842  057454  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
10843  057456  032777  040000  121454 1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
10844  057464  001114                         BNE     $OVER           ;;YES IF SW14=1
10845                                  ;#####START OF CODE FOR THE XOR TESTER#####
10846  057466  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
10847                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
10848  057470  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
10849  057474  012737  057514  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
10850  057502  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
10851  057506  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
10852  057512  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
10853  057514  022626                 5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
10854  057516  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
10855  057522  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
10856  057524                         6$:;#####END OF CODE FOR THE XOR TESTER#####
10857  057524  032777  000400  121406         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
10858  057532  001404                         BEQ     2$              ;;BR IF NO
10859  057534  127767  121400  121340         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
10860  057542  001465                         BEQ     $OVER           ;;BR IF YES
10861  057544  105767  121333         2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
10862  057550  001421                         BEQ     3$              ;;BR IF NO
10863  057552  126767  121337  121323         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
10864  057560  101015                         BHI     3$              ;;BR IF NO
10865  057562  032777  001000  121350         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
10866  057570  001404                         BEQ     4$              ;;BR IF NO
10867  057572  016767  121312  121306 7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
10868  057600  000446                         BR      $OVER
10869  057602  105067  121275         4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
10870  057606  005067  121470                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
10871  057612  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
10872  057614  032777  004000  121316 3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
10873  057622  001011                         BNE     1$              ;;BR IF YES
10874  057624  005767  121474                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
10875  057630  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
10876  057632  005267  121246                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
10877  057636  026767  121440  121240         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
10878  057644  002024                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
```

```
10879  057646  012767  000001  121230  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
10880  057654  016767  000052  121420          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
10881  057662  105267  121214          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
10882  057666  116767  121210  121426          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
10883  057674  011667  121206                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
10884  057700  011667  121204                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
10885  057704  005067  121374                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
10886  057710  112767  000001  121177          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
10887  057716  016777  121160  121216  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
10888  057724  016716  121156                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
10889  057730  000002                          RTI                     ;;FIXES PS
10890  057732  000001                  $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
10891
10892                                  .SBTTL  ERROR HANDLER ROUTINE
10893
10894                                  ;;**********************************************************************
10895                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
10896                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
10897                                  ;*AND GO TO ERTYPE ON ERROR
10898                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10899                                  ;*SW15=1         HALT ON ERROR
10900                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
10901                                  ;*SW10=1         BELL ON ERROR
10902                                  ;*SW09=1         LOOP ON ERROR
10903                                  ;*CALL
10904                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
10905
10906  057734                  $ERROR:
10907  057734  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
10908  057736  105267  121141          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
10909  057742  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
10910  057744  016777  121132  121170          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
10911  057752  032777  002000  121160          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
10912  057760  001402                          BEQ     1$              ;;NO - SKIP
10913  057762  104401  001306                  TYPE    ,$BELL          ;;RING BELL
10914  057766  005267  121120          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
10915  057772  011667  121120                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
10916  057776  162767  000002  121112          SUB     #2,$ERRPC
10917  060004  117767  121106  121102          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
10918  060012  032777  020000  121120          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
10919  060020  001004                          BNE     20$             ;;SKIP TYPEOUTS
10920  060022  004767  002314                  JSR     PC,ERTYPE       ;;GO TO USER ERROR ROUTINE
10921  060026  104401  001313                  TYPE    ,$CRLF
10922  060032                          20$:
10923  060032  122767  000001  121276          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
10924  060040  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
10925  060042  116767  121046  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
10926  060050  004767  001126                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
10927  060054      000                 21$:    .BYTE   0
10928  060055      000                         .BYTE   0
10929  060056  000777                  22$:    BR      22$             ;;APT ERROR LOOP
10930  060060  005777  121054          2$:     TST     @SWR            ;;HALT ON ERROR
10931  060064  100002                          BPL     3$              ;;SKIP IF CONTINUE
10932  060066  000000                          HALT                    ;;HALT ON ERROR!
10933  060070  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
10934  060072  032777  001000  121040  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
```

```
10935  060100  001402                          BEQ     4$              ;;BR IF NO
10936  060102  016716  121002                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
10937  060106  005767  121172          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
10938  060112  001402                          BEQ     5$              ;;BR IF NONE
10939  060114  016716  121164                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
10940  060120                          5$:
10941  060120  022737  057322  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
10942  060126  001001                          BNE     6$              ;;BRANCH IF NO
10943  060130  000000                          HALT                    ;;YES
10944  060132                          6$:
10945  060132  000002                          RTI                     ;;RETURN
10946
10947                                  .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
10948
10949                                  ;;**********************************************************************
10950                                  ;*SAVE R0-R5
10951                                  ;*CALL:
10952                                  ;*      SAVREG
10953                                  ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
10954                                  ;*
10955                                  ;*TOP---(+16)
10956                                  ;* +2---(+18)
10957                                  ;* +4---R5
10958                                  ;* +6---R4
10959                                  ;* +8---R3
10960                                  ;*+10---R2
10961                                  ;*+12---R1
10962                                  ;*+14---R0
10963
10964  060134                          $SAVREG:
10965  060134  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10966  060136  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10967  060140  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10968  060142  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10969  060144  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10970  060146  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10971  060150  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
10972  060154  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
10973  060160  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
10974  060164  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
10975  060170  000002                          RTI
10976
10977                                  ;*RESTORE R0-R5
10978                                  ;*CALL:
10979                                  ;*      RESREG
10980  060172                          $RESREG:
10981  060172  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
10982  060176  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
10983  060202  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
10984  060206  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
10985  060212  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10986  060214  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10987  060216  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10988  060220  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10989  060222  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10990  060224  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
```

```
10991  060226  000002                        RTI
10992
10993                                  .SBTTL  TYPE ROUTINE
10994
10995                                  ;;*****************************************************************
10996                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
10997                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
10998                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
10999                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
11000                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
11001                                  ;*
11002                                  ;*CALL:
11003                                  ;*1) USING A TRAP INSTRUCTION
11004                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
11005                                  ;*OR
11006                                  ;*      TYPE
11007                                  ;*      MESADR
11008                                  ;*
11009
11010  060230  105767  120723          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
11011  060234  100002                          BPL     1$              ;;BR IF YES
11012  060236  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
11013  060240  000430                          BR      3$              ;;LEAVE
11014  060242  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
11015  060244  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
11016  060250  122767  000001  121060          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
11017  060256  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
11018  060260  132767  000100  121051          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
11019  060266  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
11020  060270  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
11021  060274  004767  000672                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
11022  060300  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
11023  060302  132767  000040  121027  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
11024  060310  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
11025  060312  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
11026  060314  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
11027  060316  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
11028  060320  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
11029  060322  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
11030  060326  000002                          RTI                     ;;RETURN
11031  060330  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
11032  060334  001430                          BEQ     8$
11033  060336  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
11034  060342  001006                          BNE     5$
11035  060344  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
11036  060346  104401                          TYPE                    ;;TYPE A CR AND LF
11037  060350  001313                          $CRLF
11038  060352  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
11039  060356  000755                          BR      2$              ;;GET NEXT CHARACTER
11040  060360  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
11041  060364  126726  120566          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
11042  060370  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
11043  060372  016746  120556                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
11044                                                                  ;;AND THE NULL CHAR.
11045  060376  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
11046  060402  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
```

```
11047  060404  004767  000032          JSR     PC,$TYPEC       ;;GO TYPE A NULL
11048  060410  105367  000072          DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
11049  060414  000770                  BR      7$              ;;LOOP
11050
11051                                  ;HORIZONTAL TAB PROCESSOR
11052
11053  060416  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
11054  060422  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
11055  060426  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
11056  060434  001372                          BNE     9$              ;;TAB STOP
11057  060436  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
11058  060440  000724                          BR      2$              ;;GET NEXT CHARACTER
11059  060442  105777  120502          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
11060  060446  100375                          BPL     $TYPEC
11061  060450  116677  000002  120474        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
11062  060456  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
11063  060464  001003                        BNE     1$              ;;BRANCH IF NO
11064  060466  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
11065  060472  000406                        BR      $TYPEX          ;;EXIT
11066  060474  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
11067  060502  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
11068  060504  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
11069  060506  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
11070  060510  000207          $TYPEX: RTS     PC
11071
11072
11073                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
11074
11075                                  ;;**********************************************************************
11076                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
11077                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
11078                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
11079                                  ;*CALL:
11080                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11081                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
11082                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
11083                                  ;*      .BYTE   M               ;;M=1 OR 0
11084                                  ;*                                      ;;1=TYPE LEADING ZEROS
11085                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
11086                                  ;*
11087                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
11088                                  ;*$TYPOS OR $TYPOC
11089                                  ;*CALL:
11090                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11091                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
11092                                  ;*
11093                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
11094                                  ;*CALL:
11095                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11096                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
11097
11098  060512  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
11099  060516  116667  000001  000211        MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
11100  060524  112667  000207                MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
11101  060530  062716  000002                ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
11102  060534  000406                        BR      $TYPON
```

```
11103  060536  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
11104  060544  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
11105  060552  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
11106  060560  010346                          MOV     R3,-(SP)        ;;SAVE R3
11107  060562  010446                          MOV     R4,-(SP)        ;;SAVE R4
11108  060564  010546                          MOV     R5,-(SP)        ;;SAVE R5
11109  060566  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
11110  060572  005404                          NEG     R4
11111  060574  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
11112  060600  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
11113  060604  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
11114  060610  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
11115  060614  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
11116  060616  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
11117  060620  000404                          BR      3$              ;;GO DO MSB
11118  060622  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
11119  060624  006105                          ROL     R5
11120  060626  006105                          ROL     R5
11121  060630  010503                          MOV     R5,R3
11122  060632  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
11123  060634  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
11124  060640  100016                          BPL     7$              ;;BR IF NO
11125  060642  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
11126  060646  001002                          BNE     4$              ;;TEST FOR 0
11127  060650  005704                          TST     R4              ;;SUPPRESS THIS 0?
11128  060652  001403                          BEQ     5$              ;;BR IF YES
11129  060654  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
11130  060656  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
11131  060662  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
11132  060666  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
11133  060672  104401  060732                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
11134  060676  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
11135  060702  003347                          BGT     2$              ;;BR IF MORE TO DO
11136  060704  002402                          BLT     6$              ;;BR IF DONE
11137  060706  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
11138  060710  000744                          BR      2$              ;;GO DO THE LAST DIGIT
11139  060712  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
11140  060714  012604                          MOV     (SP)+,R4        ;;RESTORE R4
11141  060716  012603                          MOV     (SP)+,R3        ;;RESTORE R3
11142  060720  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
11143  060726  012616                          MOV     (SP)+,(SP)
11144  060730  000002                          RTI                     ;;RETURN
11145  060732     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
11146  060733     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
11147  060734     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
11148  060735     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
11149  060736  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
11150
11151                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
11152
11153                                  ;;**************************************************************
11154                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
11155                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
11156                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
11157                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
11158                                  ;*REPLACED WITH SPACES.
```

```
11159                                   ;*CALL:
11160                                   ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
11161                                   ;*      TYPDS                   ;;GO TO THE ROUTINE
11162
11163  060740                           $TYPDS:
11164  060740  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11165  060742  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11166  060744  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11167  060746  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11168  060750  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11169  060752  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
11170  060756  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
11171  060762  100004                          BPL     1$              ;;BR IF INPUT IS POS.
11172  060764  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
11173  060766  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
11174  060774  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
11175  060776  012703  061154                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
11176  061002  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
11177  061006  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
11178  061010  016001  061144                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
11179  061014  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
11180  061016  002402                          BLT     4$              ;;BR IF DONE
11181  061020  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
11182  061022  000774                          BR      3$
11183  061024  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
11184  061026  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
11185  061030  001002                          BNE     5$              ;;FALL THROUGH IF 0
11186  061032  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
11187  061034  100407                          BMI     7$              ;;BR IF YES
11188  061036  106316                  5$:     ASLB    (SP)            ;;MSD?
11189  061040  103003                          BCC     6$              ;;BR IF NO
11190  061042  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
11191  061050  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
11192  061054  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
11193  061060  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
11194  061062  005720                          TST     (R0)+           ;;JUST INCREMENTING
11195  061064  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
11196  061070  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
11197  061072  003002                          BGT     8$              ;;GO TO EXIT
11198  061074  010502                          MOV     R5,R2           ;;GET THE LSD
11199  061076  000764                          BR      6$              ;;GO CHANGE TO ASCII
11200  061100  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
11201  061102  100003                          BPL     9$              ;;BR IF NO
11202  061104  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
11203  061112  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
11204  061114  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
11205  061116  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11206  061120  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11207  061122  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11208  061124  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11209  061126  104401  061154                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
11210  061132  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
11211  061140  012616                          MOV     (SP)+,(SP)
11212  061142  000002                          RTI                     ;;RETURN TO USER
11213  061144  023420                  $DTBL:  10000.
11214  061146  001750                          1000.
```

```
11215  061150  000144                          100.
11216  061152  000012                          10.
11217  061154  000004                $DBLK:    .BLKW   4
11218
11219                                .SBTTL  APT COMMUNICATIONS ROUTINE
11220
11221                                ;;*****************************************************************
11222  061164  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
11223  061172  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
11224  061200  000403                          BR      $ATYC
11225  061202  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
11226  061210                          $ATYC:
11227  061210  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11228  061212  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11229  061214  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
11230  061220  001450                          BEQ     5$              ;;IF NOT:  BR
11231  061222  122767  000001  120106          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
11232  061230  001031                          BNE     3$              ;;IF NOT:  ER
11233  061232  132767  000100  120077          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
11234  061240  001425                          BEQ     3$              ;;IF NOT:  BR
11235  061242  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
11236  061246  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
11237  061254  005767  120036          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
11238  061260  001375                          BNE     1$              ;;IF NOT:  WAIT
11239  061262  010067  120044                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
11240  061266  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
11241  061270  001376                          BNE     2$
11242  061272  166700  120034                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
11243  061276  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
11244  061300  010067  120030                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
11245  061304  012767  000004  120004          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
11246  061312  000413                          BR      5$
11247  061314  017667  000004  000016  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
11248  061322  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
11249  061330  016746  116442                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
11250  061334  004767  176670                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
11251  061340  000000                  4$:     .WORD   0
11252  061342                          5$:
11253  061342  105767  000062          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
11254  061346  001416                          BEQ     12$             ;;IF NOT:  BR
11255  061350  005767  117762                  TST     $ENV            ;;RUNNING UNDER APT?
11256  061354  001413                          BEQ     12$             ;;IF NOT:  BR
11257  061356  005767  117734          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
11258  061362  001375                          BNE     11$             ;;IF NOT:  WAIT
11259  061364  017667  000004  117726          MOV     @4(SP),$FATAL   ;;GET ERROR #
11260  061372  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
11261  061400  005267  117712                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
11262  061404  105067  000020          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
11263  061410  105067  000013                  CLRB    $LFLG           ;;CLEAR LOG FLAG
11264  061414  105067  000006                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
11265  061420  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11266  061422  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11267  061424  000207                          RTS     PC              ;;RETURN
11268  061426     000                  $MFLG:  .BYTE   0               ;;MESSG. FLAG
11269  061427     000                  $LFLG:  .BYTE   0               ;;LOG FLAG
11270  061430     000                  $FFLG:  .BYTE   0               ;;FATAL FLAG
```

```
11271           061432                          .EVEN
11272           000200                  APTSIZE=200
11273           000001                  APTENV=001
11274           000100                  APTSPOOL=100
11275           000040                  APTCSUP=040
11276
11277                                   .SBTTL  TTY INPUT ROUTINE
11278
11279                                   ;;*********************************************************************
11280                                   .ENABL  LSB
11281
11282                                   ;;*********************************************************************
11283                                   ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11284                                   ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
11285                                   ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
11286                                   ;*WHEN OPERATING IN TTY FLAG MODE.
11287   061432  022767  000176  117500 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
11288   061440  001074                         BNE     15$             ;;BRANCH IF NO
11289   061442  105777  117476                 TSTB    @$TKS           ;;CHAR THERE?
11290   061446  100071                         BPL     15$             ;;IF NO, DON'T WAIT AROUND
11291   061450  117746  117472                 MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
11292   061454  042716  177600                 BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
11293   061460  022726  000007                 CMP     #7,(SP)+        ;;IS IT A CONTROL G?
11294   061464  001062                         BNE     15$             ;;NO, RETURN TO USER
11295   061466  126727  117442  000001         CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
11296   061474  001456                         BEQ     15$             ;;BRANCH IF YES
11297
11298   061476  104401  062041                 TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
11299   061502  104401  062046         $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
11300   061506  016746  116464                 MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
11301   061512  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11302   061514  104401  062057                 TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
11303   061520  005046                 19$:    CLR     -(SP)           ;;CLEAR COUNTER
11304   061522  005046                         CLR     -(SP)           ;;THE NEW SWR
11305   061524  105777  117414         7$:     TSTB    @$TKS           ;;CHAR THERE?
11306   061530  100375                         BPL     7$              ;;IF NOT TRY AGAIN
11307
11308   061532  117746  117410                 MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
11309   061536  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11310
11311
11312
11313   061542  021627  000025         9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
11314   061546  001005                         BNE     10$             ;;BRANCH IF NOT
11315   061550  104401  062034                 TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
11316   061554  062706  000006         20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
11317   061560  000757                         BR      19$             ;;LET'S TRY IT AGAIN
11318
11319
11320   061562  021627  000015         10$:    CMP     (SP),#15        ;;IS IT A <CR>?
11321   061566  001022                         BNE     16$             ;;BRANCH IF NO
11322   061570  005766  000004                 TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
11323   061574  001403                         BEQ     11$             ;;BRANCH IF YES
11324   061576  016677  000002  117334         MOV     2(SP),@SWR      ;;SAVE NEW SWR
11325   061604  062706  000006         11$:    ADD     #6,SP           ;;CLEAR UP STACK
11326   061610  104401  001313         14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
```

```
11327  061614  126727  117315  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
11328  061622  001003                          BNE     15$             ;;BRANCH IF NOT
11329  061624  012777  000100  117312          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
11330  061632  000002                  15$:    RTI                     ;;RETURN
11331  061634  004767  176602          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
11332  061640  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
11333  061644  002420                          BLT     18$             ;;BRANCH IF YES
11334  061646  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
11335  061652  003015                          BGT     18$             ;;BRANCH IF YES
11336  061654  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
11337  061660  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
11338  061664  001403                          BEQ     17$             ;;BRANCH IF YES
11339  061666  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
11340  061670  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
11341  061672  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
11342  061674  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
11343  061700  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
11344  061704  000707                          BR      7$              ;;GET THE NEXT ONE
11345  061706  104401  001312          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
11346  061712  000720                          BR      20$             ;;SIMULATE CONTROL-U
11347                                  .DSABL  LSB
11348
11349
11350                                  ;;*****************************************************************
11351                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
11352                                  ;*CALL:
11353                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
11354                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
11355                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
11356                                  ;
11357
11358  061714  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
11359  061716  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
11360  061724  105777  117214          1$:     TSTB    @$TKS           ;;WAIT FOR
11361  061730  100375                          BPL     1$              ;;A CHARACTER
11362  061732  117766  117210  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
11363  061740  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
11364  061746  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
11365  061754  001013                          BNE     3$              ;;BRANCH IF NO
11366  061756  105777  117162          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
11367  061762  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
11368  061764  117746  117156                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
11369  061770  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11370  061774  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
11371  062000  001366                          BNE     2$              ;;IF NOT DISCARD IT
11372  062002  000750                          BR      1$              ;;YES, RESUME
11373  062004  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
11374  062012  002407                          BLT     4$              ;;BRANCH IF YES
11375  062014  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
11376  062022  003003                          BGT     4$              ;;BRANCH IF YES
11377  062024  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
11378  062032  000002                  4$:     RTI                     ;;GO BACK TO USER
11379  062034  052536  005015     000  $CNTLU: .ASCIZ   /^U/<15><12>    ;;CONTROL 'U'
11380  062041     136  006507  000012  $CNTLG: .ASCIZ   /^G/<15><12>    ;;CONTROL 'G'
11381  062046  005015  053523  020122  $MSWR:  .ASCIZ   <15><12>/SWR - /
11382  062054  020075     000
```

```
11383  062057  040     047040  053505  $MNEW:  .ASCIZ  /  NEW = /
11384  062064  036440  000040
11385
11386                                  .SBTTL  TRAP DECODER
11387
11388                                  ;;*****************************************************************
11389                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
11390                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
11391                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
11392                                  ;*GO TO THAT ROUTINE.
11393
11394  062070  010046                 $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
11395  062072  016600  000002                 MOV     2(SP),R0        ;;GET TRAP ADDRESS
11396  062076  005740                         TST     -(R0)           ;;BACKUP BY 2
11397  062100  111000                         MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
11398  062102  006300                         ASL     R0              ;;POSITION FOR INDEXING
11399  062104  016000  062124                 MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
11400  062110  000200                         RTS     R0              ;;GO TO ROUTINE
11401
11402
11403                                  ;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO
11404
11405  062112  011646                 $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
11406  062114  016666  000004  000002         MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
11407  062122  000002                         RTI                     ;;RESTORE THE PSW
11408
11409                                  .SBTTL  TRAP TABLE
11410
11411                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
11412                                  ;*BY THE 'TRAP' INSTRUCTION.
11413
11414                                  ;       ROUTINE
11415                                  ;       -------
11416  062124  062112                 $TRPAD: .WORD   $TRAP2
11417  062126  060230                         $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
11418  062130  060536                         $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
11419  062132  060512                         $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
11420  062134  060552                         $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
11421  062136  060740                         $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
11422
11423  062140  061502                         $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
11424
11425  062142  061432                         $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
11426  062144  061714                         $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
11427  062146  060134                         $SAVREG ;;CALL=SAVREG   TRAP+11(104411) SAVE R0-R5 ROUTINE
11428  062150  060172                         $RESREG ;;CALL=RESREG   TRAP+12(104412) RESTORE R0-R5 ROUTINE
11429  062152  062620                         .RSET   ;;CALL=RSETUP   TRAP+13(104413) ROUTINE TO RESET STACK AND FPS
11430  062154  062612                         .LPER   ;;CALL=LPERR    TRAP+14(104414) ROUTINE TO SET LOOP ON ERROR ADDRESS
11431           000032                 $TERM=.-$TRPAD
11432
11433                                  .SBTTL  POWER DOWN AND UP ROUTINES
11434
11435                                  ;;*****************************************************************
11436                                  ;POWER DOWN ROUTINE
11437  062156  012737  062334  000024 $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
11438  062164  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
```

```
11439  062172  010046                          MOV     R0,-(SP)          ;;PUSH R0 ON STACK
11440  062174  010146                          MOV     R1,-(SP)          ;;PUSH R1 ON STACK
11441  062176  010246                          MOV     R2,-(SP)          ;;PUSH R2 ON STACK
11442  062200  010346                          MOV     R3,-(SP)          ;;PUSH R3 ON STACK
11443  062202  010446                          MOV     R4,-(SP)          ;;PUSH R4 ON STACK
11444  062204  010546                          MOV     R5,-(SP)          ;;PUSH R5 ON STACK
11445  062206  017746  116726                  MOV     @SWR,-(SP)        ;;PUSH @SWR ON STACK
11446  062212  010667  000122                  MOV     SP,$SAVR6         ;;SAVE SP
11447  062216  012737  062230  000024          MOV     #$PWRUP,@#PWRVEC  ;;SET UP VECTOR
11448  062224  000000                          HALT
11449  062226  000776                          BR      .-2               ;;HANG UP
11450
11451                                  ;;******************************************************************
11452                                  ;POWER UP ROUTINE
11453  062230  012737  062334  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC  ;;SET FOR FAST DOWN
11454  062236  016706  000076                  MOV     $SAVR6,SP         ;;GET SP
11455  062242  005067  000072                  CLR     $SAVR6            ;;WAIT LOOP FOR THE TTY
11456  062246  005267  000066          1$:     INC     $SAVR6            ;;WAIT FOR THE INC
11457  062252  001375                          BNE     1$                ;;OF  WORD
11458  062254  012677  116660                  MOV     (SP)+,@SWR        ;;POP STACK INTO @SWR
11459  062260  012605                          MOV     (SP)+,R5          ;;POP STACK INTO R5
11460  062262  012604                          MOV     (SP)+,R4          ;;POP STACK INTO R4
11461  062264  012603                          MOV     (SP)+,R3          ;;POP STACK INTO R3
11462  062266  012602                          MOV     (SP)+,R2          ;;POP STACK INTO R2
11463  062270  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
11464  062272  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
11465  062274  012737  062156  000024          MOV     #$PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
11466  062302  012737  000340  000026          MOV     #340,@#PWRVEC+2   ;;PRIO:7
11467  062310  104401                          TYPE                      ;;REPORT THE POWER FAILURE
11468  062312  062670                  $PWRMG: .WORD   POWERM            ;;POWER FAIL MESSAGE POINTER
11469  062314  012716                          MOV     (PC)+,(SP)        ;;RESTART AT START
11470  062316  001466                  $PWRAD: .WORD   START             ;;RESTART ADDRESS
11471  062320  042766  000020  000002          BIC     #20,2(SP)         ;;CLEAR "T" BIT
11472  062326  005067  175100                  CLR     $TBIT             ;;CLEAR THE "T" BIT FLAG
11473  062332  000002                          RTI
11474  062334  000000                  $ILLUP: HALT                      ;;THE POWER UP SEQUENCE WAS STARTED
11475  062336  000776                          BR      .-2               ;; BEFORE THE POWER DOWN WAS COMPLETE
11476  062340  000000                  $SAVR6: 0                         ;;PUT THE SP HERE
11477
11478                                          .SBTTL  ERROR TYPE OUT ROUTINE
11479                                  ;;******************************************************************
11480                                  ;;******************************************************************
11481                                  ;*THIS ROUTINE IS CALLED TO TYPE AN ERROR MESSAGE WHICH IS INCLUDED
11482                                  ;*IN THE ERROR MESSAGE DATA TABLE. IT IS CALLED BY THE $ERROR ROUTINE
11483                                  ;*OR BY FIRST SETTING $ITEMB EQUAL TO THE ERROR TABLE ITEM TO BE PRINTED
11484                                  ;*OUT AND THEN EXECUTING A:
11485                                  ;*              JSR     PC,ERTYPE
11486                                  ;*
11487  062342  104401                  ERTYPE: TYPE                                      ;TYPE A CRLF
11488  062344  001313                          .WORD   $CRLF
11489  062346  113737  001102  001232          MOVB    @#$TSTNM,@#$TMP0
11490  062354  042737  177400  001232          BIC     #177400,@#$TMP0
11491  062362  013737  001116  001234          MOV     @#$ERRPC,@#$TMP1          ;GET PC OF CALL
11492  062370  010046                          MOV     R0,-(SP)                  ;SAVE R0
11493
11494  062372  113700  001114                  MOVB    @#$ITEMB,R0               ;GET THE ITEM NUMBER.
```

```
11495  062376  042700  177400            BIC     #177400,R0
11496  062402  001007                    BNE     MULT
11497  062404  104401  062456    PCTYP:  TYPE    ,EMSG
11498
11499  062410  013746  001116            MOV     @#$ERRPC,-(SP)          ;IF ZERO THEN JUST
11500  062414  104402                    TYPOC                           ;PRINT THE PC
11501  062416  000137  062452            JMP     @#ERT5
11502  062422  005300            MULT:   DEC     R0                      ;OTHERWISE MULT R0 BY 2 TO
11503                                                                    ;GET ERROR MESSAGE POINTER
11504  062424  006300                    ASL     R0
11505  062426  062700  001442            ADD     #$ERRTB,R0
11506
11507  062432  011037  062442            MOV     (R0),@#2$               ;PICK UP THE ADDRESS
11508  062436  001405                    BEQ     ERT5                    ;OF THE EM, ERROR MESSAGE
11509  062440  104401                    TYPE
11510  062442  000000            2$:     .WORD   0
11511  062444  104401                    TYPE
11512  062446  001313                    .WORD   $CRLF
11513  062450  000755                    BR      PCTYP
11514
11515  062452  012600            ERT5:   MOV     (SP)+,R0                ;RESTORE R0.
11516  062454  000207                    RTS     PC                      ;AND RETURN.
11517
11518  062456  046106  040517  044524  EMSG: .ASCIZ /FLOATING POINT ERROR, STOPPED AT PC= /
11519  062464  043516  050040  044517
11520  062472  052116  042440  051122
11521  062500  051117  020054  052123
11522  062506  050117  042520  020104
11523  062514  052101  050040  036503
11524  062522  000040
11525                            .EVEN
11526
11527
11528
11529                                    .SBTTL  FPP SPURIOUS TRAP TO 244 HANDLER
11530                            ;;*****************************************************************
11531                            ;;*****************************************************************
11532                            ;*THIS ROUTINE HANDLES UNEXPECTED TRAPS TO THE FPP TRAP VECTOR AT 244.
11533                            ;*THE LAST FPP INSTRUCTION EXECUTED AND ITS ADDRESS HAS BEEN RECORDED
11534                            ;*THESE ALONG WITH THE FEC, FPS AND PC OF TRAP ARE REPORTED.
11535                            ;*
11536  062524  011637  001236    FPSPUR: MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
11537  062530  022626                    CMP     (SP)+,(SP)+             ;RESTORE SP.
11538  062532  170200                    STFPS   R0                      ;GET FPS
11539  062534  010037  001240        ;   MOV     R0,@#$TMP3
11540  062540  170300                    STST    R0                      ;GET FEC
11541  062542  010037  001242            MOV     R0,@#$TMP4
11542  062546  104001            1$:     ERROR   1               ;MOST LIKELY BAD FP1 CHIP
11543  062550  104413                    RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
11544                                                            ;SEE IF THE USER HAS EXPRESSED
11545                                                            ;THE DESIRE TO CHANGE THE SOFTWARE
11546                                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
11547                                                            ;THE USER TYPED CONTROL G?).
11548  062552  000137  057224            JMP     @#$EOP
11549
11550
```

```
11551                                   .SBTTL  CPU SPURIOUS TRAP TO 4 HANDLER
11552                                   ;;*****************************************************************
11553                                   ;;*****************************************************************
11554                                   ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 4.
11555                                   ;*
11556  062556  011637  001236          CPSPUR: MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
11557  062562  022626                          CMP     (SP)+,(SP)+
11558  062564  104004                  1$:     ERROR   4
11559  062566  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
11560                                                                  ;SEE IF THE USER HAS EXPRESSED
11561                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
11562                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
11563                                                                  ;THE USER TYPED CONTROL G?).
11564  062570  000137  057224                  JMP     @#$EOP
11565
11566
11567                                   .SBTTL  CPU SPURIOUS TRAP TO 10 HANDLER
11568                                   ;;*****************************************************************
11569                                   ;;*****************************************************************
11570                                   ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 10.
11571                                   ;*
11572  062574  011637  001236          CPTWO:  MOV     (SP),@#$TMP2            ;SAVE PC OF TRAP.
11573  062600  022626                          CMP     (SP)+,(SP)+
11574  062602  104002                  1$:     ERROR   2
11575  062604  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
11576                                                                  ;SEE IF THE USER HAS EXPRESSED
11577                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
11578                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
11579                                                                  ;THE USER TYPED CONTROL G?).
11580  062606  000137  057224                  JMP     @#$EOP
11581
11582
11583
11584
11585                                   .SBTTL  SET LOOP ON ERROR ADDRESS ROUTINE
11586                                   ;;*****************************************************************
11587                                   ;;*****************************************************************
11588                                   ;*
11589  062612  011637  001110          .LPER:  MOV     (SP),@#$LPERR
11590  062616  000002                          RTI
11591
11592
11593                                   .SBTTL  FLAG RESET AND CONSOLE TEST ROUTINE
11594                                   ;;*****************************************************************
11595                                   ;;*****************************************************************
11596                                   ;*THIS ROUTINE WILL BE CALLED AT THE END OF EACH TEST TO
11597                                   ;*RESET THE STACK, CLEAR THE FPS AND SEE IF THE USER HAS TYPED
11598                                   ;* CONTROL G ON THE TERMINAL. IF THE USER HAS TYPED CONTROL G AND
11599                                   ;*THERE IS NO PHYSICAL CONSOLE SWITCH REGISTER THEN THE CONTENTS
11600                                   ;*OF THE SOFTWARE SWITCH REGISTER WILL BE TYPED IN OCTAL ON THE
11601                                   ;*TELETYPE AND THE USER CAN MODIFY IT.
11602                                   ;*
11603  062620  023727  001140  177570  .RSET:  CMP     @#SWR,#177570           ;SEE IF THERE IS A PHYSICAL
11604                                                                  ;CONSOLE SWITCH REGISTER.
11605  062626  001001                          BNE     1$                      ;BRANCH IF NO.
11606  062630  104407                          CKSWR                           ;OTHERWISE TYPE THE CONTENTS
```

```
11607                                                                   ;OF THE PROGRAM VIRTUAL SWITCH REGISTER
11608                                                                   ;AND GIVE THE USER A CHANCE TO
11609                                                                   ;MODIFY IT.
11610  062632  012737  062524  000244  1$:     MOV     #FPSPUR,@#FPVECT
11611  062640  012737  062556  000004          MOV     #CPSPUR,@#ERRVECT
11612  062646  012737  062574  000010          MOV     #CPTWO,@#10
11613  062654  011600                          MOV     (SP),R0                 ;SAVE RETURN ADDRESS.
11614  062656  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK POINTER.
11615  062662  005004                          CLR     R4                      ;CLEAR THE FPS.
11616  062664  170104                          LDFPS   R4
11617  062666  000110                          JMP     (R0)                    ;RETURN.
11618
11619                                          .NLIST  BEX

                                        ;SPECIAL MESSAGES:

       062670  050200  053517  051105  POWERM: .ASCIZ  <CRLF>'POWER FAILURE. PROGRAM RESTARTING.'<CRLF>
       062735  000000                  NULL:   .BYTE   0
       062736  000011                  $TAB:   .ASCIZ  <TAB>
       062740  020040  000000          SPACE:  .ASCIZ  ' '


                                        ;ERROR MESSAGES:

       062743  000120  047522  040502  EM1:    .ASCIZ  /PROBABLY BAD FP1 CHIP/
       062771  000120  047522  040502  EM2:    .ASCIZ  /PROBABLY BAD HYBRID FP CHIP/
       063025  000120  047522  040502  EM3:    .ASCIZ  /PROBABLY BAD MMU CHIP/
       063053  000120  047522  040502  EM4:    .ASCIZ  /PROBABLY BAD MMU OR HYBRID FP CHIPS/
               000001                  .END
```

```
AAADON  040746          6994    7081#
AAA1    037744          6777#
AAA10   040362          6923#
AAA11   040420          6941#
AAA12   040456          6959#
AAA13   040514          6977#
AAA2    040002          6791#
AAA3    040040          6806#
AAA4    040076          6821#
AAA5    040134          6837#
AAA6    040172          6852#
AAA7    040230          6870#
AAA8    040266          6888#
AAA9    040324          6906#
AADAT0  023040          4227    4235    4257    4264    4269    4296    4303    4341    4347    4355    4361    4366    4372
                        4376#
AADONE  023140          4311    4331    4343    4349    4357    4363    4369    4375    4408#
AAERR0  022472          4219    4315#
AAERR1  022560          4239    4338#
AAERR2  022614          4241    4344#
AAERR3  022650          4273    4350#
AAERR4  022660          4275    4352#
AAERR5  022714          4277    4358#
AAERR6  022750          4307    4364#
AAERR7  023004          4309    4370#
AAER10  022566          4339#   4351
AAPAT0  023050          4220    4251    4286    4339    4345    4353    4359    4365    4371    4380#
AAPAT1  023060          4224    4254    4340    4346    4354    4360    4384#
AAPAT2  023070          4229    4263    4302    4338    4344    4388#
AAPAT3  023100          4234    4259    4350    4352    4358    4392#
AAPAT4  023110          4268    4396#
AAPAT5  023120          4290    4364    4370    4400#
AAPAT6  023130          4298    4367    4373    4404#
AA1     022112          4215#
AA10    022226          4247#
AA11    022256          4253    4255#
AA12    022260          4257#
AA13    022276          4261#   4278
AA14    022316          4266#   4276
AA15    022336          4271#   4274
AA16    022344          4272    4274#
AA17    022350          4267    4276#
AA2     022150          4223    4225#
AA20    022354          4262    4278#
AA21    022356          4282#
AA22    022410          4289    4291#
AA23    022412          4293#
AA24    022432          4300#   4310
AA25    022452          4305#   4308
AA26    022460          4306    4308#
AA27    022464          4301    4310#
AA3     022152          4227#
AA4     022170          4231#   4242
AA5     022210          4237#   4240
AA6     022216          4238    4240#
AA7     022224          4233    4242#
```

```
ABASE = 000000            271     312
ACDW1 = 000000            271     314
ACDW2 = 000000            271     315
ACPUOP= 000000            271     286
ADDW0 = 000000            271     316
ADDW1 = 000000            271     317
ADDW10= 000000            271     326
ADDW11= 000000            271     327
ADDW12= 000000            271     328
ADDW13= 000000            271     329
ADDW14= 000000            271     330
ADDW15= 000000            271     331
ADDW2 = 000000            271     318
ADDW3 = 000000            271     319
ADDW4 = 000000            271     320
ADDW5 = 000000            271     321
ADDW6 = 000000            271     322
ADDW7 = 000000            271     323
ADDW8 = 000000            271     324
ADDW9 = 000000            271     325
ADEVCT= 000000            271     277
ADEVM = 000000            271     313
ADONE   002444            524     532     545     558     564#
AENV  = 000000            271     282
AENVM = 000000            271     283
AERR1   002336            499     528#
AERR2   002350            500     535#
AERR3   002402            501     548#
AFATAL= 000000            271     274
AMADR1= 000000            271     299
AMADR2= 000000            271     303
AMADR3= 000000            271     306
AMADR4= 000000            271     309
AMAMS1= 000000            271     293
AMAMS2= 000000            271     301
AMAMS3= 000000            271     304
AMAMS4= 000000            271     307
AMSGAD= 000000            271     279
AMSGLG= 000000            271     280
AMSGTY  000000            271     273
AMTYP1= 000000            271     294
AMTYP2= 000000            271     302
AMTYP3= 000000            271     305
AMTYP4  000000            271     308
APASS   000000            271     276
APRIOR= 000000            271
APTCSU= 000040          11023   11275#
APTENV  000001          10923   11016   11231   11273#
APTSIZ= 000200            446   11272#
APTSPO= 000100          11018   11233   11274#
ASWREG= 000000            271     284
ATESTN= 000000            271     275
AUNIT = 000000            271     278
AUSWR = 000000            271     285
AVECT1= 000000            271     310
AVECT2= 000000            271     311
```

```
A1      002252          505#    523
A11     002252          506#    535     548
A12     00226€          511#    537     550
A2      002326          519     522#
BBBDON  041450          7186    7207    7215    7223    7236    7245#
BBBER1  041224          7107    7190#
BBBER2  041310          7120    7143    7180    7210#
BBBER3  041336          7124    7147    7184    7218#
BBBER4  041364          7165    7227#
BBBP1   041430          7109    7111    7135    7160    7238#
BBBP2   041440          7133    7156    7240#
BBB0    040754          7103#
BBB1    041010          7108    7113#
BBB2    041036          7127#
BBB3    041066          7132    7136#
BBB4    041114          7150#
BBB5    041152          7155    7162#   7167
BBB6    041160          7159    7167#
BBDATO  026102          4746    4769    4775    4800    4823    4829    4852    4874    4906    4915    4921    4930    4939#
BBDONE  026232          4884    4893    4902    4908    4917    4923    4932    4983#
BBER0   025?70          4738    4757    4786    4862    4883    4885#
BBER1   025630          4752    4903#
BBER10  025610          4810    4840    4894#
BBER11  025644          4905#   4911    4935    4938
BBER2   025666          4780    4909#
BBER3   025704          4782    4912#
BBER4   025740          4806    4918#
BBER40  025754          4920#   4926
BBER5   025774          4834    4924#
BBER6   026012          4836    4927#
BBER7   026046          4858    4933#
BBER8   026064          4880    4936#
BBPAT0  026112          4791    4802    4918    4919    4943#
BBPAT1  026122          4743    4766    4796    4819    4849    4871    4905    4914    4920    4929    4934    4947#
BBPAT2  026132          4741    4748    4903    4904    4951#
BBPAT3  026142          4815    4830    4924    4927    4955#
BBPAT4  026152          4764    4776    4909    4912    4959#
BBPAT5  026162          4847    4933    4963#
BBPAT6  026172          4869    4876    4936    4937    4967#
BBP10   026212          4771    4910    4913    4975#
BBP11   026222          4854    4979#
BBP7    026202          4825    4925    4928    4971#
BB1     024732          4734#
BB10    025106          4773#   4783
BB11    025126          4778#   4781
BB12    025136          4779    4781#
BB13    025144          4774    4783#
BB14    025156          4785    4788#
BB15    025206          4790    4797#
BB16    025232          4804#   4807
BB17    025242          4805    4807#
BB2     024770          4739    4744#
BB20    025254          4809    4812#
BB21    025304          4814    4820#
BB22    025330          4827#   4837
BB23    025350          4832#   4835
```

```
BB24    025360            4833    4835#
BB25    025366            4828    4837#
BB26    025400            4839    4842#
BB27    025430            4844    4850#
BB3     024774            4746#
BB30    025452            4856#   4859
BB31    025462            4857    4859#
BB32    025474            4861    4864#
BB33    025524            4866    4872#
BB34    025546            4878#   4881
BB35    025556            4879    4881#
BB4     025012            4750#   4753
BB5     025022            4751    4753#
BB6     025034            4756    4759#
BB7     025064            4763    4767#
BDONE   002546             594     614#
BERR    002502             591     596#
BERR1   002532             601     608#
BIT0  = 000001             146#
BIT00 = 000001             136#    146
BIT01 = 000002             135#    145
BIT02 = 000004             134#    144
BIT03 = 000010             133#    143
BIT04 = 000020             132#    142
BIT05 = 000040             131#    141
BIT06 = 000100             130#    140
BIT07 = 000200             129#    139
BIT08 = 000400             128#    138   10857
BIT09 = 001000             127#    137   10865   10934
BIT1  = 000002             145#
BIT10 = 002000             126#  10911
BIT11 = 004000             125#  10872
BIT12 = 010000             124#  10811
BIT13 = 020000             123#  10918
BIT14 = 040000             122#  10843
BIT15 = 100000             121#
BIT2  = 000004             144#
BIT3  = 000010             143#
BIT4  = 000020             142#
BIT5  = 000040             141#
BIT6  = 000100             140#
BIT7  = 000200             139#
BIT8  = 000400             138#
BIT9  = 001000             137#
BPTVEC= 000014             153#
B1      002456             582#    593
B2      002460             585#    598
B3      002476             593#    606     612
CCCDON  042460            7424    7512#
CCC1    041454            7265#
CCC10   042050            7376#
CCC11   042104            7388#
CCC12   042140            7401#
CCC13   042174            7413#
CCC2    041510            7277#
CCC3    041544            7289#
```

CCC4	041600	7302#											
CCC5	041634	7314#											
CCC6	041670	7326#											
CCC7	041724	7338#											
CCC8	041760	7351#											
CCC9	042014	7364#											
CCDATO	024566	4437	4443	4466	4472	4496	4502	4526	4532	4555	4561	4584	4590	4614
		4620	4629	4635	4641	4649	4665#							
CCDONE	024726	4602	4606	4610	4616	4622	4631	4637	4643	4652	4714#			
CCER0	024130	4454	4483	4572	4601	4603#								
CCER1	024164	4448	4611#											
CCER10	024476	4566	4653#											
CCER11	024514	4568	4656#											
CCER12	024532	4595	4659#											
CCER13	024550	4597	4662#											
CCER2	024222	4450	4617#											
CCER22	024236	4619#	4664											
CCER3	024260	4477	4623#											
CCER4	024276	4479	4626#											
CCER44	024312	4628#	4658											
CCER5	024334	4507	4632#											
CCER50	024200	4613#	4625	4655	466'									
CCER55	024350	4634#	4646											
CCER6	024370	4509	4638#											
CCER7	024424	4537	4644#											
CCER8	024442	4539	4647#											
CCER90	024146	4513	4543	4607#										
CCP0	024576	4432	4444	4461	4488	4550	4579	4613	4619	4628	4634	4640	4650	4669#
CCP1	024606	4463	44.3	4623	4626	4673#								
CCP10	024676	4528	4645	4648	4701#									
CCP11	024706	4557	4654	4657	4705#									
CCP12	024716	4709#												
CCP2	024616	4434	4439	4611	4612	4617	4618	4677#						
CCP3	024626	4503	4518	4552	4562	4653	4656	4681#						
CCP4	024636	4581	4586	4591	4659	4660	4662	4663	4685#					
CCP5	024646	4522	4533	4644	4647	4689#								
CCP6	024656	4492	4498	4632	4633	4638	4639	4693#						
CCP7	024666	4468	4624	4627	4697#									
CC1	023144	4427#												
CC10	023360	4475#	4478											
CC11	023370	4476	4478#											
CC12	023376	4471	4480#											
CC13	023410	4482	4485#											
CC14	023440	4487	4493#											
CC15	023464	4500#	4510											
CC16	023504	4505#	4508											
CC17	023514	4506	4508#											
CC18	023522	4501	4510#											
CC19	023534	4512	4515#											
CC2	023174	4431	4435#											
CC20	023564	4517	4523#											
CC21	023610	4530#	4540											
CC22	023630	4535#	4538											
CC23	023640	4536	4538#											
CC24	023646	4531	4540#											
CC25	023660	4542	4545#											

```
CC26    023710          4549    4553#
CC27    023732          4559#   4569
CC28    023752          4564#   4567
CC29    023762          4565    4567#
CC3     023216          4441#   4451
CC30    023770          4560    4569#
CC31    024002          4571    4574#
CC32    024032          4578    4582#
CC33    024054          4588#   4598
CC34    024074          4593#   4596
CC35    024104          4594    4596#
CC36    024112          4589    4598#
CC37    024124          4600    4602#
CC4     023236          4446#   4449
CC5     023246          4447    4449#
CC6     023254          4442    4451#
CC7     023266          4453    4456#
CC8     023316          4460    4464#
CC9     023340          4470#   4480
CDONE   003174           761#
CKSWR = 104407         10842   10907   10933   11425#  11606
CMPSUB  040556          6779    6793    6808    6823    6839    6854    6872    6890    6908    6925    6943    6961    6979
                        7020#
CMPTMP  040736          7060    7072    7076#
CPSPUR  062556           515     552     851    1227    1263    1434    1612    2452    2463    2540    2645    2882    2990
                        2993    3141    3144    3263    3401    6516    6529    6687   11556#  11611
CPTWO   062574           516     539   11572#  11612
CR    = 000015            61#  11062   11072
CRLF  = 000200            49#    62#     470   11033   11072   11619
C1      002562          635#
C15     002602          639     641#
C2      002624          653#
C25     002640          656     657#
C3      002664          668#
C35     002706          672     673#
C4      002732          684#
C45     002744          686     688#
C5      002770          699#
C55     003010          703     705#
C6      003032          716#
C65     003046          718     720#
C7      003072          731#
C75     003114          734     736#
C8      003140          747#
C85     003152          749     751#
DDDATO  030076          5009    5015    5039    5062    5085    5091    5114    5120    5143    5149    5173    5180    5201
                        5208    5215    5222    5229    5236    5243    5250    5257    5264    5271    5278   5282#
DDDDON  043414          7643    7740#
DDDONE  030226          5193    5197    5204    5211    5218    5225    5232    5239    5246    5253    5260    5267    5274
                        5281    5326#
DDD1    042464          7531#
DDD2    042540          7547#
DDD3    042614          7563#
DDD4    042670          7579#
DDD5    042744          7595#
DDD6    043020          7612#
```

```
DDD7    043074          7628#
DDER0   027310          5027    5050    5073    5102    5131    5161    5192    5194#
DDER1   027326          5020    5198#
DDER10  027744          5156    5261#
DDER11  030002          5185    5268#
DDER12  030040          5187    5275#
DDER2   027364          5022    5205#
DDER3   027422          5045    5212#
DDER4   027460          5068    5219#
DDER5   027516          5096    5226#
DDER6   027554          5098    5233#
DDER7   027612          5125    5240#
DDER8   027650          5127    5247#
DDER9   027706          5154    5254#
DDISP = 177570            68#     210     434
DDONE   003436          817     859#
DDP0    030106          5041    5064    5216    5223    5286#
DDP1    030116          5004    5006    5036    5057    5199    5200    5202    5206    5207    5213    5221    5290#
DDP2    030126          5034    5059    5214    5220    5294#
DDP3    030136          5080    5111    5228    5235    5241    5248    5298#
DDP4    030146          5016    5138    5170    5256    5263    5269    5276    5302#
DDP5    030156          5140    5168    5255    5262    5270    5277    5306#
DDP6    030166          5082    5109    5227    5234    5242    5249    5310#
DDP7    030176          5087    5116    5150    5181    5230    5237    5244    5251    5314#
DDP8    030206          5092    5121    5145    5176    5258    5265    5272    5279    5318#
DDP9    030216          5011    5209    5322#
DD1     026236          4999#
DD10    026436          5043#   5046
DD11    026446          5044    5046#
DD12    026464          5049    5052#
DD13    026514          5056    5060#
DD14    026536          5066#   5069
DD15    026546          5067    5069#
DD16    026564          5072    5075#
DD17    026614          5079    5083#
DD18    026636          5089#   5099
DD19    026656          5094#   5097
DD2     026266          5003    5007#
DD20    026666          5095    5097#
DD21    026674          5090    5099#
DD22    026706          5101    5104#
DD23    026736          5108    5112#
DD24    026760          5118#   5128
DD25    027000          5123#   5126
DD26    027010          5124    5126#
DD27    027016          5119    5128#
DD3     026310          5013#   5023
DD30    027030          5130    5133#
DD31    027060          5137    5141#
DD32    027102          5147#   5157
DD33    027122          5152#   5155
DD34    027132          5153    5155#
DD35    027140          5148    5157#
DD36    027156          5160    5163#
DD37    027206          5167    5171#
DD38    027230          5178#   5188
```

```
DD39    027250          5183#   5186
DD4     026330          5018#   5021
DD40    027260          5184    5186#
DD41    027266          5179    5188#
DD42    027304          5191    5193#
DD5     026340          5019    5021#
DD6     026346          5014    5023#
DD7     026364          5026    5029#
DD8     026414          5033    5037#
DERR1   003316           793     821#
DERR2   003412           792     849#
DISPLA  001142           210#    434*    442*    10887*  10910*
DISPRE  000174           178#    442
DIVDSU  043154          7533    7549    7565    7581    7597    7614    7630    7673#
DIVDT   043404          7692    7701    7707    7719    7737#
DIVFSU  042234          7267    7279    7291    7304    7316    7328    7340    7353    7366    7378    7390    7403    7415
                        7452#
DIVFT   042450          7469    7478    7509#
DOAGIN  057332         10793   10803#
DPAT3   013316          2687#
DSWR  = 177570            67#    209     433
D1      003222           795#    813     819
D10     003424           850     853#
D2      003246           798*    800     801#
D3      003250           802#    821
D4      003254           804#
D5      003266           810#    841     847     857
D6      003302           811     815#
D7      003312           816     818#
D8      003362           835#    844     849
D9      003374           840     843#
EDONE   003576           893     899     905     918#
EEDATO  030720          5347    5353    5378    5384    5405    5412    5419    5426    5430#
EEDONE  031002          5397    5401    5408    5415    5422    5429    5455#
EEEDON  044424          7918    8006#
EEER0   030512          5366    5396    5398#
EEER1   030530          5359    5402#
EEER2   030566          5361    5409#
EEER3   030624          5389    5416#
EEER4   030662          5391    5423#
EEE1    043420          7760#
EEE10   044014          7871#
EEE11   044050          7883#
EEE12   044104          7895#
EEE13   044140          7907#
EEE2    043454          7772#
EEE3    043510          7785#
EEE4    043544          7797#
EEE5    043600          7809#
EEE6    043634          7822#
EEE7    043670          7835#
EEE8    043724          7847#
EEE9    043760          7859#
EEP0    030730          5349    5380    5406    5413    5420    5427    5434#
EEP1    030742          5342    5344    5403    5404    5410    5411    5439#
EEP2    030752          5355    5443#
```

```
EEP3    030762          5373    5375    5417    5418    5424    5425    5447#
EEP4    030772          5385    5451#
EERR0   003514           888     895#
EERR1   003532           892     901#
EERR2   003546           877     907#
EE1     030232          5337#
EE10    030452          5387#   5390
EE11    030462          5388    5390#
EE12    030470          5383    5392#
EE13    030506          5395    5397#
EE2     030262          5341    5345#
EE3     030304          5351#   5362
EE4     030324          5357#   5360
EE5     030334          5358    5360#
EE6     030342          5352    5362#
EE7     030360          5365    5368#
EE8     030410          5372    5376#
EE9     030432          5382#   5392
EMSG    062456         11497   11518#
EMTVEC= 000030           156#    403*    404*
EM1     062743           352   11619#
EM2     062771           353   11619#
EM3     063025           354   11619#
EM4     063053           355   11619#
ERRVEC= 000004           149#    431     432*    443*    501*    515*    792*    955*    995*   1338*   1522*   2342*   2373*
                        2507*   2612*   2821*   2931*   3082*   3231*   3358*   6510*   6516*   6529*  10848   10849*  10851*
                       10854*  11611*
ERTYPE  062342         10920   11487#
ERT5    062452         11501   11508   11515#
E1      003452           879#
E2      003466           882#
E3      003466           881     883#
E4      003470           884#    907
FDATI0  005372           947     971    1056    1057    1066    1278#
FDATI1  005374          1279#
FDATI2  005376          1280#
FDATI3  005400          1216    1281#
FDATI4  005402           954     963    1061    1071    1209    1282#
FDATI5  005404          1225    1283#
FDATI6  005406          1284#
FDATI7  005410          1285#
FDATO0  005414           990    1011    1104    1125    1287#
FDATO1  005416          1288#
FDATO2  005420          1289#
FDATO3  005422          1252    1290#
FDATO4  005424           994    1003    1099    1109    1194    1244    1291#
FDATO5  005426          1131    1146    1261    1292#
FDATO6  005430          1162    1177    1293#
FDATO7  005432          1126    1195    1294#
FDONE   005456          1051    1059    1079    1083    1094    1116    1121    1129    1144    1160    1175    1190    1198
                        1204    1223    1233    1240    1259    1269    1276    1306#
FERR0   004200           976    1053#
FERR1   004236           965    1061#
FERR10  005016          1032    1044    1150    1181    1192#
FERR11  005052           982    1050    1200#
FERR2   004334           969    1086#
```

```
FERR20  005072   955  1206#
FERR21  005210  1214  1235#
FERR25  005232   995  1242#
FERR26  005350  1250  1271#
FERR3   004374  1005  1099#
FERR4   004366  1009  1096#
FERR5   004472  1016  1020  1024  1028  1123#
FERR6   004526  1036  1131#
FERR7   004662  1040  1162#
FER2    004340  1088# 1097
FFDATO  031300  5483  5506  5527  5535  5541#
FFDONE  031360  5517  5522  5530  5538  5567#
FFER0   031170  5493  5516  5519#
FFER1   031204  5512  5524#
FFER2   031242  5489  5532#
FFFDON  045154  8090  8185#
FFF1    044430  8026#
FFF2    044504  8043#
FFF3    044560  8059#
FFF4    044634  8075#
FFP0    031310  5501  5526  5546#
FFP1    031320  5503  5525  5550#
FFP2    031330  5478  5534  5554#
FFP3    031340  5480  5533  5558#
FFP4    031350  5485  5508  5528  5536  5562#
FF1     031006  5473#
FF10    031156  5511  5513#
FF11    031166  5515  5517#
FF2     031036  5477  5481#
FF3     031060  5487# 5490
FF4     031066  5488  5490#
FF5     031076  5492  5496#
FF6     031126  5500  5504#
FF7     031150  5510# 5513
FPSPUR  062524   512   823   909  2396  2415  3625  3667  4319  5691  5845  5975  6042  6110
                6139  6210  6289  7172  7195  8408  8680  8877  9146 11536# 11610
FPVECT= 000244    45# 499*  512*  793*  877* 2337* 2371* 3511* 3608* 4219* 4738* 5635* 5691*
                5937* 5962* 6004* 6029* 6072* 6097* 7107* 7159* 8334* 8606* 8864* 9133* 11610*
FXDAT0  005436   948   972  1012  1054  1055  1127  1296#
FXDAT1  005440  1297#
FXDAT2  005442  1298#
FXDAT3  005444  1299#
FXDAT4  005446  1196  1300#
FXDAT5  005450  1301#
FXDAT6  005452  1302#
FXDAT7  005454  1303#
F1      003602   941#
F10     004014   986   999#  1249
F11     004016  1000# 1096
F12     004034  1004  1007#
F13     004066  1015  1018#
F135    004046  1008  1011#
F14     004076  1019  1022#
F15     004106  1023  1026#
F16     004116  1027  1030#
F17     004126  1031  1034#
```

F2	003634	951#	952											
F20	004136	1035	1038#											
F21	004146	1039	1042#											
F22	004156	1043	1046#											
F23	004174	1049	1051#											
F3	003654	943	959#											
F4	003656	960#	1086	1213										
F5	003674	964	967#											
F6	003750	981	984#											
F7	003774	991#	992											
GAND0	011022	2082#												
GAND1	011024	2083#												
GAND2	011026	2084#												
GAND3	011030	2085#												
GCMP	010746	1700	1728	1756	1785	1813	1841	1869	1897	1925	1953	1981	2009	2056#
GDAT00	011042	1691	1719	1747	1776	1804	1832	1860	1888	1916	1944	1972	2000	2043
		2049	2057	2092#										
GDAT01	011044	2093#												
GDAT02	011046	2094#												
GDAT03	011050	2095#												
GDONE	011052	2026	2098#											
GFLAG1	010776	1702	1704*	1730	1732*	1758	1760*	1787	1789*	1815	1817*	1843	1845*	1871
		1873*	1899	1901*	1927	1929*	1955	1957*	1983	1985*	2011	2013*	2029	2069#
GFLAG2	011000	2070#												
GGDAT0	036100	5943	5968	5979	6010	6035	6046	6078	6103	6114	6145	6149	6159	6169
		6179	6191	6201	6216	6220	6230	6240	6250	6260	6270	6280	6295	6300
		6310	6320	6330	6340	6350	6360	6365#						
GGDONE	036220	6133	6152	6162	6172	6182	6194	6204	6223	6233	6243	6253	6263	6273
		6283	6303	6313	6323	6333	6343	6353	6363	6407#				
GGER0	034376	5937	6135#											
GGER1	034474	5949	6154#	6184										
GGER10	035206	6037	6245#											
GGER11	035254	6052	6255#											
GGER12	035322	6057	6265#											
GGER13	035370	6064	6275#											
GGER14	035436	6072	6285#											
GGER15	035534	6084	6305#											
GGER16	035602	6089	6315#											
GGER17	035650	6105	6325#											
GGER18	035716	6120	6335#											
GGER19	035764	6131	6345#											
GGER2	034542	5954	6164#											
GGER20	036032	6125	6355#											
GGER3	034610	5970	6174#											
GGER4	034656	5985	6184#											
GGER5	034660	5996	6186#											
GGER6	034726	5990	6196#											
GGER7	034774	6004	6206#											
GGER8	035072	6016	6225#											
GGER9	035140	6021	6235#											
GGGDON	052236	9502	9634#											
GGG1	050564	9265#												
GGG10	051334	9420#												
GGG11	051404	9437#												
GGG12	051454	9454#												
GGG13	051524	9471#												

```
GGG14   051574          9488#
GGG2    050634          9282#
GGG3    050704          9299#
GGG4    050754          9316#
GGG5    051024          9333#
GGG6    051074          9350#
GGG7    051144          9368#
GGG8    051214          9386#
GGG9    051264          9403#
GGP1    036110          6298    6370#
GGP2    036120          6005    6030    6073    6098    6219    6229    6239    6249    6259    6269    6279    6308    6318
                        6328    6338    6348    6358    6374#
GGP3    036130          6007    6032    6218    6228    6238    6248    6258    6268    6278    6299    6378#
GGP4    036140          6382#
GGP5    036150          5938    5940    5963    5965    6147    6148    6157    6158    6167    6168    6177    6178    6189
                        6190    6199    6200    6386#
GGP6    036160          5945    5981    6012    6080    6150    6160    6170    6180    6192    6202    6221    6231    6241
                        6301    6311    6321    6390#
GGP7    036170          6048    6251    6261    6271    6281    6395#
GGP8    036200          6075    6100    6309    6319    6329    6339    6349    6359    6399#
GGP9    036210          6116    6331    6341    6351    6361    6403#
GG1     033306          5932#
GG10    033602          5995    5999#
GG11    033640          6003    6008#
GG12    033662          6014#   6017
GG13    033672          6015    6017#
GG14    033710          6020    6024#
GG15    033746          6028    6033#
GG16    033764          6029    6038#
GG17    034032          6050#   6053
GG18    034042          6051    6053#
GG19    034076          6063    6067#
GG2     033344          5936    5941#
GG20    034134          6071    6076#
GG21    034156          6082#   6085
GG22    034166          6083    6085#
GG23    034204          6088    6092#
GG24    034242          6096    6101#
GG25    034260          6097    6106#
GG26    034326          6118#   6121
GG27    034336          6119    6121#
GG28    034372          6130    6133#
GG3     033366          5947#   5950
GG4     033376          5948    5950#
GG5     033414          5953    5957#
GG6     033452          5961    5966#
GG7     033470          5962    5971#
GG8     033536          5983#   5986
GG9     033546          5984    5986#
GNS   = ****** U          177     469   11417   11418   11419   11420   11421   11423   11425   11426   11427   11428   11429
                        11430
GOR0    011032          2087#
GOR1    011034          2088#
GOR2    011036          2089#
GOR3    011040          2090#
GPAT00  011002          1690    1746    1803    1859    1915    1971    2039    2072#
```

GPAT01	011004	2073#												
GPAT02	011006	2074#												
GPAT03	011010	2075#												
GPAT10	011012	1718	1775	1831	1887	1943	1999	2034	2077#					
GPAT11	011014	2078#												
GPAT12	011016	2079#												
GPAT13	011020	2080#												
GRESET	010726	1712	1740	1768	1797	1825	1853	1881	1909	1937	1965	1993	2021	2049#
GSETUP	010650	1693	1721	1749	1778	1806	1834	1862	1890	1918	1946	1974	2002	2029#
GS1	010700	2039#	2053											
GTSWR =	104406	464	11423#											
G1	007134	1689	1695#	1714										
G10	007406	1759	1763#											
G11	007410	1762	1764#											
G12	007464	1774	1780#	1799										
G13	007516	1788	1792#											
G14	007520	1791	1793#											
G15	007574	1802	1808#	1827										
G16	007626	1816	1820#											
G17	007630	1819	1821#											
G2	007166	1703	1707#											
G20	007704	1830	1836#	1855										
G21	007736	1844	1848#											
G22	007740	1847	1849#											
G23	010014	1858	1864#	1883										
G24	010046	1872	1876#											
G25	010050	1875	1877#											
G26	010124	1886	1892#	1911										
G27	010156	1900	1904#											
G3	007170	1706	1708#											
G30	010160	1903	1905#											
G31	010234	1914	1920#	1939										
G32	010266	1928	1932#											
G33	010270	1931	1933#											
G34	010344	1942	1948#	1967										
G35	010376	1956	1960#											
G36	010400	1959	1961#											
G37	010454	1970	1976#	1995										
G4	007244	1717	1723#	1742										
G40	010506	1984	1988#											
G41	010510	1987	1989#											
G42	010564	1998	2004#	2023										
G43	010616	2012	2016#											
G44	010620	2015	2017#											
G5	007276	1731	1735#											
G6	007300	1734	1736#											
G7	007354	1745	1751#	1770										
HADR	011602	2249*	2257	2278#										
HA1R	011656	2227	2251	2260	2292#									
HA1W	011606	2117	2127	2154	2250	2268	2281#							
HA2R	011666	2231	2294#											
HA2W	011616	2131	2167	2283#										
HA3R	011676	2235	2296#											
HA3W	011626	2135	2180	2285#										
HA4R	011706	2239	2298#											
HA4W	011636	2139	2193	2287#										

```
HA5R    011716          2243    2300#
HA5W    011646          2143    2206    2289#
HCLR    011532          2123    2225    2260#
HCLR1   011542          2262#   2263
HCMP    011474          2149    2161    2174    2187    2200    2213    2249#
HCMP1   011514          2253#   2256
HCMP2   011524          2254    2256#
HDAT1   011726          2118    2303#
HDAT2   011736          2305#
HDAT3   011746          2307#
HDAT4   011756          2309#
HDAT5   011766          2311#
HDONE   011776          2219    2275    2314#
HERROR  011550          2255    2267#
HFLAG   011604          2116*   2217    2221*   2279#
HHDATO  033132          5606    5612    5641    5666    5672    5697    5720    5738    5747    5756    5765    5774    5783
                        5792    5801    5810    5819    5828    5837    5855#
HHDONE  033302          5732    5741    5750    5759    5768    5777    5786    5795    5804    5813    5822    5831    5840
                        5854    5907#
HHER0   032152          5617    5733#
HHER00  032266          5623    5751#
HHER1   032220          5619    5742#
HHER10  033014          5731    5832#
HHER2   032334          5647    5760#
HHER3   032402          5652    5769#
HHER4   032450          5677    5778#
HHER5   032516          5679    5787#
HHER6   032564          5684    5796#
HHER7   032632          5703    5805#
HHER8   032700          5708    5814#
HHER9   032746          5726    5823#
HHHDON  054344          9956    10096#
HHH1    052242          9655#
HHH10   053342          9867#
HHH11   053442          9890#
HHH12   053542          9913#
HHH13   053642          9936#
HHH2    052342          9678#
HHH3    052442          9701#
HHH4    052542          9724#
HHH5    052642          9747#
HHH6    052742          9770#
HHH7    053042          9794#
HHH8    053142          9818#
HHH9    053242          9841#
HHP0    033142          5601    5737    5746    5755    5859#
HHP1    033152          5603    5736    5745    5754    5863#
HHP10   033262          5699    5811    5820    5899#
HHP11   033272          5722    5829    5838    5903#
HHP2    033162          5608    5739    5748    5757    5867#
HHP3    033172          5613    5871#
HHP4    033202          5643    5673    5766    5775    5875#
HHP5    033212          5636    5715    5717    5764    5773    5826    5827    5835    5836    5879#
HHP6    033222          5638    5763    5772    5883#
HHP7    033232          5668    5784    5793    5802    5887#
HHP8    033242          5661    5692    5694    5782    5791    5800    5808    5809    5817    5818    5891#
```

```
HHP9    033252          5663    5781    5790    5799    5895#
HHTRAP  033062          5635    5841#
HH1     031362          5596#
HH10    031574          5646    5648#
HH11    031612          5651    5655#
HH12    031642          5660    5664#
HH13    031664          5670#   5680
HH14    031704          5675#   5678
HH15    031714          5676    5678#
HH16    031722          5671    5680#
HH17    031740          5683    5686#
HH18    031776          5690    5695#
HH19    032020          5701#   5704
HH2     031412          5600    5604#
HH20    032030          5702    5704#
HH21    032046          5707    5710#
HH22    032076          5714    5718#
HH23    032120          5724#   5727
HH24    032130          5725    5727#
HH25    032146          5730    5732#
HH3     031434          5610#   5620
HH4     031454          5615#   5618
HH5     031464          5616    5618#
HH6     031472          5611    5620#
HH7     031504          5622    5629#
HH8     031542          5634    5639#
HH9     031564          5645#   5648
HSTD    011416          2147    2160    2173    2186    2199    2212    2225#
HT    = 000011            59#   11031   11072
HXDATO  037630          6437    6444    6478    6487    6533    6553    6562    6590    6621    6651    6657    6672    6679
                        6708    6716#
HXDONE  037740          6635    6642    6647    6653    6659    6664    6669    6674    6681    6692    6701    6710    6756#
HXER1   037216          6434    6639#
HXER10  037506          6519    6694#
HXER11  037540          6566    6703#
HXER12  037560          6568    6706#
HXER13  037610          6597    6628    6712#
HXER2   037256          6448    6539    6649#
HXER22  037272          6651#   6705    6714
HXER3   037306          6450    6655#
HXER33  037322          6657#
HXER4   037336          6603    6634    6661#
HXER5   037236          6474    6644#
HXER6   037372          6491    6670#
HXER66  037406          6672#
HXER7   037422          6493    6676#
HXER8   037354          6455    6500    6666#
HXER9   037452          6510    6683#
HXP1    037640          6427    6462    6580    6585    6593    6616    6624    6712    6713    6721#
HXP2    037650          6429    6432    6443    6465    6472    6530    6639    6644    6649    6655    6670    6676    6726#
HXP3    037660          6730#
HXP4    037670          6550    6703    6706    6734#
HXP5    037700          6556    6704    6707    6738#
HXP6    037710          6527    6548    6612    6742#
HXP7    037720          6439    6486    6535    6561    6650    6656    6747#
HXP8    037730          6481    6671    6677    6751#
```

HX1	036224	6423#							
HX10	036432	6473	6476#						
HX11	036454	6483#	6495						
HX12	036474	6489#	6492						
HX13	036504	6490	6492#						
HX14	036512	6484	6495#						
HX15	036530	6499	6504#						
HX16	036556	6509	6512#						
HX165	036562	6513#	6685	6696					
HX17	036610	6518	6521#						
HX18	036646	6526	6531#						
HX19	036666	6537#	6540						
HX2	036254	6430	6431#						
HX20	036676	6538	6540#						
HX21	036700	6543#							
HX22	036730	6547	6551#						
HX23	036750	6558#	6571						
HX24	036770	6564#	6567						
HX25	037000	6565	6567#						
HX26	037006	6559	6571#						
HX27	037010	6575#							
HX28	037042	6584	6586#						
HX29	037064	6595#	6598						
HX3	036270	6433	6435#						
HX30	037074	6596	6598#						
HX31	037112	6602	6607#						
HX32	037142	6615	6617#						
HX33	037164	6626#	6629						
HX34	037174	6627	6629#						
HX35	037212	6633	6635#						
HX4	036310	6441#	6451						
HX5	036330	6446#	6449						
HX6	036340	6447	6449#						
HX7	036346	6442	6451#						
HX8	036364	6454	6457#						
HX9	036416	6466	6470#						
H1	011060	2116#							
H10	011322	2196#	2201						
H11	011354	2209#	2214						
H12	011406	2218	2221#						
H2	011100	2120#	2121						
H3	011110	2125#	2222						
H4	011162	2147#							
H5	011204	2157#	2162						
H6	011236	2170#	2175						
H7	011270	2183#	2188						
IDATI0	006274	1325	1331	1348	1376	1381	1409	1445	1486#
IDATI1	006276	1487#							
IDATI2	006300	1406*	1488#						
IDATI3	006302	1407*	1489#						
IDATO0	006264	1344	1347	1403	1408	1446	1481#		
IDATO1	006266	1482#							
IDATO2	006270	1353	1358	1414	1483#				
IDATO3	006272	1417	1484#						
IDONE	006304	1423	1441	1448	1456	1468	1491#		
IERR0	006044	1338	1426#						

```
IERR1   006126          1362    1416    1444#
IERR2   006146          1366    1451#
IERR25  006170          1454#   1460
IERR3   006220          1398    1463#
IERR4   006174          1419    1457#
IIIDON  046014          8273    8443#
IIII    045160          8206#
IIII2   045224          8223#
IIII3   045270          8241#
IIII4   045334          8258#
IOTVEC= 000020           154#    401*    402*
IPAT10  006244          1326    1375    1471#
IPAT11  006246          1472#
IPAT12  006250          1414    1473#
IPAT13  006252          1417    1474#
IPAT20  006254          1334    1384    1476#
IPAT21  006256          1477#
IPAT22  006260          1478#
IPAT23  006262          1479#
I1      005464          1323#
I10     005640          1356    1365#
I105    005646          1351    1368#
I106    005642          1364    1366#
I11     005650          1372#
I12     005652          1375#
I13     005666          1378#   1379
I14     005722          1388    1391#   1464
I15     005724          1392#   1430
I16     005726          1393#   1432
I17     005744          1397    1400#
I2      005502          1328#   1329
I20     006004          1411#   1421
I21     006020          1416#   1418
I22     006024          1415    1417#
I23     006040          1412    1421#
I3      005542          1339    1340#
I4      005544          1341#   1426
I5      005546          1342#   1428
I6      005572          1350#   1368
I7      005606          1355#   1365
JBUF0   012766          2516    2565#
JBUF1   012770          2566#
JBUF2   012772          2567#
JBUF3   012774          2568#
JDATI0  012776          2505    2520    2528    2552    2560    2570#
JDATI1  013000          2571#
JDATI2  013002          2572#
JDATI3  013004          2573#
JDAT00  013006          2513    2521    2561    2575#
JDAT0   013016          2502    2580#
JDAT01  013010          2576#
JDAT02  013012          2577#
JDAT03  013014          2578#
JDAT1   013020          2581#
JDAT2   013022          2582#
JDAT3   013024          2583#
```

```
JDONE   013026          2532    2547    2554    2563    2586#
JERR0   012646          2507    2536#
JERR1   012714          2518    2530    2549#
JERR2   012740          2525    2558#
JJJDON  046754          8547    8715#
JJJ1    046020          8465#
JJJ2    046104          8486#
JJJ3    046170          8507#
JJJ4    046254          8529#
J1      012540          2499#
J10     012666          2537    2539    2542#
J2      012564          2509#   2550    2559
J3      012566          2510#   2536
J4      012570          2511#   2538
J5      012624          2523#   2526
J6      012632          2524    2526#
J7      012644          2529    2532#
KBUF0   013270          2610    2621    2633    2674#
KBUF1   013272          2675#
KBUF2   013274          2676#
KBUF3   013276          2677#
KDATI0  013260          2625    2656    2664    2669#
KDATI1  013262          2670#
KDATI2  013264          2671#
KDATI3  013266          2672#
KDATO0  013300          2618    2626    2665    2679#
KDATO1  013302          2680#                   ;
KDATO2  013304          2681#
KDATO3  013306          2682#
KDONE   013320          2637    2651    2658    2667    2689#
KERR0   013142          2612    2641#
KERR1   013206          2623    2635    2653#
KERR2   013232          2630    2662#
KKKDON  047616          8806    8974#
KKK1    046760          8740#
KKK3    047024          8758#
KKK4    047070          8775#
KKK5    047134          8792#
KPAT0   013310          2607    2684#
KPAT1   013312          2685#
KPAT2   013314          2686#
K1      013034          2604#
K10     013162          2642    2644    2647#
K2      013060          2614#   2654    2663
K3      013062          2615#   2641
K4      013064          2616#   2643
K5      013120          2628#   2631
K6      013126          2629    2631#
K7      013140          2634    2637#
LDATI0  013620          2713    2720    2739    2759    2771    2786#
LDATI1  013622          2787#
LDATI2  013624          2733    2737*   2753    2788#
LDATI3  013626          2738*   2789#
LDATO0  013632          2730    2740    2760    2765    2791#
LDATO1  013634          2772    2792#
LDATO2  013636          2793#
```

```
LDAT03  013640          2794#
LDONE   013642          2748    2755    2762    2774    2796#
LERR1   013454          2735    2750#
LERR2   013526          2745    2764#
LERR3   013500          2757#   2768
LF    = 000012            60#   11066   11072
LLLDON  050560          9076    9243#
LLL1    047622          8995#
LLL2    047706          9016#
LLL3    047772          9037#
LLL4    050056          9058#
LOOP    002160           472#   10820
LPAT10  013576          2710    2776#
LPAT11  013600          2777#
LPAT12  013602          2778#
LPAT13  013604          2779#
LPAT20  013606          2714    2764    2781#
LPAT21  013610          2782#
LPAT22  013612          2783#
LPAT23  013614          2784#
LPERR = 104414           497     579     633     652     667     683     698     715     730     746     790     876     942
                         985    1321    1373    1507    1550    1692    1720    1748    1777    1805    1833    1861    1889
                        1917    1945    1973    2001    2114    2332    2365    2497    2602    2705    3513    3535    3557
                        3576    3606    3723    3751    3780    3808    3865    3889    3913    3937    4029    4132    4216
                        4248    4283    4428    4457    4486    4516    4546    4575    4735    4760    4789    4813    4843
                        4865    5000    5030    5053    5076    5105    5134    5164    5338    5369    5474    5497    5597
                        5630    5656    5687    5711    5933    5958    6000    6025    6068    6093    6424    6458    6505
                        6522    6544    6576    6608    6778    6792    6807    6822    6838    6853    6871    6889    6907
                        6924    6942    6960    6978    7102    7128    7151    7266    7278    7290    7303    7315    7327
                        7339    7352    7365    7377    7389    7402    7414    7532    7548    7564    7580    7596    7613
                        7629    7761    7773    7786    7798    7810    7823    7836    7848    7860    7872    7884    7896
                        7908    8027    8044    8060    8076    8207    8224    8242    8259    8466    8487    8508    8530
                        8741    8759    8776    8793    8996    9017    9038    9059    9266    9283    9300    9317    9334
                        9351    9369    9387    9404    9421    9438    9455    9472    9489    9656    9679    9702    9725
                        9748    9771    9795    9819    9842    9868    9891    9914    9937   10118   10136   10154   10173
                       10192   10361   10385   10409   10433   10609   10673   11430#
L1      013326          2707#
L2      013366          2724#   2751    2758    2770
L3      013370          2725#
L4      013372          2727#
L5      013442          2743#   2746
L6      013450          2744    2746#
MDAT00  014162          2836    2839    2849    2859    2875    2901#
MDAT01  014164          2902#
MDAT02  014166          2903#
MDAT03  014170          2904#
MDONE   014172          2847    2861    2871    2878    2889    2906#
MERR0   014016          2832    2863#
MERR1   014054          2842    2853    2873#
MERR2   013770          2856#
MERR3   014102          2821    2880#
MMMDON  055272         10206   10339#
MMM1    054350         10117#
MMM2    054422         10135#
MMM3    054474         10153#
MMM4    054546         10172#
```

```
MMM5    054620  10191#
MODDD0  056300  10507  10520  10527  10587#
MODDD1  056310  10509  10521  10535  10558  10590#
MODDOV  055712  10362  10386  10410  10434  10488#
MODDSU  053746   9657   9680   9703   9726   9749   9772   9796   9820   9843   9869   9892   9915   9938
                9992#
MODDT0  054324  10010  10023  10028  10051  10090#
MODDT1  054334  10012  10024  10036  10068  10093#
MODFD0  055252  10261  10274  10281  10333#
MODFD1  055262  10263  10275  10287  10336#
MODFOV  054676  10119  10137  10155  10174  10193  10242#
MODFSU  051650   9267   9284   9301   9318   9335   9352   9370   9388   9405   9422   9439   9456   9473
                9490   9538#
MODFT0  052206   9556   9569   9574   9625#
MODFT1  052216   9558   9570   9580   9628#
MODP1   052226   9543   9631#  9997   10247  10493
MPAT10  014142   2817   2891#
MPAT11  014144   2892#
MPAT12  014146   2893#
MPAT13  014150   2894#
MPAT20  014152   2858   2876   2896#
MPAT21  014154   2897#
MPAT22  014156   2898#
MPAT23  014160   2899#
MULDSU  044714   8028   8045   8061   8077   8118#
MULDT   045144   8137   8146   8152   8164   8182#
MULFSU  044200   7762   7774   7787   7799   7811   7824   7837   7849   7861   7873   7885   7897   7909
                7946#
MULFT   044414   7963   7972   8003#
MULT    062422  11496  11502#
M1      013646   2814#
M15     013666   2823#  2857   2869   2874   2886
M2      013672   2826#  2863   2885
M3      013674   2827#
M4      013676   2828#
M5      013732   2841   2843#
M6      013736   2844#  2846
M7      013746   2845   2849#
M8      013756   2851#  2854
M9      013766   2852   2854#
NDATI0  014670   2973   3030   3041   3051#
NDATI1  014672   3052#
NDATI2  014674   3053#
NDATI3  014676   3054#
NDATO0  014626   2938   2955   2962   2972   3029   3034#
NDATO1  014630   3035#
NDATO2  014632   3036#
NDATO3  014634   3037#
NDONE   014700   2978   2999   3007   3013   3025   3032   3056#
NERR0   014370   2931   2986#
NERR1   014470   2946   3008#
NERR10  014422   2987   2995#
NERR11  014434   2992   3001#
NERR2   014524   2950   3014#
NERR20  014476   3009#  3015   3017   3019
NERR3   014534   2960   3016#
```

```
NERR4   014544          2968    3018#
NERR5   014554          2970    3021#
NERR6   014600          2980    3027#
NNNDON  056320         10453   10593#
NNN1    055276         10360#
NNN2    055400         10384#
NNN3    055502         10408#
NNN4    055604         10432#
NPAT10  014660          2926    3046#
NPAT11  014662          3047#
NPAT12  014664          3048#
NPAT13  014666          3049#
NPAT20  014646          2929    2941    2944    2948    2952    2963    2991    3022    3041#
NPAT21  014650          3042#
NPAT22  014652          3043#
NPAT23  014654          3044#
NULL    062735          1206    1242   11619#
N1      014176          2923#
N10     014330          2965#   2967
N11     014340          2953    2966    2970#
N12     014342          2942    2972#
N13     014356          2975#   2977
N14     014366          2976    2980#
N2      014222          2934#   3011    3023    3028
N3      014224          2935#   2988
N4      014226          2936#   2986
N5      014244          2944#
N6      014254          2945    2948#
N7      014264          2949    2952#
N8      014302          2957#   2959
N9      014314          2958    2962#
ODATI0  015374          3124    3183    3193    3204#
ODATI1  015376          3205#
ODATI2  015400          3206#
ODATI3  015402          3207#
ODATO0  015332          3090    3113    3123    3182    3187#
ODATO1  015334          3106    3188#
ODATO2  015336          3189#
ODATO3  015340          3190#
ODONE   015404          3129    3150    3158    3165    3178    3185    3209#
OERR0   015074          3082    3137#
OERR1   015174          3098    3143    3160#
OERR10  015126          3138    3146#
OERR11  015140          3152#
OERR2   015230          3102    3166#
OERR20  015202          3161#   3167    3169    3171
OERR3   015240          3111    3168#
OERR4   015250          3119    3170#
OERR5   015260          3121    3174#
OERR6   015304          3131    3180#
OPAT10  015364          3077    3199#
OPAT11  015366          3200#
OPAT12  015370          3201#
OPAT13  015372          3202#
OPAT20  015350          3093    3175    3193#
OPAT21  015352          3080    3096    3100    3104    3114    3142    3194#
```

```
OPAT22  015354            3195#
OPAT23  015356            3196#
OPAT24  015360            3197#
OVDNTT  046744            8601    8617    8622    8648    8688    8712#
OVDTT   050550            9128    9153    9158    9198    9231    9240#
OVFNTT  046004            8329    8345    8350    8376    8416    8440#
OVFTT   047606            8858    8884    8889    8929    8962    8971#
OVUNDN  046344            8467    8488    8509    8531    8587#
OVUNDT  050146            8997    9018    9039    9060    9114#
OVUNFN  045404            8208    8225    8243    8260    8315#
OVUNFT  047204            8742    8760    8777    8794    8844#
01      014704            3074#
010     015034            3116#   3118
011     015044            3117    3121#
012     015046            3094    3123#
013     015062            3126#   3128
014     015072            3127    3131#
02      014730            3086#   3163    3181
03      014732            3087#   3139
04      014734            3088#   3137    3176
05      014752            3096#
06      014762            3097    3100#
07      014772            3101    3104#
08      015006            3108#   3110
09      015020            3109    3113#
PCTYP   062404           11497#  11513
PDATI0  016040            3233    3241    3246    3267    3271    3275    3279    3283    3301    3307    3325#
PDATI1  016042            3326#
PDATI2  016044            3327#
PDATI3  016046            3328#
PDATO0  016050            3238    3242    3251    3308    3330#
PDATO1  016052            3331#
PDATO2  016054            3332#
PDATO3  016056            3333#
PDONE   016060            3249    3293    3303    3310    3318    3335#
PERR0   015526            3231    3259#
PERR1   015746            3255    3305#
PERR10  015546            3262    3265#
PERR11  015556            3260    3267#
PERR12  015574            3268    3271#
PERR13  015612            3272    3275#
PERR14  015630            3276    3279#
PERR15  015646            3280    3283#
PERR16  015656            3284    3286#
PERR17  015664            3266    3270    3274    3278    3282    3288#
PERR2   015774            3257    3312#
PERR20  015712            3285    3295#
PERR21  015722            3248    3298#
PERR22  015730            3297    3299#
PIRQ  = 177772              66#
PIRQVE= 000240             160#
POWERM  062670           11468   11619#
PPAT10  016030            3228    3320#
PPAT11  016032            3321#
PPAT12  016034            3322#
PPAT13  016036            3323#
```

```
PR0     = 000000            83#
PR1     = 000040            84#
PR2     = 000100            85#
PR3     = 000140            86#
PR4     = 000200            87#
PR5     = 000240            88#
PR6     = 000300            89#
PR7     = 000340            90#
PS      = 177776            63#     64
PSW     = 177776            64#    588
PWRVEC= 000024             155#    407*    408*  11437*  11438*  11447*  11453*  11465*  11466*
P1      015410            3225#
P2      015432            3235#   3236    3298    3306    3313
P3      015434            3236#   3259    3400
P4      015436            3238#   3261
P5      015460            3243#   3245
P6      015502            3244    3251#
P7      015512            3253#   3256
P8      015522            3254    3256#
QDATI0  016550            3372    3455    3465    3475#
QDATI1  016552            3476#
QDATI2  016554            3477#
QDATI3  016556            3478#
QDATO0  016540            3367    3371    3382    3390    3456    3470#
QDATO1  016542            3471#
QDATO2  016544            3472#
QDATO3  016546            3473#
QDONE   016560            3380    3431    3441    3451    3458    3480#
QERR0   016226            3358    3400#
QERR1   016472            3394    3453#
QERR11  016236            3403#
QERR12  016254            3404    3407#
QERR13  016272            3408    3411#
QERR14  016310            3412    3415#
QERR15  016326            3416    3420#
QERR16  016336            3421    3424#
QERR17  016344            3406    3410    3414    3418    3426#
QERR2   016426            3387    3443#
QERR20  016372            3422    3433#
QERR21  016402            3379    3436#
QERR22  016410            3435    3437#
QERR3   016436            3396    3445#
QERR4   016444            3444    3447#
QPAT10  016520            3355    3460#
QPAT11  016522            3461#
QPAT12  016524            3462#
QPAT13  016526            3463#
QPAT20  016530            3362    3377    3389    3403    3407    3411    3415    3420    3439    3465#
QPAT21  016532            3466#
QPAT22  016534            3467#
QPAT23  016536            3468#
Q1      016064            3352#
Q10     016222            3393    3395#
Q2      016106            3364#   3365    3436    3449    3454
Q3    = 016110            3365#
Q4      016112            3367#
```

```
Q5      016134          3373#   3375
Q6      016156          3374    3382#
Q7      016166          3384#   3386
Q8      016200          3385    3389#
Q9      016214          3392#   3395
RDCHR = 104410         11426#
RESREG= 104412         11428#
RESVEC= 000010           150#    418*    425*
RSETUP= 104413           565     615     762     860     919    1307    1492    1667    2099    2315    2483    2587    2690
                        2797    2907    3057    3210    3336    3481    3849    4009    4112    4199    4409    4715    4888
                        4897    4984    5456    5908    6408    6757    7082    7246    7513    7741    8007    8186    8444
                        8716    8975    9244    9635   10097   10340   10594   11429#  11543   11559   11575
SADR    012506          2398*   2411    2416*   2423    2429    2470#
SAVREG= 104411         11427#
SDAT00  012522          2350    2353    2381    2384    2477#
SDAT01  012524          2478#
SDAT02  012526          2479#
SDAT03  012530          2480#
SDONE   012532          2416    2435    2438    2442    2445    2457    2468    2482#
SERR0   012212          2337    2394#
SERR1   012422          2342    2448#
SERR10  012232          2400#   2417
SERR15  012312          2405    2420#
SERR2   012352          2358    2433#
SERR20  012332          2410    2426#
SERR3   012376          2361    2440#
SERR4   012270          2371    2413#
SERR5   012454          2373    2459#
SERR6   012364          2389    2436#
SERR7   012410          2391    2443#
SPACE   062740         11619#
SPAT10  012512          2334    2354    2368    2385    2472#
SPAT11  012514          2473#
SPAT12  012516          2474#
SPAT13  012520          2475#
STACK - 001100            54#     399     473   11614
START   001466           181     392#  11470
STKLMT  177774            65#
SWR     001140           209#     397     433*    435     441*    448*    462   10811   10843   10857   10859   10865   10872
                       10911   10918   10930   10934   11287   11324*  11445   11458*  11603
SWREG   000176           179#     441     462   11287   11300
SW0     000001           118#
SW00    000001           108#     118
SW01    000002           107#     117
SW02  - 000004           106#     116
SW03  - 000010           105#     115
SW04  = 000020           104#     114
SW05  - 000040           103#     113
SW06    000100           102#     112
SW07  - 000200           101#     111
SW08  - 000400           100#     110
SW09  - 001000            99#     109
SW1   = 000002           117#
SW10  = 002000            98#
SW11  = 004000            97#
SW12  - 010000            96#
```

```
SW13  = 020000            95#
SW14  = 040000            94#
SW15  = 100000            93#
SW2   = 000004           116#
SW3   = 000010           115#
SW4   = 000020           114#
SW5   = 000040           113#
SW6   = 000100           112#
SW7   = 000200           111#
SW8   = 000400           110#
SW9   = 001000           109#
S1      012002          2331#
S10     012150          2379#
S11     012174          2387#  2390
S12     012204          2388   2390#
S2      012036          2345#  2433   2440
S3      012040          2346#  2394   2448
S4      012044          2348#  2450
S5      012070          2356#  2359
S6      012100          2357   2359#
S7      012106          2364#  2398
S8      012142          2376#  2436   2443   2459
S9      012144          2377#  2413   2461
TAB   = 000011            48#  11619
TBITVE= 000014           151#   415*   416*
TDATI0  007066          1510   1515   1532   1553   1558   1587   1623   1661#
TDATI1  007070          1662#
TDATI2  007072          1584*  1663#
TDATI3  007074          1585*  1664#
TDATO0  007056          1528   1531   1580   1586   1624   1656#
TDATO1  007060          1657#
TDATO2  007062          1537   1592   1658#
TDATO3  007064          1595   1659#
TDONE   007076          1600   1619   1626   1633   1644   1666#
TERR0   006644          1522   1604#
TERR1   006726          1541   1594   1622#
TERR2   006746          1543   1629#
TERR25  006762          1631#  1636
TERR3   007012          1575   1639#
TERR4   006774          1597   1634#
TKVEC = 000060           158#
TPAT10  007036          1509   1552   1646#
TPAT11  007040          1647#
TPAT12  007042          1592   1648#
TPAT13  007044          1595   1649#
TPAT20  007046          1518   1561   1651#
TPAT21  007050          1652#
TPAT22  007052          1653#
TPAT23  007054          1654#
TPVEC = 000064           159#
TRAPVE= 000034           157#   405*   406*
TRTVEC= 000014           152#
TST1    002220           496#
TST10   006306          1505#
TST11   007100          1686#
TST12   011054          2113#
```

```
TST13   012000            2330#
TST14   012534            2496#
TST15   013030            2601#
TST16   013322            2704#
TST17   013644            2812#
TST2    002446             578#
TST20   014174            2921#
TST21   014702            3072#
TST22   015406            3223#
TST23   016062            3350#
TST24   016562            3502#
TST25   017676            3721#
TST26   020406            3863#
TST27   021202            4023#
TST3    002550             632#
TST30   021606            4127#
TST31   022110            4214#
TST32   023142            4425#
TST33   024730            4732#
TST34   026234            4997#
TST35   030230            5335#
TST36   031004            5471#
TST37   031360            5570#   5592#
TST4    003176             789#
TST40   033304            5929#
TST41   036222            6421#
TST42   037742            6774#
TST43   040750            7099#
TST44   041452            7262#
TST45   042462            7528#
TST46   043416            7757#
TST47   044426            8023#
TST5    003440             875#
TST50   045156            8203#
TST51   046016            8462#
TST52   046756            8737#
TST53   047620            8992#
TST54   050562            9262#
TST55   052240            9652#
TST56   054346           10114#
TST57   055274           10357#
TST6    003600             939#
TST60   056322           10607#
TST7    005460            1320#
TYPDS = 104405           10790   11421#
TYPE  = 104401             456   10788   10791   10913   10921   11036   11133   11209   11298   11299   11302   11315   11326
                         11345   11417#  11467   11487   11497   11509   11511
TYPOC = 104402           11301   11418#  11500
TYPON = 104404           11420#
TYPOS = 104403           11419#
T1      006310            1506#
T10     006444            1540    1542#
T105    006452            1535    1545#
T11     006454            1549#
T12     006456            1552#
T13     006472            1555#   1556
```

```
T14     006526          1565    1567#
T15     006530          1568#   1608    1640
T16     006532          1569#   1610
T17     006550          1574    1577#
T2      006330          1512#   1513
T20     006610          1589#   1599
T21     006624          1594#   1596
T22     006626          1593    1595#
T23     006640          1590    1599#
T3      006370          1523    1524#
T4      006372          1525#   1604
T5      006374          1526#   1606
T6      006420          1534#   1545
T7      006434          1539#   1542
UDONE   017676          3620    3629    3647    3662    3673    3678    3710#
UERR0   017342          3511    3623#
UERR1   017366          3532    3593    3633#
UERR10  017374          3635#   3649
UERR11  017436          3644#
UERR2   017452          3555    3574    3648#
UERR20  017460          3634    3650#
UERR21  017522          3659#
UERR3   017536          3608    3665#
UERR4   017566          3672    3674#
UFLAG   017662          3503*   3594    3601*   3704#
UPAT00  017612          3504    3516    3539    3560    3579    3597    3611    3679#
UPAT01  017614          3680#
UPAT02  017616          3681#
UPAT03  017620          3682#
UPAT10  017622          3522    3684#
UPAT11  017624          3685#
UPAT12  017626          3686#
UPAT13  017630          3687#
UPAT20  017632          3545    3689#
UPAT21  017634          3690#
UPAT22  017636          3691#
UPAT23  017640          3692#
UPAT30  017642          3565    3694#
UPAT31  017644          3695#
UPAT32  017646          3696#
UPAT33  017650          3697#
UPAT40  017652          3584    3613    3699#
UPAT41  017654          3700#
UPAT42  017656          3701#
UPAT43  017660          3702#
UROM1   017670          3518*   3541*   3562*   3581*   3650    3707#
UROM2   017672          3519*   3542*   3563*   3582*   3651    3708#
UROM3   017674          3520*   3543*   3564*   3583*   3636    3709#
UTMP1   017664          3509*   3518    3581    3602*   3705#
UTMP2   017666          3510*   3541    3562    3603*   3706#
U0      016600          3506#   3507
U1      016630          3512#   3604
U10     017166          3585#   3588
U11     017170          3586#   3623
U12     017222          3592    3594#
U13     017240          3599#   3600
```

```
U14     017272          3595    3605#
U15     017322          3614#   3617    3674
U16     017324          3615#   3665
U2      016674          3523#   3525
U3      016726          3531    3534#
U4      016772          3547#   3549
U5      017024          3554    3556#
U6      017070          3567#   3569
U7      017122          3573    3575#
WDAP00  020374          3843#
WDAT01  020376          3844#
WDAT02  020400          3845#
WDAT03  020402          3846#
WDONE   020404          3741    3749    3770    3778    3798    3806    3828    3831    3836    3848#
WPAT00  020364          3726    3729    3733    3735    3754    3759    3763    3765    3783    3786    3790    3792    3811
                        3816    3820    3822    3838#
WPAT01  020366          3839#
WPAT02  020370          3840#
WPAT03  020372          3841#
W1      017700          3722#
W10     020106          3768    3771#
W11     020134          3773    3779#
W12     020164          3785    3787#
W13     020210          3794#   3799
W14     020220          3795    3799#
W15     020246          3801    3807#
W16     020302          3813    3817#
W17     020326          3824#   3829
W2      017730          3728    3730#
W20     020336          3825    3829#
W3      017754          3737#   3742
W4      017766          3738    3742#
W5      020016          3744    3750#
W6      020052          3756    3760#
W7      020076          3767#   3771
XAPT11  021162          4000#
XBUF    057114          10652   10654   10666   10668   10683   10685   10731   10733   10747#
XDAT00  021140          3876    3900    3924    3948    3966    3971    3989#
XDAT01  021142          3990#
XDAT02  021144          3991#
XDAT03  021146          3992#
XDONE   021200          3960    3968    3974    3982    3987    4008#
XERR1   021024          3882    3906    3964#
XERR2   021106          3887    3911    3978#
XERR3   021052          3930    3954    3969#
XERR4   021122          3935    3959    3983#
XNEXT   057224          10744   10766#
XPAT0   057214          10691*  10692   10698*  10699   10763#
XPAT0   057124          10621   10623   10632   10634   10645   10676   10705   10749#
XPAT00  021150          3868    3878    3916    3926    3994#
XPAT01  021152          3995#
XPAT02  021154          3996#
XPAT03  021156          3997#
XPAT1   057134          10751#
XPAT10  021160          3872    3896    3920    3944    3964    3969    3999#
XPAT12  021164          4001#
```

```
XPAT13  021166          4002#
XPAT2   057144         10730   10753#
XPAT20  021170          3892    3902    3940    3950    4003#
XPAT21  021172          4004#
XPAT22  021174          4005#
XPAT23  021176          4006#
XPAT3   057154         10646   10660   10677   10755#
XPAT4   057164         10690   10697   10706   10713   10714   10721   10722   10729   10757#
XPAT5   057174         10738   10759#
XPAT6   057204         10739   10761#
XTMP    021136          3869*   3893*   3917*   3941*   3965    3970    3972    3988#
XT1     056326         10610#
XT1A    056342         10610   10613#
XT10    056754         10709   10712#
XT11    057002         10717   10720#
XT12    057030         10725   10728#
XT13    057064         10734   10737#
XT13B   057112         10742   10744#
XT2     056344         10612   10615#
XT2A    056364         10619   10621#
XT2B    056402         10624   10626#
XT3     056414         10628   10631#
XT3A    056436         10635   10637#
XT4     056446         10639   10642#
XT4A    056476         10643   10648#
XT4B    056510         10650   10652#
XT5     056526         10655   10658#
XT5A    056546         10658   10662#
XT5B    056560         10664   10666#
XT6     056576         10669   10672#
XT6A    056624         10674   10679#
XT6B    056636         10681   10683#
XT7     056654         10686   10689#
XT8     056702         10693   10696#
XT9     056730         10700   10704#
XX1     056324         10608#
X1      020410          3864#
X10     020602          3905    3907#
X11     020616          3910    3912#
X12     020652          3919    3921#
X13     020676          3928#   3931
X14     020704          3929    3931#
X15     020720          3934    3936#
X16     020754          3943    3945#
X17     021000          3952#   3955
X2      020444          3871    3873#
X20     021006          3953    3955#
X21     021022          3958    3960#
X3      020470          3880#   3883
X4      020476          3881    3883#
X5      020514          3886    3888#
X6      020550          3895    3897#
X7      020574          3904#   3907
YDAT00  021544          4038    4058    4067    4075    4091#
YDAT01  021546          4092#
YDAT02  021550          4093#
```

```
YDAT03  021552          4094#
YDONE   021604          4055    4070    4078    4084    4111#
YERR1   021404          4060    4063#
YERR2   021446          4062    4071#
YERR3   021510          4047    4079#
YFLAG   021534          4024*   4048    4050*   4086#
YPAT00  021554          4025    4052    4096#
YPAT01  021556          4097#
YPAT02  021560          4098#
YPAT03  021562          4099#
YPAT10  021564          4026    4051    4101#
YPAT11  021566          4102#
YPAT12  021570          4103#
YPAT13  021572          4104#
YPAT20  021574          4032    4066    4074    4106#
YPAT21  021576          4107#
YPAT22  021600          4108#
YPAT23  021602          4109#
YTMP1   021536          4025*   4034    4051*   4057    4065    4073    4087#
YTMP2   021540          4026*   4041    4052*   4068    4076    4088#
YTMP3   021542          4027*   4045    4053*   4082    4089#
Y1      021232          4028#   4054
Y2      021254          4035#   4064    4072    4080
Y3      021300          4042#   4044
Y4      021316          4046    4048#
Y5      021356          4049    4055#
Y6      021360          4043    4056#
Y7      021374          4059#   4061
ZDAT00  022046          4141    4163    4178#
ZDAT01  022050          4179#
ZDAT02  022052          4180#
ZDAT03  022054          4181#
ZDONE   022106          4158    4166    4172    4198#
ZERR1   021752          4147    4159#
ZERR2   022014          4151    4167#
ZFLAG   022040          4128*   4152    4154*   4174#
ZPAT00  022056          4129    4183#
ZPAT01  022060          4184#
ZPAT02  022062          4185#
ZPAT03  022064          4186#
ZPAT10  022066          4155    4188#
ZPAT11  022070          4189#
ZPAT12  022072          4190#
ZPAT13  022074          4191#
ZPAT20  022076          4135    4162    4193#
ZPAT21  022100          4194#
ZPAT22  022102          4195#
ZPAT23  022104          4196#
ZTMP1   022042          4129*   4137    4144    4155*   4161    4164    4175#
ZTMP2   022044          4130*   4149    4156*   4170    4176#
Z1      021630          4131#   4157
Z2      021652          4138#   4160    4168
Z3      021676          4145#   4148
Z4      021704          4146    4148#
Z5      021716          4150    4152#
Z6      021750          4153    4158#
```

```
$APTHD  001452            377    383#
$ASTAT= ****** U        11253  11268
$ATYC   061210          11224  11226#
$ATY1   061164          11222#
$ATY3   061172          11021  11223#
$ATY4   061202          10926  11225#
$AUTOB  001134            206#   466*  11295  11385
$BASE   001372            312#
$BDADR  001122            201#
$BDDAT  001126            203#   648*   663*   679*   694*   711*   726*   742*   757*
$BELL   001306            263#  10913  10946
$CDW1   001376            314#
$CDW2   001400            315#
$CHARC  060506          11038*  11048*  11055  11064*  11069#
$CKSWR  061432          11287#  11425
$CLR.T  057320          10795   10798#
$CMTAG  001100            189#   394    395    403    409    410    411
$CM1  = 000024            221#   222#   223#   224#   225#   226#   227#   228#   229#   230#   231#   232#   233#
                          234#   235#   236#   237#   238#   239#   240#   241#
$CM2  = 000050            221#   222#   223#   224#   225#   226#   227#   228#   229#   230#   231#   232#   233#
                          234#   235#   236#   237#   238#   239#   240#   241#
$CM3  = 000024            219#   221
$CM4  = 000024            241#   242#   243#   244#   245#   246#   247#   248#   249#   250#   251#   252#   253#
                          254#   255#   256#   257#   258#   259#   260#   261#
$CNTLG  062041          11298  11380#
$CNTLU  062034          11315  11379#
$CPUOP  001344            286#
$CRLF   001313            265#  10921  10946  11037  11072  11326  11488  11512
$DBLK   061154          11175  11209  11217#
$DDW0   001402            316#
$DDW1   001404            317#
$DDW10  001426            326#
$DDW11  001430            327#
$DDW12  001432            328#
$DDW13  001434            329#
$DDW14  001436            330#
$DDW15  001440            331#
$DDW2   001406            318#
$DDW3   001410            319#
$DDW4   001412            320#
$DDW5   001414            321#
$DDW6   001416            322#
$DDW7   001420            323#
$DDW8   001422            324#
$DDW9   001424            325#
$DEVCT  001326            277#
$DEVM   001374            313#
$DOAGN  057370          10784  10809#
$DTBL   061144          11178  11213#
$ENDAD  057322            364    454  10799#  10941
$ENDCT  057260            409  10786#
$ENDMG  057437          10788  10823#
$ENULL  057434          10791  10822#
$ENV    001336            282#   460  10923  11016  11231  11255
$ENVM   001337            283#   446  11018  11023  11233
$EOP    057224          10777#  11548  11564  11580
```

```
$EOPCT  057252          409*  10783#  10787
$ERFLG  001103          192#  10832   10861   10863   10869*  10891   10908*  10946
$ERMAX  001115          198#    412*  10863   10886*  10891
$ERROR  057734          403   10906#
$ERRPC  001116          199#  10915*  10916*  10917   10946   11491   11499
$ERRTB  001442          350#  11505
$ERTTL  001112          196#  10914*  10946
$ESCAP  001304          262#    411*  10885*  10937   10939   10946
$ETABL  001336          281#
$ETEND  001442          334#    389
$FATAL  001320          274#  11259*
$FFLG   061430        11222*  11225*  11253   11262*  11270#
$FILLC  001156          217#  11041   11072
$FILLS  001155          216#  11072
$GDADR  001120          200#
$GDDAT  001124          202#
$GET42  057302        10792#
$GTSWR  061502        11299#  11423
$HD   - 000003           41     42
$HIBTS  001452          384#
$ICNT   001104          193#  10876*  10877   10879*  10890
$ILLUP  062334        11437   11453   11474#
$INTAG  001135          207#  11327   11385
$ITEMB  001114          197#  10917*  10925   10946   11494
$LF     001314          266#  10946   11072
$LFLG   061427        11263*  11269#
$LOOP   057426        10816   10819#
$LPADR  001106          194#    427*  10867*  10883*  10888   10890
$LPERR  001110          195#    428*  10867   10884*  10890   10936   11589*
$MADR1  001350          299#
$MADR2  001354          303#
$MADR3  001360          306#
$MADR4  001364          309#
$MAIL   001316          272#    385     389     445     460   10882   10923   11016
$MAMS1  001346          293#
$MAMS2  001352          301#
$MAMS3  001356          304#
$MAMS4  001362          307#
$MBADR  001454          385#
$MFLG   061426        11223*  11229   11264*  11268#
$MNEW   062057        11302   11383#
$MSGAD  001332          279#  11239*  11242
$MSGLG  001334          280#  11244*
$MSGTY  001316          273#  11237   11245*  11257   11261*
$MSWR   062046        11299   11381#
$MTYP1  001347          294#
$MTYP2  001353          302#
$MTYP3  001357          305#
$MTYP4  001363          308#
$MXCNT  057732        10880   10890#
$NULL   001154          215#  11043   11072
$NWTST- 000001          485#    487     572#    574     621#    623     769#    771     867#    869     926#    928    1314#
                       1316   1499#   1501    1675#   1677    2106#   2108    2322#   2324    2489#   2491    2594#   2596
                       2697#  2699    2804#   2806    2914#   2916    3064#   3066    3217#   3219    3343#   3345    3487#
                       3489   3713#   3715    3856#   3858    4015#   4017    4117#   4119    4206#   4208    4414#   4416
                       4721#  4723    4989#   4991    5327#   5329    5461#   5463    5581#   5583    5915#   5917    6415#
```

```
                        6417   6766#  6768   7091#  7093   7254#  7256   7521#  7523   7749#  7751   8015#  8017
                        8194#  8196   8453#  8455   8725#  8727   8984#  8986   9254#  9256   9644#  9646   10106#
                       10108   10349# 10351  10601# 10603
$OCNT   060734         11105*  11134* 11147#
$OMODE  060736         11100*  11104* 11109  11112* 11123* 11149#
$OVER   057716         10844   10860  10868  10878  10887#
$PASS   001324           276#   445*  10780* 10781* 10789  10874  10891
$PASTM  001460           387#
$PWRAD  062316         11470#
$PWRDN  062156           407   11437# 11465
$PWRMG  062312         11468#
$PWRUP  062230         11447   11453#
$QUES   001312           264#  10946  11072  11345
$RDCHR  061714         11358#  11426
$RDDEC= ****** U       11427
$RDLIN= ****** U       11427
$RDOCT= ****** U       11427
$RDSZ - 000001         11379#
$REGAD  001160           219#
$REG0   001162           221#
$REG1   001164           222#
$REG10  001202           229#
$REG11  001204           230#
$REG12  001206           231#
$REG13  001210           232#
$REG14  001212           233#
$REG15  001214           234#
$REG16  001216           235#
$REG17  001220           236#
$REG2   001166           223#
$REG20  001222           237#
$REG21  001224           238#
$REG22  001226           239#
$REG23  001230           240#
$REG3   001170           224#
$REG4   001172           225#
$REG5   001174           226#
$REG6   001176           227#
$REG7   001200           228#
$RESRE  060172         10980#  11428
$RTNAD  057430         10820#
$RTRN   057424           415    417*   422*  10796  10817#
$R2A  - ****** U       11429
$SAVRE  060134         10964#  11427
$SAVR6  062340         11446*  11454  11455* 11456* 11476#
$SCOPE  057454           401   10841#
$SETUP= 000137           170#   400    401   403    405    407    409    410    411    413    427    454    457
                       10842   10907  10933  10941  11282  11385  11472
$STUP - 177777           170#
$SVLAD  057662         10852   10881#
$SVPC - 001452           362#   367
$SWR    177400            41    42#    46#   261    262    263    410    411    413    427    428    497    579
                         633    790    876   940   1321   1506   1687   2114   2331   2497   2602   2705   2813
                        2922   3073   3224  3351  3503   3722   3864   4024   4128   4215   4426   4733   4998
                        5336   5472   5593  5930  6422   6775   7100   7263   7529   7758   8024   8204   8463
                        8738   8993   9263  9653  10115  10358  10608  10833  10834  10835  10836  10837  10843
```

		10855 10857 10858 10861 10862 10863 10870 10871 10872 10884 10887 10890 10898 10899 10900 10901 10902 10911 10918 10930 10934 10946 11471
$SWREG	001340	284# 448
$SWRMK=	000000	10837 10838 10859
$SWRMS=	000200	47#
$TAB	062736	11619#
$TBIT	057432	426* 10813* 10821# 11472*
$TERM =	000032	11431#
$TESTN	001322	275# 10882*
$TIMES	001302	261# 410* 10779* 10870* 10877 10880* 10890
$TKB	001146	212# 11280 11291 11308 11362 11368
$TKS	001144	211# 11280 11289 11305 11329* 11360 11366
$TMP0	001232	241# 10803* 10808 11489* 11490*
$TMP1	001234	242# 11491*
$TMP10	001252	249# 1207* 1238* 1243* 1273* 3524* 3548* 3568* 3587* 5734* 5743* 5752* 5761* 5770* 5779* 5788* 5797* 5806* 5815* 5824* 5833* 6155* 6165* 6175* 6187* 6197* 6226* 6236* 6246* 6256* 6266* 6276* 6306* 6316* 6326* 6336* 6346* 6356* 7234* 7480* 7703* 7974* 8148* 8328* 8600* 8857* 9127* 9570* 10024* 10275* 10521*
$TMP11	001254	250# 5735* 5744* 5753* 5762* 5771* 5780* 5789* 5798* 5807* 5816* 5825* 5834* 6156* 6166* 6176* 6188* 6198* 6227* 6237* 6247* 6257* 6267* 6277* 6307* 6317* 6327* 6337* 6347* 6357* 8348* 8418* 8620* 8690* 8887* 8965* 9156* 9234* 9571* 9620* 10026* 10085* 10276* 10321* 10522* 10575*
$TMP12	001256	251# 9572* 9621* 10025* 10086* 10277* 10322* 10523* 10576*
$TMP13	001260	252# 10278* 10327* 10524* 10581*
$TMP14	001262	253# 10279* 10328* 10525* 10582*
$TMP15	001264	254# 1206* 1242*
$TMP16	001266	255#
$TMP17	001270	256#
$TMP2	001236	243# 528* 542* 555* 560* 598* 639* 656* 672* 686* 703* 718* 734* 749* 800* 854* 881* 912* 943* 986* 1208* 1245* 1339* 1388* 1436* 1464* 1523* 1565* 1614* 1640* 1689* 1717* 1745* 1774* 1802* 1830* 1858* 1886* 1914* 1942* 1970* 1998* 2269 2400* 2433* 2436* 2440* 2443* 2454* 2465* 2544* 2550* 2559* 2649* 2654* 2663* 2751* 2758* 2770* 2857* 2869* 2874* 2886* 2996* 3001* 3011* 3023* 3028* 3147* 3163* 3176* 3181* 3290* 3295* 3298* 3306* 3313* 3428* 3433* 3436* 3449* 3454* 3525* 3549* 3569* 3588* 3617* 3626* 3674* 3728* 3756* 3785* 3813* 3871* 3895* 3919* 3943* 4064* 4072* 4080* 4160* 4168* 4223* 4253* 4289* 4315 4328* 4332* 4431* 4460* 4487* 4517* 4549* 4578* 4739* 4763* 4790* 4814* 4844* 4866* 5003* 5033* 5056* 5079* 5108* 5137* 5167* 5341* 5372* 5477* 5500* 5600* 5634* 5660* 5690* 5714* 5841 5846* 5936* 5961* 5971 5976* 6003* 6028* 6038 6043* 6071* 6096* 6106 6111* 6135 6206 6285 6430* 6466* 6509* 6526* 6547* 6584* 6615* 6689* 7025* 7108* 7132* 7155* 7190* 7193 7459* 7682* 7953* 8127* 8333* 8386* 8411* 8605* 8658* 8683* 8862* 8939* 9132* 9208* 9547* 10001* 10251* 10497* 11536* 11556* 11572*
$TMP20	001272	257#
$TMP21	001274	258#
$TMP22	001276	259#
$TMP23	001300	260#
$TMP3	001240	244# 603* 609* 807* 830* 844* 896* 902* 915* 1054* 1062* 1092* 1100* 1124* 1139* 1154* 1170* 1185* 1193* 1201* 1209* 1244* 1437* 1465* 1616* 1641* 2421* 2427* 2542* 2551* 2560* 2647* 2655* 2664* 2752* 2759* 2771* 2858* 2868* 2875* 2884* 3021* 3029* 3152* 3174* 3182* 3300* 3307* 3438* 3455* 3644* 3659* 3676* 3747* 3776* 3804* 3834* 3964* 3969* 3979* 3984* 4065* 4073* 4081* 4161* 4169* 4340* 4346* 4354* 4360* 4364* 4370* 4604* 4608* 4611* 4617* 4623* 4626* 4632* 4638* 4644* 4647* 4653* 4656* 4659* 4662* 4886* 4895* 4905* 4914* 4920* 4929* 5195* 5199* 5206* 5213* 5220* 5227* 5234* 5241* 5248* 5255* 5262* 5269* 5276* 5399* 5403* 5410* 5417* 5424* 5519* 5525* 5533* 5736* 5745* 5754* 5763*

		5772*	5781*	5790*	5799*	5808*	5817*	5826*	5835*	5850*	6147*	6157*	6167*	6177*
		6189*	6199*	6218*	6228*	6238*	6248*	6258*	6268*	6278*	6298*	6308*	6318*	6328*
		6338*	6348*	6358*	6640*	6645*	6661*	6666*	6699*	7039*	7071*	7200*	7219*	7230*
		7473*	7696*	7967*	8141*	8323*	8595*	8852*	9122*	9562*	10016*	10267*	10513*	11539*
$TMP4	001242	245#	604*	610*	831*	845*	897*	903*	1055*	1061*	1088*	1099*	1125*	1194*
		1202*	1438*	1445*	1466*	1617*	1623*	1642*	2420*	2426*	2543*	2552*	2561*	2648*
		2656*	2665*	2753*	2760*	2772*	2859*	2864*	2876*	2885*	3022*	3030*	3161*	3175*
		3183*	3301*	3308*	3439*	3456*	3645*	3660*	3675*	3746*	3775*	3803*	3833*	3965*
		3970*	3980*	3985*	4066*	4074*	4082*	4162*	4170*	4339*	4345*	4353*	4359*	4365*
		4371*	4603*	4607*	4613*	4619*	4628*	4634*	4640*	4650*	4885*	4894*	4903*	4909*
		4912*	4918*	4924*	4927*	4933*	4936*	5194*	5200*	5207*	5214*	5221*	5228*	5235*
		5242*	5249*	5256*	5263*	5270*	5277*	5398*	5404*	5411*	5418*	5425*	5520*	5526*
		5534*	5737*	5746*	5755*	5764*	5773*	5782*	5791*	5800*	5809*	5818*	5827*	5836*
		5852*	6148*	6158*	6168*	6178*	6190*	6200*	6219*	6229*	6239*	6249*	6259*	6269*
		6279*	6299*	6309*	6319*	6329*	6339*	6349*	6359*	6639*	6644*	6662*	6667*	6697*
		7040*	7041*	7072*	7201*	7210*	7218*	7229*	7475*	7698*	7969*	8143*	8325*	8597*
		8854*	9124*	9564*	10018*	10269*	10515*	11541*						
$TMP5	001244	246#	634*	799*	1056*	1063*	1101*	1126*	1140*	1155*	1171*	1186*	1195*	1211*
		1246*	1446*	1451*	1457*	1624*	1631*	2056*	3003*	3009*	3154*	3288*	3314*	3426*
		3447*	3635*	3650*	3966*	3971*	4067*	4075*	4163*	4341*	4347*	4355*	4361*	4366*
		4372*	4614*	4620*	4629*	4635*	4641*	4649*	4906*	4915*	4921*	4930*	5201*	5208*
		5215*	5222*	5229*	5236*	5243*	5250*	5257*	5264*	5271*	5278*	5405*	5412*	5419*
		5426*	5527*	5535*	5738*	5747*	5756*	5765*	5774*	5783*	5792*	5801*	5810*	5819*
		5828*	5837*	6149*	6159*	6169*	6179*	6191*	6201*	6220*	6230*	6240*	6250*	6260*
		6270*	6280*	6300*	6310*	6320*	6330*	6340*	6350*	6360*	6649*	6655*	6670*	6676*
		6703*	6706*	6712*	7042*	7202*	7211*	7231*	7477*	7700*	7971*	8145*	8327*	8599*
		8856*	9126*	9566*	10020*	10271*	10517*							
$TMP6	001246	247#	1057*	1068*	1073*	1106*	1111*	1127*	1141*	1156*	1172*	1187*	1196*	1220*
		1230*	1237*	1256*	1266*	1272*	1452*	1458*	1629*	1634*	3005*	3008*	3014*	3016*
		3018*	3156*	3160*	3166*	3168*	3170*	3265*	3269*	3273*	3277*	3281*	3286*	3316*
		3405*	3409*	3413*	3417*	3424*	3443*	3445*	3643*	3658*	3972*	4068*	4076*	4164*
		4338*	4344*	4350*	4352*	4358*	4367*	4373*	4612*	4618*	4624*	4627*	4633*	4639*
		4645*	4648*	4654*	4657*	4660*	4663*	4904*	4910*	4913*	4919*	4925*	4928*	4934*
		4937*	5202*	5209*	5216*	5223*	5230*	5237*	5244*	5251*	5258*	5265*	5272*	5279*
		5406*	5413*	5420*	5427*	5528*	5536*	5739*	5748*	5757*	5766*	5775*	5784*	5793*
		5802*	5811*	5820*	5829*	5838*	6150*	6160*	6170*	6180*	6192*	6202*	6221*	6231*
		6241*	6251*	6261*	6271*	6281*	6301*	6311*	6321*	6331*	6341*	6351*	6361*	6651*
		6657*	6672*	6679*	6708*	7043*	7204*	7212*	7220*	7232*	7478*	7701*	7972*	8146*
		8329*	8601*	8858*	9128*	9568*	10022*	10273*	10519*					
$TMP7	001250	248#	635*	668*	699*	731*	1064*	1102*	1142*	1157*	1173*	1188*	1210*	1247*
		1453*	1459*	1630*	1635*	3004*	3010*	3155*	3162*	3289*	3315*	3427*	3448*	3657*
		3652*	6650*	6656*	6671*	6677*	6704*	6707*	6713*	7203*	7213*	7221*	7233*	7479*
		7702*	7973*	8147*	8347*	8619*	8886*	8964*	9155*	9233*	9569*	10023*	10274*	10520*
$TN	000061	41#	485	497#	572	579#	621	633#	769	790#	867	876#	926	940#
		1314	1321#	1499	1506#	1675	1687#	2106	2114#	2322	2331#	2489	2497#	2594
		2602#	2697	2705#	2804	2813#	2914	2922#	3064	3073#	3217	3224#	3343	3351#
		3487	3503#	3713	3722#	3856	3864#	4015	4024#	4117	4128#	4206	4215#	4414
		4426#	4721	4733#	4989	4998#	5327	5336#	5461	5472#	5570	5581	5593#	5915
		5930#	6415	6422#	6766	6775#	7091	7100#	7254	7263#	7521	7529#	7749	7758#
		8015	8024#	8194	8204#	8453	8463#	8725	8738#	8984	8993#	9254	9263#	9644
		9653#	10106	10115#	10349	10358#	10601	10608#						
$TPB	001152	214#	11061*	11072										
$TPFLG	001157	218#	11010	11072										
$TPS	001150	213#	11059	11072										
$TRAP	062070	405	11394#											
$TRAP2	062112	11405#	11416											

```
$TRP   = 000015          11409#  11418#  11419#  11420#  11421#  11422#  11423   11424#  11425   11426#  11427#  11428#  11429#
                         11430#  11431#
$TRPAD  062124           11399   11416#  11431
$TSTM   001456             386#
$TSTNM  001102             191#  10778*  10832   10859   10881*  10882   10887   10891   10910   10946   11489
$TYPBN= ****** U         11422
$TYPDS  060740           11163#  11421
$TYPE   060230           11010#  11250   11409   11417
$TYPEC  060442           11040   11047   11054   11059#  11060   11331
$TYPEX  060510           11065   11067   11070#
$TYPOC  060536           11103#  11418
$TYPON  060552           11102   11105#  11420
$TYPOS  060512           11098#  11419
$UNIT   001330             278#
$UNITM  001462             388#
$USWR   001342             285#
$VECT1  001366             310#
$VECT2  001370             311#
$XTSTR  057466           10846#
$OFILL  060735           11099*  11103*  11113   11148#
$40CAT= ****** U         10843   10920
.       = 063117           173#    177#    188#    267     362     363#    365#    367#    373     374#    376#    378#    398
                           427     428     470#   2824#  10890   10891   10946   11072   11217#  11271#  11280   11385   11431
                         11449   11475
.LPER   062612           11430   11589#
.RSET   062620           11429   11603#
.$ASTA- ****** U         11223   11226
.$X     001452             373#    378
```

```
        8223    8241    8258    8465    8486    8507    8529    8740    8758    8775    8792    8995    9016    9037    9058
        9265    9282    9299    9316    9333    9350    9368    9386    9403    9420    9437    9454    9471    9488    9655
        9678    9701    9724    9747    9770    9794    9818    9841    9867    9890    9913    9936   10117   10135   10153
       10172   10191   10360   10384   10408   10432   10608   10672
MSG     485#     487     572#     574     621#     623     769#     771     867#     869     926#     928    1314#    1316    1498#
        1501    1674#    1677    2106#    2108    2322#    2324    2489#    2491    2593#    2596    2696#    2699    2803#    2806
        2913#    2916    3063#    3066    3216#    3219    3342#    3345    3487#    3489    3713#    3715    3856#    3858    4015#
        4017    4117#    4119    4206#    4208    4414#    4416    4721#    4723    4989#    4991    5327#    5329    5461#    5463
        5581#    5583    5914#    5917    6414#    6417    6765#    6768    7090#    7093    7253#    7256    7520#    7523    7748#
        7751    8014#    8017    8193#    8196    8452#    8455    8724#    8727    8983#    8986    9253#    9256    9643#    9646
       10105#   10108   10348#   10351   10600#   10603
MULT    161#
NAMEP     8#     450
NEWTST  161#     485     572     621     769     867     926    1314    1499    1675    2106    2322    2489    2594    2697
        2804    2914    3064    3217    3343    3487    3713    3856    4015    4117    4206    4414    4721    4989    5327
        5461    5581    5915    6415    6766    7091    7254    7521    7749    8015    8194    8453    8725    8984    9254
        9644   10106   10349   10601
NTMPM     6#
POP     161#   10985   11204   11265   11266   11458   11459
PUSH    161#   10965   11163   11226   11228   11249   11439   11445
REPORT  161#
ROMAC    20#
RSET      8#     564     614     761     859     918    1306    1491    1666    2098    2314    2482    2586    2689    2796
        2906    3056    3209    3335    3480    3848    4008    4111    4198    4408    4714    4888    4897    4983    5455
        5907    6407    6756    7081    7245    7512    7740    8006    8185    8443    8715    8974    9243    9634   10096
       10339   10593   11543   11559   11575
RTMPM     5#
SCOPE     56#     496     578     632     789     875     939    1320    1505    1686    2113    2330    2496    2601    2704
        2812    2921    3072    3223    3350    3502    3721    3863    4023    4127    4214    4425    4732    4997    5335
        5471    5592    5929    6421    6774    7099    7262    7528    7757    8023    8203    8462    8737    8992    9262
        9652   10114   10357   10607   10777
SETPRI  161#
SETTRA 11409#   11418   11419   11420   11421   11423   11425   11426   11427   11428   11429   11430
SETUP   161#     392
SKIP    161#
SLASH   161#
SPACE   161#
SSKAD    22#
STARS   161#     184     267     270     360     370     372     379     485     495     572     577     621     631     769
         788     867     874     926     938    1314    1319    1499    1504    1675    1685    2106    2112    2322    2329
        2489    2495    2594    2600    2697    2703    2804    2811    2914    2920    3064    3071    3217    3222    3343
        3349    3487    3501    3713    3720    3856    3862    4015    4022    4117    4126    4206    4213    4414    4424
        4721    4731    4989    4996    5327    5334    5461    5470    5581    5591    5915    5928    6415    6420    6766
        6773    7091    7098    7254    7261    7521    7527    7749    7756    8015    8022    8194    8202    8453    8461
        8725    8736    8984    8991    9254    9261    9644    9651   10106   10113   10349   10356   10601   10606   10769
       10776   10829   10894   10949   10995   11075   11153   11221   11279   11282   11350   11388   11435   11451   11479
       11530   11552   11568   11586   11594
SUMMAC    21#
SWRSU   161#     429#
TMAC      10#    5570
TRMTRP 11409#   11431
TYPBIN  161#
TYPDEC  161#
TYPNAM  161#     450
TYPNUM  161#
TYPOCS  161#
```

```
TYPOCT    161#  11300
TYPTXT    161#
$$CMRE    182#   221   222   223   224   225   226   227   228   229   230   231   232   233   234
          235    236   237   238   239   240
$$CMTM    182#   241   242   243   244   245   246   247   248   249   250   251   252   253   254
          255    256   257   258   259   260
$$ESCA    161#
$$NEWT    161#   485   572   621   769   867   926  1314  1499  1675  2106  2322  2489  2594  2697
         2804   2914  3064  3217  3343  3487  3713  3856  4015  4117  4206  4414  4721  4989  5327
         5461   5581  5915  6415  6766  7091  7254  7521  7749  8015  8194  8453  8725  8984  9254
         9644  10106 10349 10601
$$SET   11409#  11418 11419 11420 11421 11423 11425 11426 11427 11428 11429 11430
$$SETM    445#
$$SETU    413#
$$SKIP    161#
.EQUAT     29#    51
.HEADE     29#    31
.SETUP     29#   170
.SWRHI     29#
.$ACT1     29#   358
.$APTB     29#   268#
.$APTH     29#   368
.$APTY     29#  11219
.$CATC     29#   171
.$CMTA     29#   182
.$EOP      29#
.$ERRO     29#  10892
.$POWE     29#  11433
.$READ     29#  11277
.$SAVE     29#  10947
.$SCOP     29#  10827
.$TRAP     29#  11386
.$TYPD     29#  11151
.$TYPE     29#  10993
.$TYPO     29#  11073

. ABS.  063117     000

ERRORS DETECTED:  0

CJKDCB.BIN,CJKDCB.LST/CRF/SOL/NL:TOC=CJKDCB.P11
RUN-TIME: 112 89 15 SECONDS
RUN-TIME RATIO: 889/217=4.0
CORE USED.  30K  (59 PAGES)
```