IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DBKEA-A-D
PRODUCT NAME:	KE11F (PDP-11 FIS) INSTRUCTION TESTS
DATE CREATED:	1-AUG-72
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	KEN CHAPMAN

CONTENTS

Table of Contents

5) The interupt test section will type three random length lines of @'s on the console teletype every pass,
6) The program will loop and bell will ring once every pass,
7) A minimum of two passes should always be run,

5, OPERATING PROCEDURE

5,1 Operational switch settings

At SA 200, all switches down is worst case testing, Each subtest will be looped upon until completion of 256 passes of that subtest, The bell will ring upon completion of a pass of the entire program, Alternate pass will run with the T-bit set,

5,1,1 Switch settings are:

```
SW<15> = 1 ,,,,, HALT ON ERROR
SW<14> = 1 ,,,,, SCOPE LOOP
SW<13> = 1 ,,,,, INHIBIT PRINTOUT
SW<12> = 1 ,,,,, INHIBIT TRACE TRAPPING
SW<11> = 1 ,,,,, INHIBIT ITERATIONS OF SUBTEST
SW<10> = 1 ,,,,, BELL ON ERROR
         @ ,,,,, BELL ON PASS COMPLETE
SW<09> = 1 ,,,,, LOOP ON ERROR
SW<08> = 1 ,,,,, LOOP ON TEST IN SW<7:0>
```

Caution: SW<8:0> are also used for ROM word match with KM11 maintenance card,

5,2 Subroutine Abstracts

5,2,1 SCOPE

This subroutine call (via a TRAP instruction) is placed between each subtest in the instruction section, It records the starting address of each subtest as it is being entered in location "LAD$", If a scope loop is requested, the current subtest will be looped upon, SW<11> on a 1 inhibits iteration of subtests, The contents of "LAD$" may be used to determine the last subtest successfully completed,

5,2,2 HLT

This routine (called by an EMT instruction) prints out an error message (See 6,1,), If SW<9> is on a 1 and a HLT is executed, the subtest will be looped upon until 256

consecutive good passes are completed. To inhibit typeouts, put SW<13> on a 1. To ring the bell on an error, put SW<10> on a 1.

5,2,3 NOP

A NOP is placed just before each FIS instruction. This allows the operator to patch in a HALT for debugging purposes.

5,2,4 TRTRAP

If SW<12> is on a 0, the T-bit will be set on alternate passes. When the T-bit is set, the processor traps after each instruction. The first instruction executed upon trapping is an "RTT" which returns to the interrupted sequence of instructions. This sequence is continued until the end of the program is reached.

5,2,5 TRAPCATCHER

A ".+2" = "HALT" sequence is repeated from 0 - 776 to catch any unexpected traps. Thus any unexpected traps or interupts will HALT at the vector + 2.

5,2,6 FLOATING ERROR TRAP (to 244)

If a floating point error (overflow, underflow, or divide by zero) was expected, the vector will point to a unique ISR within the subtest where the error occured which checks the data on the stack(s). If an error was not anticipated, an erroneous trap will be detected in TRAPER.

6, ERRORS

6,1 Error printout

The format is as follows:

ADR PS SP ANS1 ANS2 ANS3 ANS4 ANS5 ANS6

Where:

ADR = Address of error HLT
PS = Processor Status
SP = Contents of Stack Pointer Register
ANS1-6 = Error data read from the STACK(s). From 0 - 6 of

these may be typed depending on the number following the HLT; e.g., HLT+3 would type ANS1 thru ANS3, HLT (by itself) would stop after ADR, PS, and SP.

To find the failing test, look at the listing above the address typed. In most cases the comment beside the HLT tells what was being checked and what was expected.

6.2 Error recovery

Restart at 200

6.3 Error counter

An error count is kept in "ERRORS" (LOC 1002). It can only be cleared from the console or by reloading the program.

7. RESTRICTIONS

None

8. MISCELLANEOUS

8.1 Execution time

Due to the random characteristic of the interupt tests, the execution time can be half a minute or more. However, normally a bell will ring within 15 seconds with all switches down.

8.2 Stack Pointer

Stack is initally set to 500

8.3 Pass count

A 32 bit (2 words) pass count is kept in "PASSES" (LOC 1004,1006). It can only be cleared from the console or by reloading the program.

8.4 Power Fail

Each test can be power failed with no errors, To use, start the test as usual and power down then up at any time, The program should type "POWER" and continue to run from where the power fail interupted with no other error typeouts,

9, PROGRAM DESCRIPTION

This program tests all the instructions of the KE11F (FADD, FSUB, FMUL, and FDIV), All registers are check to see if they function properly as the stack pointer, The program has many subtests (the code between 2 SCOPE statements) which are run 256 times before continuing to the next, SW<11> on a 1 causes each subtest to be run only once, SW<9> on a 1 enables loop on error, The address ICNT (LOC 1000) contains the iteration count in the left byte and the test number in the right byte, All the subtests should be run sequentially by starting at 200 not by starting at the beginning of the subtest, To loop on a particular subtest, put the test number (see listing) in the right byte of the switch register and SW<8> on a 1, This test will be looped upon until SW<8> is put on a 0 or the right byte is changed, If the test is non-existant, the program will be run as usual,

```
                            .TITLE  MAINDEC-11-DBKEA-A      KE11F (PDP-11 FIS) INSTRUCTION TESTS,
                            .ENABL  ABS
                            ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
                            ;PROGRAM BY KEN CHAPMAN

                            ;       SWITCH                  USE
                            ;       ******          ********************
                            ;          8            LOOP ON TEST IN SW<7:0>
                            ;          9            LOOP ON ERROR
                            ;         10            0 = BELL ON PASS COMPLETE
                            ;                       1 = BELL ON ERROR
                            ;         11            INHIBIT ITERATIONS
                            ;         12            INHIBIT TRACE TRAP
                            ;         13            INHIBIT ERROR TYPEOUTS
                            ;         14            LOOP ON TEST
                            ;         15            HALT ON ERROR

                            ;ERROR MESSAGE FORMAT:
                            ;  ADR  PSW  SP  ANS1 ANS2 ANS3 ANS4 ANS5 ANS6
                            ;
                            ;WHERE  ADR =    ADDRESS OF "HLT" INSTRUCTION + 2
                            ;       PSW =    PROCESSOR STATUS WORD
                            ;       SP =     STACK POINTER
                            ;       ANS1 THRU ANS6 =        DATA OFF THE STACK(S)
                            ;       NOTE:    ANS1 THRU ANS6 ARE NOT ALWAYS TYPED, DEPENDING ON THE
                            ;                NUMBER ADDED TO THE "HLT":  "HLT" ALONE TYPES NONE,
                            ;                "HLT+1" TYPES ANS1, "HLT+2" TYPES ANS1 AND ANS2, ETC.

  104400                    SCOPE=  TRAP
  104000                    HLT=    EMT
  000004                    TYPE=   IOT
  177776                    PS=     177776
  177570                    SWR=    177570
  177570                    DISPLAY=SWR
  000007                    BELL=   7
  000000                    R0=     %0
  000001                    R1=     %1
  000002                    R2=     %2
  000003                    R3=     %3
  000004                    R4=     %4
  000005                    R5=     %5
  000005                    TTY=    %5
  000006                    SP=     %6
  000007                    PC=     %7
  100000                    SW15=   100000
  040000                    SW14=    40000
  020000                    SW13=    20000
  010000                    SW12=    10000
  004000                    SW11=     4000
  002000                    SW10=     2000
  001000                    SW09=     1000
  000400                    SW08=      400
```

```
                        000000                  .=      0                       ;TRAP CATCHER FROM 0 - 776
                        000200                  .=      200
000200  000167  000604                          JMP     BEGIN                   ;JUMP TO STARTING ADDRESS OF PROGRAM
                        000600                  .=      600
000600  000000          $PSW:   0               ;PROCESSOR STATUS WORD
000602  000000          $SP:    0               ;STACK POINTER
000604  000000          ANS1:   0               ;FIRST ANSWER (SEE CODE)
000606  000000          ANS2:   0
000610  000000          ANS3:   0
000612  000000          ANS4:   0
000614  000000          ANS5:   0
000616  000000          ANS6:   0
000620  000000  000000  000000                  0,0,0,0                 ;NON-X6 STACK BUFFER
000626  000000
000630  000000          STACK0: 0               ;NON-X6 STACK NORMAL LIMIT
000632  000000          STACK2: 0               ;
000634  000000          STACK4: 0               ;
000636  000000          STACK6: 0               ;
000640  000000  000000  000000  STACK8: 0,0,0,0                 ;NON-X6 STACK BUFFER
000646  000000
                        000631          STACK1 = STACK0+1
000650  000244          FISVEC: 244             ;FIS TRAP VECTOR ADDRESS
000652  000246          FISLVL: 246
177564          TPS:    177564                  ;TELEPRINTER STATUS
177566          TPB:    177566                  ;TELEPRINTER BUFFER
                        001000                  .=      1000
001000  000000          ICNT:   0               ;ITERATION COUNT - LH   TEST NO. - RH
001002  000000          ERRORS: 0               ;ERROR COUNT
001004  000000  000000  PASSES: 0,0             ;PASS COUNTER
001010  000005          BEGIN:  RESET
001012  012706  000500          MOV     #500,   SP
001016  012737  016004  000014  MOV     #YESRT, @#14    ;SET TRACE TRAP VECTOR
001024  012777  017136  016262  MOV     #PDOWN$,@PDVEC$ ;SET UP POWER FAIL VECTOR
001032  012777  000340  016256  MOV     #340,   @PDVEC$+2
001040  012737  017334  000020  MOV     #,IOT,  @#20    ;SET UP VECTOR 20
001046  012700  000030          MOV     #30,    R0      ;SET R0 TO VECTOR 30
001052  012720  016612          MOV     #HLT$,  (0)+    ;SET EMT VECTOR
001056  012720  000340          MOV     #340,   (0)+
001062  012720  016006          MOV     #SCOPE$,(0)+    ;SET TRAP VECTOR
001066  012710  000340          MOV     #340,   (0)
001072  016737  000006  000004  MOV     1$,@#4
001100  005037  177774          CLR     @#177774
001104  012737  000006  000004  1$:     MOV     #6,     @#4     ;RESTORE TIME-OUT VECTOR
001112  005067  177662          CLR     ICNT
001116  005067  015016          CLR     LAD$            ;CLEAR LOOP ADDRESS
001122  012767  000377  015012  MOV     #377,   TIMES   ;INITIALIZE NUMBER OF ITERATIONS
001130  104400                  SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 1:        FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       000000,000000 + 000000,000000 = 000000,000000
                                ;       PS = 004,       STACK POINTER = R0
                                ;*****************************************************************

001132  004567  015164          TST1:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
001136  000000  000000                  ,WORD   000000,000000   ;SECOND OPERAND ON TOP
001142  000000  000000                  ,WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
001146  000000                          ,WORD   000             ;PROCESSOR PRIORITY LEVEL
001150  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR
001154  012700  000630                  MOV     #STACK0,R0      ;SET UP STACK POINTER

001160  000240                          NOP
001162  075000                          FADD+   R0              ;FLOATING ADD ON THE R0 STACK

001164  004767  015164                  JSR     PC,     POPR    ;POP THE ANSWER
001170  010067  177406                  MOV     R0,     $SP     ;SAVE "STACK POINTER"
001174  022767  000004  177376          CMP     #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
001202  001401                          BEQ     .+4             ;BRANCH IF OK
001204  104000                          HLT                     ;PS NOT EQUAL TO 004

001206  022767  000634  177366          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
001214  001401                          BEQ     .+4             ;BRANCH IF OK
001216  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4

001220  005767  177360                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
001224  001401                          BEQ     .+4             ;BRANCH IF OK
001226  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

001230  005767  177352                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001234  001401                          BEQ     .+4             ;BRANCH IF OK
001236  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

001240  122767  000001  177532  END1:   CMPB    #1,     ICNT    ;CHECK THE TEST NUMBER
001246  001401                          BEQ     .+4             ;BRANCH IF OK
001250  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

001252  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 2:        FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       040200,000000 + 040200,000000 = 040400,000000
                                ;       PS = 040,       STACK POINTER = R1
                                ;*****************************************************************

001254  004567  015042          TST2:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
001260  040200  000000                  ,WORD   040200,000000   ;SECOND OPERAND ON TOP
001264  040200  000000                  ,WORD   040200,000000   ;FIRST OPERAND ON BOTTOM
001270  000040                          ,WORD   040             ;PROCESSOR PRIORITY LEVEL
001272  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR
001276  012701  000630                  MOV     #STACK0,R1      ;SET UP STACK POINTER

001302  000240                          NOP
001304  075001                          FADD+   R1              ;FLOATING ADD ON THE R1 STACK

001306  004767  015042                  JSR     PC,     POPR    ;POP THE ANSWER
001312  010167  177264                  MOV     R1,     $SP     ;SAVE "STACK POINTER"
001316  022767  000040  177254          CMP     #040,   $PSW    ;CHECK PS (EXCEPT T BIT)
001324  001401                          BEQ     .+4             ;BRANCH IF OK
001326  104000                          HLT                     ;PS NOT EQUAL TO 040

001330  022767  000634  177244          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
001336  001401                          BEQ     .+4             ;BRANCH IF OK
001340  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4

001342  022767  040400  177234          CMP     #040400,ANS1    ;CHECK FIRST HALF OF ANSWER
001350  001401                          BEQ     .+4             ;BRANCH IF OK
001352  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 040400

001354  005767  177226                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001360  001401                          BEQ     .+4             ;BRANCH IF OK
001362  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

001364  122767  000002  177406  END2:   CMPB    #2,     ICNT    ;CHECK THE TEST NUMBER
001372  001401                          BEQ     .+4             ;BRANCH IF OK
001374  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

001376  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 3:        FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       077777,177777 + 177777,177777 = 000000,000000
                                ;       PS = 104,       STACK POINTER = R2
                                ;*****************************************************************

001400  004567  014716          TST3:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
001404  177777  177777                  ,WORD   177777,177777   ;SECOND OPERAND ON TOP
001410  077777  177777                  ,WORD   077777,177777   ;FIRST OPERAND ON BOTTOM
001414  000100                          ,WORD   100             ;PROCESSOR PRIORITY LEVEL
001416  016626  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR
001422  012702  000630                  MOV     #STACK0,R2      ;SET UP STACK POINTER

001426  000240                          NOP
001430  075002                          FADD+   R2              ;FLOATING ADD ON THE R2 STACK

001432  004767  014716                  JSR     PC,     POPR    ;POP THE ANSWER
001436  010267  177140                  MOV     R2,     $SP     ;SAVE "STACK POINTER"
001442  022767  000104  177130          CMP     #104,   $PSW    ;CHECK PS (EXCEPT T BIT)
001450  001401                          BEQ     .+4             ;BRANCH IF OK
001452  104000                          HLT                     ;PS NOT EQUAL TO 104

001454  022767  000634  177120          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
001462  001401                          BEQ     .+4             ;BRANCH IF OK
001464  104000                          HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4

001466  005767  177112                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
001472  001401                          BEQ     .+4             ;BRANCH IF OK
001474  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

001476  005767  177104                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001502  001401                          BEQ     .+4             ;BRANCH IF OK
001504  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

001506  122767  000003  177264  END3:   CMPB    #3,     ICNT    ;CHECK THE TEST NUMBER
001514  001401                          BEQ     .+4             ;BRANCH IF OK
001516  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

001520  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 41        FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       052525,052525 + 152525,052524 = 044600,000000
                                ;       PS = 200,       STACK POINTER = SP
                                ;*****************************************************************

001522  004567  014416          TST4:   JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
001526  152525  052524                  ,WORD   152525,052524   ;SECOND OPERAND ON TOP
001532  052525  052525                  ,WORD   052525,052525   ;FIRST OPERAND ON BOTTOM
001536  000217                          ,WORD   217             ;PROCESSOR PRIORITY LEVEL
001540  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR

001544  000240                          NOP
001546  075006                          FADD+   SP              ;FLOATING ADD ON THE STACK

001550  004767  014430                  JSR     PC,     POPS    ;POP THE ANSWER
001554  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
001560  001404                          BEQ     TSA4            ;BRANCH IF OK
001562  012706  000500                  MOV     #500,   SP      ;RESTORE STACK POINTER
001566  104000                          HLT                     ;STACK POINTER FOULED UP
001570  000416                          BR      END4            ;SKIP REST OF TEST

001572  022767  000200  177000  TSA4:   CMP     #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
001600  001401                          BEQ     .+4             ;BRANCH IF OK
001602  104000                          HLT                     ;PS NOT EQUAL TO 200

001604  022767  044600  176772          CMP     #044600,ANS1    ;CHECK FIRST HALF OF ANSWER
001612  001401                          BEQ     .+4             ;BRANCH IF OK
001614  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 044600

001616  005767  176764                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001622  001401                          BEQ     .+4             ;BRANCH IF OK
001624  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

001626  122767  000004  177144  END4:   CMPB    #4,     ICNT    ;CHECK THE TEST NUMBER
001634  001401                          BEQ     .+4             ;BRANCH IF OK
001636  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

001640  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 5:        FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       125200,000000 + 025177,177777 = 117200,000000
                                ;       PS = 310,       STACK POINTER = SP
                                ;*****************************************************************

001642  004567  014276          TST5:   JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
001646  025177  177777                  ,WORD   025177,177777   ;SECOND OPERAND ON TOP
001652  125200  000000                  ,WORD   125200,000000   ;FIRST OPERAND ON BOTTOM
001656  000307                          ,WORD   307             ;PROCESSOR PRIORITY LEVEL
001660  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR

001664  000240                          NOP
001666  075006                          FADD+   SP              ;FLOATING ADD ON THE STACK

001670  004767  014310                  JSR     PC,     POPS    ;POP THE ANSWER
001674  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
001700  001404                          BEQ     TSA5            ;BRANCH IF OK
001702  012706  000500                  MOV     #500,   SP      ;RESTORE STACK POINTER
001706  104000                          HLT                     ;STACK POINTER FOULED UP
001710  000416                          BR      END5            ;SKIP REST OF TEST

001712  022767  000310  176660  TSA5:   CMP     #310,   SPSW    ;CHECK PS (EXCEPT T BIT)
001720  001401                          BEQ     .+4             ;BRANCH IF OK
001722  104000                          HLT                     ;PS NOT EQUAL TO 310

001724  022767  117200  176652          CMP     #117200,ANS1    ;CHECK FIRST HALF OF ANSWER
001732  001401                          BEQ     .+4             ;BRANCH IF OK
001734  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 117200

001736  005767  176644                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001742  001401                          BEQ     .+4             ;BRANCH IF OK
001744  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

001746  122767  000005  177024  END5:   CMPB    #5,     ICNT    ;CHECK THE TEST NUMBER
001754  001401                          BEQ     .+4             ;BRANCH IF OK
001756  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

001760  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 6:        FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       135753,024642 + 100125,052525 = 135753,024642
                                ;       PS = 350,       STACK POINTER = R5
                                ;*****************************************************************

001762  004567  014334          TST6:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
001766  100125  052525                  ,WORD   100125,052525   ;SECOND OPERAND ON TOP
001772  135753  024642                  ,WORD   135753,024642   ;FIRST OPERAND ON BOTTOM
001776  000347                          ,WORD   347             ;PROCESSOR PRIORITY LEVEL
002000  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR
002004  012705  000630                  MOV     #STACK0,R5      ;SET UP STACK POINTER

002010  000240                          NOP
002012  075005                          FADD+   R5              ;FLOATING ADD ON THE R5 STACK

002014  004767  014334                  JSR     PC,     POPR    ;POP THE ANSWER
002020  010567  176556                  MOV     R5,     $SP     ;SAVE "STACK POINTER"
002024  022767  000350  176546          CMP     #350,   $PSW    ;CHECK PS (EXCEPT T BIT)
002032  001401                          BEQ     .+4             ;BRANCH IF OK
002034  104000                          HLT                     ;PS NOT EQUAL TO 350

002036  022767  000634  176536          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
002044  001401                          BEQ     .+4             ;BRANCH IF OK
002046  104000                          HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4

002050  022767  135753  176526          CMP     #135753,ANS1    ;CHECK FIRST HALF OF ANSWER
002056  001401                          BEQ     .+4             ;BRANCH IF OK
002060  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 135753

002062  022767  024642  176516          CMP     #024642,ANS2    ;CHECK SECOND HALF OF ANSWER
002070  001401                          BEQ     .+4             ;BRANCH IF OK
002072  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 024642

002074  122767  000006  176676  END6:   CMPB    #6,     ICNT    ;CHECK THE TEST NUMBER
002102  001401                          BEQ     .+4             ;BRANCH IF OK
002104  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

002106  104400                          SCOPE
```

```
354
355                                     ;*************************************************************************
356                                     ;TEST 7:        FADD (KE11F FLOATING ADD INSTRUCTION)
357                                     ;       000052,125252 + 001357,024642 = 001357,024642
358                                     ;       PS = 240,       STACK POINTER = R1
359                                     ;*************************************************************************
360
361  002110  004567  014206            TST7:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
362  002114  001357  024642                    ,WORD   001357,024642   ;SECOND OPERAND ON TOP
363  002120  000052  125252                    ,WORD   000052,125252   ;FIRST OPERAND ON BOTTOM
364  002124  000257                            ,WORD   257             ;PROCESSOR PRIORITY LEVEL
365  002126  016606  000340                    ,WORD   TRAPER,340      ;FIS TRAP VECTOR
366  002132  012701  000630                    MOV     #STACK0,R1      ;SET UP STACK POINTER
367
368  002136  000240                            NOP
369  002140  075001                            FADD+   R1              ;FLOATING ADD ON THE R1 STACK
370
371  002142  004767  014206                    JSR     PC,     POPR    ;POP THE ANSWER
372  002146  010167  176430                    MOV     R1,     $SP     ;SAVE "STACK POINTER"
373  002152  022767  000240  176420            CMP     #240,   $PSW    ;CHECK PS (EXCEPT T BIT)
374  002160  001401                            BEQ     .+4             ;BRANCH IF OK
375  002162  104000                            HLT                     ;PS NOT EQUAL TO 240
376
377  002164  022767  000634  176410            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
378  002172  001401                            BEQ     .+4             ;BRANCH IF OK
379  002174  104000                            HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
380
381  002176  022767  001357  176400            CMP     #001357,ANS1    ;CHECK FIRST HALF OF ANSWER
382  002204  001401                            BEQ     .+4             ;BRANCH IF OK
383  002206  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 001357
384
385  002210  022767  024642  176370            CMP     #024642,ANS2    ;CHECK SECOND HALF OF ANSWER
386  002216  001401                            BEQ     .+4             ;BRANCH IF OK
387  002220  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 024642
388
389  002222  122767  000007  176550  END7:    CMPB    #7,     ICNT    ;CHECK THE TEST NUMBER
390  002230  001401                            BEQ     .+4             ;BRANCH IF OK
391  002232  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
392
393  002234  104400                            SCOPE
394
```

```
                                ;*****************************************************************
                                ;TEST 10:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       100400,000000 + 000200,000000 = 100200,000000
                                ;       PS = 150,       STACK POINTER = R5
                                ;*****************************************************************

002236  004567  014060          TST10:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
002242  000200  000000                  ,WORD   000200,000000   ;SECOND OPERAND ON TOP
002246  100400  000000                  ,WORD   100400,000000   ;FIRST OPERAND ON BOTTOM
002252  000140                          ,WORD   140             ;PROCESSOR PRIORITY LEVEL
002254  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR
002260  012705  000630                  MOV     #STACK0,R5      ;SET UP STACK POINTER

002264  000240                          NOP
002266  075005                          FADD+   R5              ;FLOATING ADD ON THE R5 STACK

002270  004767  014060                  JSR     PC,     POPR    ;POP THE ANSWER
002274  010567  176302                  MOV     R5,     $SP     ;SAVE "STACK POINTER"
002300  022767  000150  176272          CMP     #150,   $PSW    ;CHECK PS (EXCEPT T BIT)
002306  001401                          BEQ     .+4             ;BRANCH IF OK
002310  104000                          HLT                     ;PS NOT EQUAL TO 150

002312  022767  000634  176262          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
002320  001401                          BEQ     .+4             ;BRANCH IF OK
002322  104000                          HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4

002324  022767  100200  176252          CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
002332  001401                          BEQ     .+4             ;BRANCH IF OK
002334  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 100200

002336  005767  176244                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
002342  001401                          BEQ     .+4             ;BRANCH IF OK
002344  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

002346  122767  000010  176424  END10:  CMPB    #10,    ICNT    ;CHECK THE TEST NUMBER
002354  001401                          BEQ     .+4             ;BRANCH IF OK
002356  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

002360  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 11:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       100200,000000 + 000377,177777 ==> UNDERFLOW
                                ;       PS(ON STACK) = 152,     STACK POINTER = R3
                                ;*****************************************************************

002362  004567  013734          TST11:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
002366  000377  177777                  ,WORD   000377,177777   ;SECOND OPERAND ON TOP
002372  100200  000000                  ,WORD   100200,000000   ;FIRST OPERAND ON BOTTOM
002376  000157                        ,WORD   157             ;PROCESSOR PRIORITY LEVEL
002400  002430  000000                  ,WORD   ISR11,  000     ;FIS TRAP VECTOR
002404  012703  000630                  MOV     #STACK0,R3      ;SET UP R3 AS STACK POINTER

002410  000240                          NOP
002412  075003                          FADD+   R3              ;FLOATING ADD ON THE R3 STACK

002414  004767  013734          RTA11:  JSR     %7,     POPR    ;POP THE "ANSWER"
002420  010367  176156                  MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
002424  104002                          HLT+2                   ;FIS TRAP DIDN'T OCCURE!
002426  000452                          BR      END11

002430  004767  013752          ISR11:  JSR     %7,     POPER   ;POP ALL DATA OFF THE STACKS
002434  010367  176142                  MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
002440  005767  176134                  TST     $PSW            ;CHECK PS AFTER FIS TRAP
002444  001401                          BEQ     .+4             ;BRANCH IF OK
002446  104000                          HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 000

002450  022767  000630  176124          CMP     #$TACK0,$SP     ;CHECK THE STACK POINTER (R3)
002456  001401                          BEQ     .+4             ;BRANCH IF OK
002460  104000                          HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK0

002462  022767  002414  176114          CMP     #RTA11, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
002470  001401                          BEQ     .+4             ;BRANCH IF OK
002472  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS

002474  022767  000152  176104          CMP     #152,   ANS2    ;CHECK PS BEFORE FIS TRAP
002502  001401                          BEQ     .+4             ;BRANCH IF OK
002504  104002                          HLT+2                   ;PS AT FIS TRAP TIME NOT 152

002506  022767  000377  176074          CMP     #000377,ANS3    ;CHECK DATA FROM THE STACK
002514  001401                          BEQ     .+4             ;BRANCH IF OK
002516  104004                          HLT+4                   ;DATA ON STACK (000377) CHANGED

002520  022767  177777  176064          CMP     #177777,ANS4    ;CHECK DATA FROM STACK
002526  001401                          BEQ     .+4             ;BRANCH IF OK
002530  104004                          HLT+4                   ;DATA ON STACK (177777) CHANGED

002532  022767  100200  176054          CMP     #100200,ANS5    ;CHECK DATA FROM STACK
002540  001401                          BEQ     .+4             ;BRANCH IF OK
002542  104006                          HLT+6                   ;DATA ON STACK (100200) CHANGED

002544  005767  176046                  TST     ANS6            ;CHECK DATA FROM STACK
002550  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
                                        HLT+6                   ;DATA ON STACK (000000) CHANGED

002554  122767  000011  176216  END11:  CMPB    #11,    ICNT    ;CHECK THE TEST NUMBER
002562  001401                          BEQ     .+4             ;BRANCH IF OK
002564  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

002566  104400                          SCOPE


                                ;*****************************************************************
                                ;TEST 12:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       000425,052525 + 100252,125252 = 000200,000000
                                ;       PS = 200,       STACK POINTER = R4
                                ;*****************************************************************

002570  004567  013526          TST12:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
002574  100252  125252                  ,WORD   100252,125252   ;SECOND OPERAND ON TOP
002600  000425  052525                  ,WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
002604  000217                          ,WORD   217             ;PROCESSOR PRIORITY LEVEL
002606  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR
002612  012704  000630                  MOV     #STACK0,R4      ;SET UP STACK POINTER

002616  000240                          NOP
002620  075004                          FADD+   R4              ;FLOATING ADD ON THE R4 STACK

002622  004767  013526                  JSR     PC,     POPR    ;POP THE ANSWER
002626  010467  175750                  MOV     R4,     $SP     ;SAVE "STACK POINTER"
002632  022767  000200  175740          CMP     #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
002640  001401                          BEQ     .+4             ;BRANCH IF OK
002642  104000                          HLT                     ;PS NOT EQUAL TO 200

002644  022767  000634  175730          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
002652  001401                          BEQ     .+4             ;BRANCH IF OK
002654  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4

002656  022767  000200  175720          CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
002664  001401                          BEQ     .+4             ;BRANCH IF OK
002666  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000200

002670  005767  175712                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
002674  001401                          BEQ     .+4             ;BRANCH IF OK
002676  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

002700  122767  000012  176072  END12:  CMPB    #12,    ICNT    ;CHECK THE TEST NUMBER
002706  001401                          BEQ     .+4             ;BRANCH IF OK
002710  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

002712  104400                          SCOPE
```

```
                                        ;*****************************************************************
                                        ;TEST 13:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                        ;       000200,000000 + 100377,177777 ==> UNDERFLOW
                                        ;       PS(ON STACK) = 252,     STACK POINTER = SP
                                        ;*****************************************************************

002714  004567  013224          TST13:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
002720  100377  177777                  .WORD   100377,177777   ;SECOND OPERAND ON TOP
002724  000200  000000                  .WORD   000200,000000   ;FIRST OPERAND ON BOTTOM
002730  000257                          .WORD   257             ;PROCESSOR PRIORITY LEVEL
002732  002756  000340                  .WORD   ISR13,  340     ;FIS TRAP VECTOR

002736  000240                          NOP
002740  075006                          FADD+   SP              ;FLOATING ADD ON THE STACK

002742  004767  013236          RTA13:  JSR     %7,     POPS    ;POP THE "ANSWER"
002746  104002                          HLT+2                   ;FIS TRAP DIDN'T OCCURE!
002750  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK POINTER
002754  000453                          BR      END13

002756  004767  013256          ISR13:  JSR     %7,     POPES   ;POP ALL DATA OFF THE STACK
002762  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
002766  001404                          BEQ     ISA13           ;BRANCH IF OK
002770  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK POINTER
002774  104000                          HLT                     ;STACK POINTER FOULED UP
002776  000442                          BR      END13           ;SKIP REST OF TEST

003000  022767  000340  175572  ISA13:  CMP     #340,   SPSW    ;CHECK PS AFTER FIS TRAP
003006  001401                          BEQ     .+4             ;BRANCH IF OK
003010  104000                          HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 340

003012  022767  002742  175564          CMP     #RTA13, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
003020  001401                          BEQ     .+4             ;BRANCH IF OK
003022  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS

003024  022767  000252  175554          CMP     #252,   ANS2    ;CHECK PS BEFORE FIS TRAP
003032  001401                          BEQ     .+4             ;BRANCH IF OK
003034  104002                          HLT+2                   ;PS AT FIS TRAP TIME NOT 252

003036  022767  100377  175544          CMP     #100377,ANS3    ;CHECK DATA FROM THE STACK
003044  001401                          BEQ     .+4             ;BRANCH IF OK
003046  104004                          HLT+4                   ;DATA ON STACK (100377) CHANGED

003050  022767  177777  175534          CMP     #177777,ANS4    ;CHECK DATA FROM STACK
003056  001401                          BEQ     .+4             ;BRANCH IF OK
003060  104004                          HLT+4                   ;DATA ON STACK (177777) CHANGED

003062  022767  000200  175524          CMP     #000200,ANS5    ;CHECK DATA FROM STACK
003070  001401                          BEQ     .+4             ;BRANCH IF OK
003072  104006                          HLT+6                   ;DATA ON STACK (000200) CHANGED

003074  005767  175516                  TST     ANS6            ;CHECK DATA FROM STACK
003100  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
                                        HLT+6                   ;DATA ON STACK (000000) CHANGED

003102  104006
                                        CMPB    #13,    ICNT    ;CHECK THE TEST NUMBER
003104  122767  000013  175666  END13:  BEQ     .+4             ;BRANCH IF OK
003112  001401                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
003114  104000
                                        SCOPE
003116  104400


                                ;*****************************************************************
                                ;TEST 14:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       100425,052525 + 000252,125252 = 100200,000000
                                ;       PS = 310,       STACK POINTER = SP
                                ;*****************************************************************

003120  004567  013020          TST14:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
003124  000252  125252                  ,WORD   000252,125252   ;SECOND OPERAND ON TOP
003130  100425  052525                  ,WORD   100425,052525   ;FIRST OPERAND ON BOTTOM
003134  000307                          ,WORD   307             ;PROCESSOR PRIORITY LEVEL
003136  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR

003142  000240                          NOP
003144  075006                          FADD+   SP              ;FLOATING ADD ON THE STACK

003146  004767  013032                  JSR     PC,     POPS    ;POP THE ANSWER
003152  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
003156  001404                          BEQ     TSA14           ;BRANCH IF OK
003160  012706  000500                  MOV     #500,   SP      ;RESTORE STACK POINTER
003164  104000                          HLT                     ;STACK POINTER FOULED UP
003166  000416                          BR      END14           ;SKIP REST OF TEST

003170  022767  000310  175402  TSA14:  CMP     #310,   SPSW    ;CHECK PS (EXCEPT T BIT)
003176  001401                          BEQ     .+4             ;BRANCH IF OK
003200  104000                          HLT                     ;PS NOT EQUAL TO 310

003202  022767  100200  175374          CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
003210  001401                          BEQ     .+4             ;BRANCH IF OK
003212  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 100200

003214  005767  175366                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
003220  001401                          BEQ     .+4             ;BRANCH IF OK
003222  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

003224  122767  000014  175546  END14:  CMPB    #14,    ICNT    ;CHECK THE TEST NUMBER
003232  001401                          BEQ     .+4             ;BRANCH IF OK
003234  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

003236  104400                          SCOPE
```

```
                                        ;*****************************************************************
                                        ;TEST 15:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                        ;       077652,125252 + 077452,125252 = 077777,177777
                                        ;       PS = 240,       STACK POINTER = SP
                                        ;*****************************************************************

003240  004567  012700          TST15:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
003244  077452  125252                  ,WORD   077452,125252   ;SECOND OPERAND ON TOP
003250  077652  125252                  ,WORD   077652,125252   ;FIRST OPERAND ON BOTTOM
003254  000257                          ,WORD   257             ;PROCESSOR PRIORITY LEVEL
003256  016606  000340                  ,WORD   TRAPER,340      ;FIS TRAP VECTOR

003262  000240                          NOP
003264  075006                          FADD+   SP              ;FLOATING ADD ON THE STACK

003266  004767  012712                  JSR     PC,     POPS    ;POP THE ANSWER
003272  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
003276  001404                          BEQ     TSA15           ;BRANCH IF OK
003300  012706  000500                  MOV     #500,   SP      ;RESTORE STACK POINTER
003304  104000                          HLT                     ;STACK POINTER FOULED UP
003306  000417                          BR      END15           ;SKIP REST OF TEST

003310  022767  000240  175262  TSA15:  CMP     #240,   $PSW    ;CHECK PS (EXCEPT T BIT)
003316  001401                          BEQ     .+4             ;BRANCH IF OK
003320  104000                          HLT                     ;PS NOT EQUAL TO 240

003322  022767  077777  175254          CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
003330  001401                          BEQ     .+4             ;BRANCH IF OK
003332  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 077777

003334  022767  177777  175244          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
003342  001401                          BEQ     .+4             ;BRANCH IF OK
003344  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777

003346  122767  000015  175424  END15:  CMPB    #15,    ICNT    ;CHECK THE TEST NUMBER
003354  001401                          BEQ     .+4             ;BRANCH IF OK
003356  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP,

003360  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 16:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       177452,125253 + 177652,125252 ==> OVERFLOW
                                ;       PS(ON STACK) = 102,     STACK POINTER = R1
                                ;*****************************************************************

002362  004567  012734          TST16:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
002366  177652  125252                  ,WORD   177652,125252   ;SECOND OPERAND ON TOP
002372  177452  125253                  ,WORD   177452,125253   ;FIRST OPERAND ON BOTTOM
002376  000105                        ,WORD   105             ;PROCESSOR PRIORITY LEVEL
002400  003430  000252                ,WORD   ISR16,  252     ;FIS TRAP VECTOR
002404  012701  000630                MOV     #STACK0,R1      ;SET UP R1 AS STACK POINTER

002410  000240                        NOP
002412  075001                        FADD+   R1              ;FLOATING ADD ON THE R1 STACK

002414  004767  012734          RTA16:  JSR     X7,     POPR    ;POP THE "ANSWER"
002420  010167  175156                  MOV     R1,     SSP     ;SAVE STACK POINTER (R1)
002424  104002                        HLT+2                   ;FIS TRAP DIDN'T OCCURE!
002426  000454                        BR      END16

002430  004767  012752          ISR16:  JSR     X7,     POPER   ;POP ALL DATA OFF THE STACKS
002434  010167  175142                  MOV     R1,     SSP     ;SAVE STACK POINTER (R1)
002440  022767  000252  175132          CMP     #252,   SPSW    ;CHECK PS AFTER FIS TRAP
002446  001401                        BEQ     .+4             ;BRANCH IF OK
002450  104000                        HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 252

002452  022767  000630  175122          CMP     #STACK0,SSP     ;CHECK THE STACK POINTER (R1)
002460  001401                        BEQ     .+4             ;BRANCH IF OK
002462  104000                        HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0

002464  022767  003414  175112          CMP     #RTA16, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
002472  001401                        BEQ     .+4             ;BRANCH IF OK
002474  104001                        HLT+1                   ;FIS TRAP AT WRONG ADDRESS

002476  022767  000102  175102          CMP     #102,   ANS2    ;CHECK PS BEFORE FIS TRAP
002504  001401                        BEQ     .+4             ;BRANCH IF OK
002506  104002                        HLT+2                   ;PS AT FIS TRAP TIME NOT 102

002510  022767  177652  175072          CMP     #177652,ANS3    ;CHECK DATA FROM THE STACK
002516  001401                        BEQ     .+4             ;BRANCH IF OK
002520  104004                        HLT+4                   ;DATA ON STACK (177652) CHANGED

002522  022767  125252  175062          CMP     #125252,ANS4    ;CHECK DATA FROM STACK
002530  001401                        BEQ     .+4             ;BRANCH IF OK
002532  104004                        HLT+4                   ;DATA ON STACK (125252) CHANGED

002534  022767  177452  175052          CMP     #177452,ANS5    ;CHECK DATA FROM STACK
002542  001401                        BEQ     .+4             ;BRANCH IF OK
002544  104006                        HLT+6                   ;DATA ON STACK (177452) CHANGED

002546  022767  125253  175042          CMP     #125253,ANS6    ;CHECK DATA FROM STACK
002554  001401                        BEQ     .+4             ;BRANCH IF OK
```

```
                                        HLT+6                   ;DATA ON STACK (125253) CHANGED

003560  122767  000016  175212  END16:  CMPB    #16,    ICNT    ;CHECK THE TEST NUMBER
003566  001401                          BEQ     .+4             ;BRANCH IF OK
003570  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

003572  104400                          SCOPE


                                ;*****************************************************************
                                ;TEST 17:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       177452,125252 + 177652,125252 = 177777,177777
                                ;       PS = 350,       STACK POINTER = R4
                                ;*****************************************************************

003574  004567  012522          TST17:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
003600  177652  125252                  .WORD   177652,125252   ;SECOND OPERAND ON TOP
003604  177452  125252                  .WORD   177452,125252   ;FIRST OPERAND ON BOTTOM
003610  000357                          .WORD   357             ;PROCESSOR PRIORITY LEVEL
003612  016606  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
003616  012704  000630                  MOV     #STACK0,R4      ;SET UP STACK POINTER

003622  000240                          NOP
003624  075004                          FADD*   R4              ;FLOATING ADD ON THE R4 STACK

003626  004767  012522                  JSR     PC,     POPR    ;POP THE ANSWER
003632  010467  174744                  MOV     R4,     $SP     ;SAVE "STACK POINTER"
003636  022767  000350  174734          CMP     #350,   $PSW    ;CHECK PS (EXCEPT T BIT)
003644  001401                          BEQ     .+4             ;BRANCH IF OK
003646  104000                          HLT                     ;PS NOT EQUAL TO 350

003650  022767  000634  174724          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
003656  001401                          BEQ     .+4             ;BRANCH IF OK
003660  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4

003662  022767  177777  174714          CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
003670  001401                          BEQ     .+4             ;BRANCH IF OK
003672  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777

003674  022767  177777  174704          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
003702  001401                          BEQ     .+4             ;BRANCH IF OK
003704  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777

003706  122767  000017  175064  END17:  CMPB    #17,    ICNT    ;CHECK THE TEST NUMBER
003714  001401                          BEQ     .+4             ;BRANCH IF OK
003716  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

003720  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 20:       FADD (KE11F FLOATING ADD INSTRUCTION)
                                ;       077652,125253 * 077452,125252 ==> OVERFLOW
                                ;       PS(ON STACK) = 002,     STACK POINTER = SP
                                ;*****************************************************************

003722  004567  012216          TST20:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
003726  077452  125252                  ,WORD   077452,125252   ;SECOND OPERAND ON TOP
003732  077652  125253                  ,WORD   077652,125253   ;FIRST OPERAND ON BOTTOM
003736  000003                          ,WORD   003             ;PROCESSOR PRIORITY LEVEL
003740  003764  000344                  ,WORD   ISR20,  344     ;FIS TRAP VECTOR

003744  000240                          NOP
003746  075006                          FADD+   SP              ;FLOATING ADD ON THE STACK

003750  004767  012230          RTA20:  JSR     %7,     POPS    ;POP THE "ANSWER"
003754  104002                          HLT+2                   ;FIS TRAP DIDN'T OCCURE!
003756  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK POINTER
003762  000454                          BR      END20

003764  004767  012250          ISR20:  JSR     %7,     POPES   ;POP ALL DATA OFF THE STACK
003770  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
003774  001404                          BEQ     ISA20           ;BRANCH IF OK
003776  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK POINTER
004002  104000                          HLT                     ;STACK POINTER FOULED UP
004004  000443                          BR      END20           ;SKIP REST OF TEST

004006  022767  000344  174564  ISA20:  CMP     #344,   $PSW    ;CHECK PS AFTER FIS TRAP
004014  001401                          BEQ     .+4             ;BRANCH IF OK
004016  104000                          HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 344

004020  022767  003750  174556          CMP     #RTA20, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
004026  001401                          BEQ     .+4             ;BRANCH IF OK
004030  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS

004032  022767  000002  174546          CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
004040  001401                          BEQ     .+4             ;BRANCH IF OK
004042  104002                          HLT+2                   ;PS AT FIS TRAP TIME NOT 002

004044  022767  077452  174536          CMP     #077452,ANS3    ;CHECK DATA FROM THE STACK
004052  001401                          BEQ     .+4             ;BRANCH IF OK
004054  104004                          HLT+4                   ;DATA ON STACK (077452) CHANGED

004056  022767  125252  174526          CMP     #125252,ANS4    ;CHECK DATA FROM STACK
004064  001401                          BEQ     .+4             ;BRANCH IF OK
004066  104004                          HLT+4                   ;DATA ON STACK (125252) CHANGED

004070  022767  077652  174516          CMP     #077652,ANS5    ;CHECK DATA FROM STACK
004076  001401                          BEQ     .+4             ;BRANCH IF OK
004100  104006                          HLT+6                   ;DATA ON STACK (077652) CHANGED

004102  022767  125253  174506          CMP     #125253,ANS6    ;CHECK DATA FROM STACK
004110  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
840  004112  104006                          HLT+6                   ;DATA ON STACK (125253) CHANGED
841
842  004114  122767  000020  174656  END22:  CMPB    #20,    ICNT    ;CHECK THE TEST NUMBER
843  004122  001401                          BEQ     .+4             ;BRANCH IF OK
844  004124  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
845
846  004126  104400                          SCOPE
847
848
849                                  ;*************************************************************************
850                                  ;TEST 21:       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
851                                  ;       177520,017552 - 135352,051107 = 177520,017552
852                                  ;       PS = 050,       STACK POINTER = R1
853                                  ;*************************************************************************
854
855  004130  004567  012166          TST21:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
856  004134  135352  051107                  ,WORD   135352,051107   ;SECOND OPERAND ON TOP
857  004140  177520  017552                  ,WORD   177520,017552   ;FIRST OPERAND ON BOTTOM
858  004144  000040                          ,WORD   040             ;PROCESSOR PRIORITY LEVEL
859  004146  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
860  004152  012701  000630                  MOV     #STACK0,R1      ;SET UP STACK POINTER
861
862  004156  000240                          NOP
863  004160  075011                          FSUB+   R1              ;FLOATING SUBTRACT ON THE R1 STACK
864
865  004162  004767  012166                  JSR     PC,     POPR    ;POP THE ANSWER
866  004166  010167  174410                  MOV     R1,     $SP     ;SAVE "STACK POINTER"
867  004172  022767  000050  174400          CMP     #050,   $PSW    ;CHECK PS (EXCEPT T BIT)
868  004200  001401                          BEQ     .+4             ;BRANCH IF OK
869  004202  104000                          HLT                     ;PS NOT EQUAL TO 050
870
871  004204  022767  000634  174370          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
872  004212  001401                          BEQ     .+4             ;BRANCH IF OK
873  004214  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
874
875  004216  022767  177520  174360          CMP     #177520,ANS1    ;CHECK FIRST HALF OF ANSWER
876  004224  001401                          BEQ     .+4             ;BRANCH IF OK
877  004226  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177520
878
879  004230  022767  017552  174350          CMP     #017552,ANS2    ;CHECK SECOND HALF OF ANSWER
880  004236  001401                          BEQ     .+4             ;BRANCH IF OK
881  004240  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 017552
882
883  004242  122767  000021  174530  END21:  CMPB    #21,    ICNT    ;CHECK THE TEST NUMBER
884  004250  001401                          BEQ     .+4             ;BRANCH IF OK
885  004252  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
886
887  004254  104400                          SCOPE
888
```

```
                                ;*****************************************************************
                                ;TEST 221       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
                                ;       125252,125252 - 125252,125253 = 017400,000000
                                ;       PS = 040,       STACK POINTER = R0
                                ;*****************************************************************

004256  004567  012040          TST221: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
004262  125252  125253                  .WORD   125252,125253   ;SECOND OPERAND ON TOP
004266  125252  125252                  .WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
004272  000047                          .WORD   047             ;PROCESSOR PRIORITY LEVEL
004274  016606  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
004300  012700  000630                  MOV     #STACK0,R0      ;SET UP STACK POINTER

004304  000240                          NOP
004306  075010                          FSUB*   R0              ;FLOATING SUBTRACT ON THE R0 STACK

004310  004767  012040                  JSR     PC,     POPR    ;POP THE ANSWER
004314  010067  174262                  MOV     R0,     $SP     ;SAVE "STACK POINTER"
004320  022767  000040  174252          CMP     #040,   $PSW    ;CHECK PS (EXCEPT T BIT)
004326  001401                          BEQ     .+4             ;BRANCH IF OK
004330  104000                          HLT                     ;PS NOT EQUAL TO 040

004332  022767  000634  174242          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
004340  001401                          BEQ     .+4             ;BRANCH IF OK
004342  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4

004344  022767  017400  174232          CMP     #017400,ANS1    ;CHECK FIRST HALF OF ANSWER
004352  001401                          BEQ     .+4             ;BRANCH IF OK
004354  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 017400

004356  005767  174224                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
004362  001401                          BEQ     .+4             ;BRANCH IF OK
004364  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

004366  122767  000022  174404  END221: CMPB    #22,    ICNT    ;CHECK THE TEST NUMBER
004374  001401                          BEQ     .+4             ;BRANCH IF OK
004376  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.

004400  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 23:       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
                                ;       002460,123456 - 100177,177777 = 002460,123456
                                ;       PS = 000,       STACK POINTER = SP
                                ;*****************************************************************

 004402  004567  011536        TST23:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 004406  100177  177777                .WORD   100177,177777   ;SECOND OPERAND ON TOP
 004412  002460  123456                .WORD   002460,123456   ;FIRST OPERAND ON BOTTOM
 004416  000015                        .WORD   015             ;PROCESSOR PRIORITY LEVEL
 004420  016606  000340                .WORD   TRAPER, 340     ;FIS TRAP VECTOR

 004424  000240                        NOP
 004426  075016                        FSUB+   SP              ;FLOATING SUBTRACT ON THE STACK

 004430  004767  011550                JSR     PC,     POPS    ;POP THE ANSWER
 004434  022706  000500                CMP     #500,   SP      ;CHECK THE STACK POINTER
 004440  001404                        BEQ     TSA23           ;BRANCH IF OK
 004442  012706  000500                MOV     #500,   SP      ;RESTORE STACK POINTER
 004446  104000                        HLT                     ;STACK POINTER FOULED UP
 004450  000417                        BR      END23           ;SKIP REST OF TEST

 004452  022767  000000  174120  TSA23:  CMP     #000,   SPSW    ;CHECK PS (EXCEPT T BIT)
 004460  001401                        BEQ     .+4             ;BRANCH IF OK
 004462  104000                        HLT                     ;PS NOT EQUAL TO 000

 004464  022767  002460  174112        CMP     #002460,ANS1    ;CHECK FIRST HALF OF ANSWER
 004472  001401                        BEQ     .+4             ;BRANCH IF OK
 004474  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 002460

 004476  022767  123456  174102        CMP     #123456,ANS2    ;CHECK SECOND HALF OF ANSWER
 004504  001401                        BEQ     .+4             ;BRANCH IF OK
 004506  104002                        HLT+2                   ;ANS2 NOT EQUAL TO 123456

 004510  122767  000023  174262  END23:  CMPB    #23,    ICNT    ;CHECK THE TEST NUMBER
 004516  001401                        BEQ     .+4             ;BRANCH IF OK
 004520  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP!

 004522  104400                        SCOPE
```

```
                                ;*****************************************************************
                                ;TEST 241       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
                                ;       000425,052525 - 000252,125252 = 000200,000000
                                ;       PS = 200,       STACK POINTER = R4
                                ;*****************************************************************

004524  004567  011572          TST241: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
004530  000252  125252                  ,WORD   000252,125252   ;SECOND OPERAND ON TOP
004534  000425  052525                  ,WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
004540  000217                          ,WORD   217             ;PROCESSOR PRIORITY LEVEL
004542  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
004546  012704  000630                  MOV     #STACK0,R4      ;SET UP STACK POINTER

004552  000240                          NOP
004554  075014                          FSUB+   R4              ;FLOATING SUBTRACT ON THE R4 STACK

004556  004767  011572                  JSR     PC,     POPR    ;POP THE ANSWER
004562  010467  174014                  MOV     R4,     $SP     ;SAVE "STACK POINTER"
004566  022767  000200  174004          CMP     #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
004574  001401                          BEQ     .+4             ;BRANCH IF OK
004576  104000                          HLT                     ;PS NOT EQUAL TO 200

004600  022767  000634  173774          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
004606  001401                          BEQ     .+4             ;BRANCH IF OK
004610  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4

004612  022767  000200  173764          CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
004620  001401                          BEQ     .+4             ;BRANCH IF OK
004622  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000200

004624  005767  173756                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
004630  001401                          BEQ     .+4             ;BRANCH IF OK
004632  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

004634  122767  000024  174136  END241: CMPB    #24,    ICNT    ;CHECK THE TEST NUMBER
004642  001401                          BEQ     .+4             ;BRANCH IF OK
004644  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP,

004646  104400                          SCOPE
```

```
1012
1013                                    ;*****************************************************************
1014                                    ;TEST 25:       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
1015                                    ;       000425,052525 - 000252,125253 ==> UNDERFLOW
1016                                    ;       PS(ON STACK) = 252,     STACK POINTER = SP
1017                                    ;*****************************************************************
1018
1019  004650  004567  011270           TST25:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1020  004654  000252  125253                   ,WORD   000252,125253   ;SECOND OPERAND ON TOP
1021  004660  000425  052525                   ,WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
1022  004664  000257                           ,WORD   257             ;PROCESSOR PRIORITY LEVEL
1023  004666  004712  000340                   ,WORD   ISR25,  340     ;FIS TRAP VECTOR
1024
1025  004672  000240                           NOP
1026  004674  075016                           FSUB+   SP              ;FLOATING SUBTRACT ON THE STACK
1027
1028  004676  004767  011302           RTA25:  JSR     %7,     POPS    ;POP THE "ANSWER"
1029  004702  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1030  004704  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK POINTER
1031  004710  000454                           BR      END25
1032
1033  004712  004767  011322           ISR25:  JSR     %7,     POPES   ;POP ALL DATA OFF THE STACK
1034  004716  022706  000500                   CMP     #500,   SP      ;CHECK THE STACK POINTER
1035  004722  001404                           BEQ     ISA25           ;BRANCH IF OK
1036  004724  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK POINTER
1037  004730  104000                           HLT                     ;STACK POINTER FOULED UP
1038  004732  000443                           BR      END25           ;SKIP REST OF TEST
1039
1040  004734  022767  000340  173636   ISA25:  CMP     #340,   $PSW    ;CHECK PS AFTER FIS TRAP
1041  004742  001401                           BEQ     .+4             ;BRANCH IF OK
1042  004744  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 340
1043
1044  004746  022767  004676  173630           CMP     #RTA25, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1045  004754  001401                           BEQ     .+4             ;BRANCH IF OK
1046  004756  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1047
1048  004760  022767  000252  173620           CMP     #252,   ANS2    ;CHECK PS BEFORE FIS TRAP
1049  004766  001401                           BEQ     .+4             ;BRANCH IF OK
1050  004770  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 252
1051
1052  004772  022767  000252  173610           CMP     #000252,ANS3    ;CHECK DATA FROM THE STACK
1053  005000  001401                           BEQ     .+4             ;BRANCH IF OK
1054  005002  104004                           HLT+4                   ;DATA ON STACK (000252) CHANGED
1055
1056  005004  022767  125253  173600           CMP     #125253,ANS4    ;CHECK DATA FROM STACK
1057  005012  001401                           BEQ     .+4             ;BRANCH IF OK
1058  005014  104004                           HLT+4                   ;DATA ON STACK (125253) CHANGED
1059
1060  005016  022767  000425  173570           CMP     #000425,ANS5    ;CHECK DATA FROM STACK
1061  005024  001401                           BEQ     .+4             ;BRANCH IF OK
1062  005026  104006                           HLT+6                   ;DATA ON STACK (000425) CHANGED
1063
1064  005030  022767  052525  173560           CMP     #052525,ANS6    ;CHECK DATA FROM STACK
1065  005036  001401                           BEQ     .+4             ;BRANCH IF OK
```

```
1066  005040  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
1067
1068  005042  122767  000025  173730  END25:  CMPB    #25,    ICNT    ;CHECK THE TEST NUMBER
1069  005050  001401                          BEQ     .+4             ;BRANCH IF OK
1070  005052  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1071
1072  005054  104400                          SCOPE
1073
1074
1075                                  ;*************************************************************************
1076                                  ;TEST 261       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
1077                                  ;       177452,125252 - 077652,125252 = 177777,177777
1078                                  ;       PS = 350,       STACK POINTER = SP
1079                                  ;*************************************************************************
1080
1081  005056  004567  011062          TST26:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1082  005062  077652  125252                  ,WORD   077652,125252   ;SECOND OPERAND ON TOP
1083  005066  177452  125252                  ,WORD   177452,125252   ;FIRST OPERAND ON BOTTOM
1084  005072  000357                          ,WORD   357             ;PROCESSOR PRIORITY LEVEL
1085  005074  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1086
1087  005100  000240                          NOP
1088  005102  075016                          FSUB+   SP              ;FLOATING SUBTRACT ON THE STACK
1089
1090  005104  004767  011074                  JSR     PC,     POPS    ;POP THE ANSWER
1091  005110  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
1092  005114  001404                          BEQ     TSA26           ;BRANCH IF OK
1093  005116  012706  000500                  MOV     #500,   SP      ;RESTORE STACK POINTER
1094  005122  104000                          HLT                     ;STACK POINTER FOULED UP
1095  005124  000417                          BR      END26           ;SKIP REST OF TEST
1096
1097  005126  022767  000350  173444  TSA26:  CMP     #350,   SPSW    ;CHECK PS (EXCEPT T BIT)
1098  005134  001401                          BEQ     .+4             ;BRANCH IF OK
1099  005136  104000                          HLT                     ;PS NOT EQUAL TO 350
1100
1101  005140  022767  177777  173436          CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
1102  005146  001401                          BEQ     .+4             ;BRANCH IF OK
1103  005150  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777
1104
1105  005152  022767  177777  173426          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
1106  005160  001401                          BEQ     .+4             ;BRANCH IF OK
1107  005162  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777
1108
1109  005164  122767  000026  173606  END26:  CMPB    #26,    ICNT    ;CHECK THE TEST NUMBER
1110  005172  001401                          BEQ     .+4             ;BRANCH IF OK
1111  005174  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1112
1113  005176  104400                          SCOPE
1114
```

```
1115
1116                                    ;*****************************************************************
1117                                    ;TEST 27:       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
1118                                    ;       077652,125253 - 177452,125252 ==> OVERFLOW
1119                                    ;       PS(ON STACK) = 002,     STACK POINTER = R3
1120                                    ;*****************************************************************
1121
1122  005200  004567  011116           TST27:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
1123  005204  177452  125252                   ,WORD   177452,125252   ;SECOND OPERAND ON TOP
1124  005210  077652  125253                   ,WORD   077652,125253   ;FIRST OPERAND ON BOTTOM
1125  005214  000015                           ,WORD   015             ;PROCESSOR PRIORITY LEVEL
1126  005216  005246  000344                   ,WORD   ISR27,  344     ;FIS TRAP VECTOR
1127  005222  012703  000630                   MOV     #STACK0,R3      ;SET UP R3 AS STACK POINTER
1128
1129  005226  000240                           NOP
1130  005230  075013                           FSUB+   R3              ;FLOATING SUBTRACT ON THE R3 STACK
1131
1132  005232  004767  011116           RTA27:  JSR     %7,     POPR    ;POP THE "ANSWER"
1133  005236  010367  173340                   MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
1134  005242  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1135  005244  000454                           BR      END27
1136
1137  005246  004767  011134           ISR27:  JSR     %7,     POPER   ;POP ALL DATA OFF THE STACKS
1138  005252  010367  173324                   MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
1139  005256  022767  000344  173314           CMP     #344,   $PSW    ;CHECK PS AFTER FIS TRAP
1140  005264  001401                           BEQ     ,+4             ;BRANCH IF OK
1141  005266  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 344
1142
1143  005270  022767  000630  173304           CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R3)
1144  005276  001401                           BEQ     ,+4             ;BRANCH IF OK
1145  005300  104000                           HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK0
1146
1147  005302  022767  005232  173274           CMP     #RTA27, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1148  005310  001401                           BEQ     ,+4             ;BRANCH IF OK
1149  005312  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1150
1151  005314  022767  000002  173264           CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
1152  005322  001401                           BEQ     ,+4             ;BRANCH IF OK
1153  005324  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 002
1154
1155  005326  022767  177452  173254           CMP     #177452,ANS3    ;CHECK DATA FROM THE STACK
1156  005334  001401                           BEQ     ,+4             ;BRANCH IF OK
1157  005336  104004                           HLT+4                   ;DATA ON STACK (177452) CHANGED
1158
1159  005340  022767  125252  173244           CMP     #125252,ANS4    ;CHECK DATA FROM STACK
1160  005346  001401                           BEQ     ,+4             ;BRANCH IF OK
1161  005350  104004                           HLT+4                   ;DATA ON STACK (125252) CHANGED
1162
1163  005352  022767  077652  173234           CMP     #077652,ANS5    ;CHECK DATA FROM STACK
1164  005360  001401                           BEQ     ,+4             ;BRANCH IF OK
1165  005362  104006                           HLT+6                   ;DATA ON STACK (077652) CHANGED
1166
1167  005364  022767  125253  173224           CMP     #125253,ANS6    ;CHECK DATA FROM STACK
1168  005372  001401                           BEQ     ,+4             ;BRANCH IF OK
```

```
1169  005374  104006                          HLT+6                   ;DATA ON STACK (125253) CHANGED
1170
1171  005376  122767  000027  173374  END27:  CMPB    #27,    ICNT    ;CHECK THE TEST NUMBER
1172  005404  001401                          BEQ     .+4             ;BRANCH IF OK
1173  005406  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1174
1175  005410  104400                          SCOPE
1176
1177
1178                                  ;*****************************************************************
1179                                  ;TEST 30:       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
1180                                  ;       035152,125252 - 043125,052525 = 143125,052524
1181                                  ;       PS = 150,       STACK POINTER = R3
1182                                  ;*****************************************************************
1183
1184  005412  004567  010704          TST30:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
1185  005416  043125  052525                  .WORD   043125,052525   ;SECOND OPERAND ON TOP
1186  005422  035152  125252                  .WORD   035152,125252   ;FIRST OPERAND ON BOTTOM
1187  005426  000147                          .WORD   147             ;PROCESSOR PRIORITY LEVEL
1188  005430  016606  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1189  005434  012703  000630                  MOV     #STACK0,R3      ;SET UP STACK POINTER
1190
1191  005440  000240                          NOP
1192  005442  075013                          FSUB+   R3              ;FLOATING SUBTRACT ON THE R3 STACK
1193
1194  005444  004767  010704                  JSR     PC,     POPR    ;POP THE ANSWER
1195  005450  010367  173126                  MOV     R3,     $SP     ;SAVE "STACK POINTER"
1196  005454  022767  000150  173116          CMP     #150,   $PSW    ;CHECK PS (EXCEPT T BIT)
1197  005462  001401                          BEQ     .+4             ;BRANCH IF OK
1198  005464  104000                          HLT                     ;PS NOT EQUAL TO 150
1199
1200  005466  022767  000634  173106          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
1201  005474  001401                          BEQ     .+4             ;BRANCH IF OK
1202  005476  104000                          HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
1203
1204  005500  022767  143125  173076          CMP     #143125,ANS1    ;CHECK FIRST HALF OF ANSWER
1205  005506  001401                          BEQ     .+4             ;BRANCH IF OK
1206  005510  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 143125
1207
1208  005512  022767  052524  173066          CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
1209  005520  001401                          BEQ     .+4             ;BRANCH IF OK
1210  005522  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 052524
1211
1212  005524  122767  000030  173246  END30:  CMPB    #30,    ICNT    ;CHECK THE TEST NUMBER
1213  005532  001401                          BEQ     .+4             ;BRANCH IF OK
1214  005534  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1215
1216  005536  104400                          SCOPE
1217
```

```
1218
1219                                    ;*****************************************************************
1220                                    ;TEST 31:       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
1221                                    ;       143325,052525 - 135152,125252 = 143325,052525
1222                                    ;       PS = 250,       STACK POINTER = R0
1223                                    ;*****************************************************************
1224
1225  005540  004567  010556           TST31:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
1226  005544  135152  125252                   ,WORD   135152,125252   ;SECOND OPERAND ON TOP
1227  005550  143325  052525                   ,WORD   143325,052525   ;FIRST OPERAND ON BOTTOM
1228  005554  000243                           ,WORD   243             ;PROCESSOR PRIORITY LEVEL
1229  005556  016606  000340                   ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1230  005562  012700  000630                   MOV     #STACK0,R0      ;SET UP STACK POINTER
1231
1232  005566  000240                           NOP
1233  005570  075010                           FSUB+   R0              ;FLOATING SUBTRACT ON THE R0 STACK
1234
1235  005572  004767  010556                   JSR     PC,     POPR    ;POP THE ANSWER
1236  005576  010067  173000                   MOV     R0,     $SP     ;SAVE "STACK POINTER"
1237  005602  022767  000250  172770           CMP     #250,   $PSW    ;CHECK PS (EXCEPT T BIT)
1238  005610  001401                           BEQ     .+4             ;BRANCH IF OK
1239  005612  104000                           HLT                     ;PS NOT EQUAL TO 250
1240
1241  005614  022767  000634  172760           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
1242  005622  001401                           BEQ     .+4             ;BRANCH IF OK
1243  005624  104000                           HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
1244
1245  005626  022767  143325  172750           CMP     #143325,ANS1    ;CHECK FIRST HALF OF ANSWER
1246  005634  001401                           BEQ     .+4             ;BRANCH IF OK
1247  005636  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 143325
1248
1249  005640  022767  052525  172740           CMP     #052525,ANS2    ;CHECK SECOND HALF OF ANSWER
1250  005646  001401                           BEQ     .+4             ;BRANCH IF OK
1251  005650  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 052525
1252
1253  005652  122767  000031  173120  END31:  CMPB    #31,    ICNT    ;CHECK THE TEST NUMBER
1254  005660  001401                           BEQ     .+4             ;BRANCH IF OK
1255  005662  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1256
1257  005664  104400                           SCOPE
1258
```

```
1259
1260                                    ;*****************************************************************
1261                                    ;TEST 321       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
1262                                    ;       135152,125252 - 143325,052525 = 043325,052525
1263                                    ;       PS = 340,       STACK POINTER = R5
1264                                    ;*****************************************************************
1265
1266  005666  004567  010430           TST32:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
1267  005672  143325  052525                   ,WORD   143325,052525   ;SECOND OPERAND ON TOP
1268  005676  135152  125252                   ,WORD   135152,125252   ;FIRST OPERAND ON BOTTOM
1269  005702  000357                           ,WORD   357             ;PROCESSOR PRIORITY LEVEL
1270  005704  016606  000340                   ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1271  005710  012705  000630                   MOV     #STACK0,R5      ;SET UP STACK POINTER
1272
1273  005714  000240                           NOP
1274  005716  075015                           FSUB+   R5              ;FLOATING SUBTRACT ON THE R5 STACK
1275
1276  005720  004767  010430                   JSR     PC,     POPR    ;POP THE ANSWER
1277  005724  010567  172652                   MOV     R5,     $SP     ;SAVE "STACK POINTER"
1278  005730  022767  000340  172642           CMP     #340,   $PSW    ;CHECK PS (EXCEPT T BIT)
1279  005736  001401                           BEQ     .+4             ;BRANCH IF OK
1280  005740  104000                           HLT                     ;PS NOT EQUAL TO 340
1281
1282  005742  022767  000634  172632           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
1283  005750  001401                           BEQ     .+4             ;BRANCH IF OK
1284  005752  104000                           HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
1285
1286  005754  022767  043325  172622           CMP     #043325,ANS1    ;CHECK FIRST HALF OF ANSWER
1287  005762  001401                           BEQ     .+4             ;BRANCH IF OK
1288  005764  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 043325
1289
1290  005766  022767  052525  172612           CMP     #052525,ANS2    ;CHECK SECOND HALF OF ANSWER
1291  005774  001401                           BEQ     .+4             ;BRANCH IF OK
1292  005776  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 052525
1293
1294  006000  122767  000032  172772  END32:  CMPB    #32,    ICNT    ;CHECK THE TEST NUMBER
1295  006006  001401                           BEQ     .+4             ;BRANCH IF OK
1296  006010  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP,
1297
1298  006012  104400                           SCOPE
1299
```

```
1300
1301                                    ;*****************************************************************
1302                                    ;TEST 331       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
1303                                    ;       043125,052525 - 035152,125252 = 043125,052524
1304                                    ;       PS = 040,       STACK POINTER = R2
1305                                    ;*****************************************************************
1306
1307  006014  004567  010302           TST331: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
1308  006020  035152  125252                   ,WORD   035152,125252   ;SECOND OPERAND ON TOP
1309  006024  043125  052525                   ,WORD   043125,052525   ;FIRST OPERAND ON BOTTOM
1310  006030  000040                           ,WORD   040             ;PROCESSOR PRIORITY LEVEL
1311  006032  016606  000340                   ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1312  006036  012702  000630                   MOV     #STACK0,R2      ;SET UP STACK POINTER
1313
1314  006042  000240                           NOP
1315  006044  075012                           FSUB+   R2              ;FLOATING SUBTRACT ON THE R2 STACK
1316
1317  006046  004767  010302                   JSR     PC,     POPR    ;POP THE ANSWER
1318  006052  010267  172524                   MOV     R2,     $SP     ;SAVE "STACK POINTER"
1319  006056  022767  000040  172514           CMP     #040,   SPSW    ;CHECK PS (EXCEPT T BIT)
1320  006064  001401                           BEQ     .+4             ;BRANCH IF OK
1321  006066  104000                           HLT                     ;PS NOT EQUAL TO 040
1322
1323  006070  022767  000634  172504           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
1324  006076  001401                           BEQ     .+4             ;BRANCH IF OK
1325  006100  104000                           HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
1326
1327  006102  022767  043125  172474           CMP     #043125,ANS1    ;CHECK FIRST HALF OF ANSWER
1328  006110  001401                           BEQ     .+4             ;BRANCH IF OK
1329  006112  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 043125
1330
1331  006114  022767  052524  172464           CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
1332  006122  001401                           BEQ     .+4             ;BRANCH IF OK
1333  006124  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 052524
1334
1335  006126  122767  000033  172644  END331: CMPB    #33,    ICNT    ;CHECK THE TEST NUMBER
1336  006134  001401                           BEQ     .+4             ;BRANCH IF OK
1337  006136  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1338
1339  006140  104400                           SCOPE
1340
```

```
1341
1342                                    ;*************************************************************************
1343                                    ;TEST 341       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
1344                                    ;       000000,000000 * 000000,000000 = 000000,000000
1345                                    ;       PS = 104,       STACK POINTER = R4
1346                                    ;*************************************************************************
1347
1348  006142  004567  010154           TST341: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
1349  006146  000000  000000                   ,WORD   000000,000000   ;SECOND OPERAND ON TOP
1350  006152  000000  000000                   ,WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
1351  006156  000111                           ,WORD   111             ;PROCESSOR PRIORITY LEVEL
1352  006160  016606  000340                   ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1353  006164  012704  000630                   MOV     #STACK0,R4      ;SET UP STACK POINTER
1354
1355  006170  000240                           NOP
1356  006172  075024                           FMUL*   R4              ;FLOATING MULTIPLY ON THE R4 STACK
1357
1358  006174  004767  010154                   JSR     PC,     POPR    ;POP THE ANSWER
1359  006200  010467  172376                   MOV     R4,     $SP     ;SAVE "STACK POINTER"
1360  006204  022767  000104  172366           CMP     #104,   $PSW    ;CHECK PS (EXCEPT T BIT)
1361  006212  001401                           BEQ     .+4             ;BRANCH IF OK
1362  006214  104000                           HLT                     ;PS NOT EQUAL TO 104
1363
1364  006216  022767  000634  172356           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
1365  006224  001401                           BEQ     .+4             ;BRANCH IF OK
1366  006226  104000                           HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
1367
1368  006230  005767  172350                   TST     ANS1            ;CHECK FIRST HALF OF ANSWER
1369  006234  001401                           BEQ     .+4             ;BRANCH IF OK
1370  006236  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 000000
1371
1372  006240  005767  172342                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
1373  006244  001401                           BEQ     .+4             ;BRANCH IF OK
1374  006246  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
1375
1376  006250  122767  000034  172522  END341: CMPB    #34,    ICNT    ;CHECK THE TEST NUMBER
1377  006256  001401                           BEQ     .+4             ;BRANCH IF OK
1378  006260  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1379
1380  006262  104400                           SCOPE
1381
```

```
1382
1383                                    ;*****************************************************************
1384                                    ;TEST 35:       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
1385                                    ;       140200,000000 * 052345,123456 = 152345,123456
1386                                    ;       PS = 350,       STACK POINTER = R2
1387                                    ;*****************************************************************
1388
1389  006264  004567  010032           TST35:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
1390  006270  052345  123456                   ,WORD   052345,123456   ;SECOND OPERAND ON TOP
1391  006274  140200  000000                   ,WORD   140200,000000   ;FIRST OPERAND ON BOTTOM
1392  006300  000343                           ,WORD   343             ;PROCESSOR PRIORITY LEVEL
1393  006302  016606  000340                   ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1394  006306  012702  000630                   MOV     #STACK0,R2      ;SET UP STACK POINTER
1395
1396  006312  000240                           NOP
1397  006314  075022                           FMUL+   R2              ;FLOATING MULTIPLY ON THE R2 STACK
1398
1399  006316  004767  010032                   JSR     PC,     POPR    ;POP THE ANSWER
1400  006322  010267  172254                   MOV     R2,     SSP     ;SAVE "STACK POINTER"
1401  006326  022767  000350  172244           CMP     #350,   SPSW    ;CHECK PS (EXCEPT T BIT)
1402  006334  001401                           BEQ     .+4             ;BRANCH IF OK
1403  006336  104000                           HLT                     ;PS NOT EQUAL TO 350
1404
1405  006340  022767  000634  172234           CMP     #STACK4,SSP     ;CHECK THE STACK POINTER (R2)
1406  006346  001401                           BEQ     .+4             ;BRANCH IF OK
1407  006350  104000                           HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
1408
1409  006352  022767  152345  172224           CMP     #152345,ANS1    ;CHECK FIRST HALF OF ANSWER
1410  006360  001401                           BEQ     .+4             ;BRANCH IF OK
1411  006362  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 152345
1412
1413  006364  022767  123456  172214           CMP     #123456,ANS2    ;CHECK SECOND HALF OF ANSWER
1414  006372  001401                           BEQ     .+4             ;BRANCH IF OK
1415  006374  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 123456
1416
1417  006376  122767  000035  172374  END35:  CMPB    #35,    ICNT    ;CHECK THE TEST NUMBER
1418  006404  001401                           BEQ     .+4             ;BRANCH IF OK
1419  006406  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1420
1421  006410  104400                           SCOPE
1422
```

```
1423
1424                                    ;*****************************************************************
1425                                    ;TEST 36:       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
1426                                    ;       100125,052525 * 135753,024642 = 000000,000000
1427                                    ;       PS = 104,       STACK POINTER = R5
1428                                    ;*****************************************************************
1429
1430  006412  004567  007704           TST36:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
1431  006416  135753  024642                   ,WORD   135753,024642   ;SECOND OPERAND ON TOP
1432  006422  100125  052525                   ,WORD   100125,052525   ;FIRST OPERAND ON BOTTOM
1433  006426  000117                           ,WORD   117             ;PROCESSOR PRIORITY LEVEL
1434  006430  016606  000340                   ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1435  006434  012705  000630                   MOV     #STACK0,R5      ;SET UP STACK POINTER
1436
1437  006440  000240                           NOP
1438  006442  075025                           FMUL+   R5              ;FLOATING MULTIPLY ON THE R5 STACK
1439
1440  006444  004767  007704                   JSR     PC,     POPR    ;POP THE ANSWER
1441  006450  010567  172126                   MOV     R5,     SSP     ;SAVE "STACK POINTER"
1442  006454  022767  000104  172116           CMP     #104,   SPSW    ;CHECK PS (EXCEPT T BIT)
1443  006462  001401                           BEQ     .+4             ;BRANCH IF OK
1444  006464  104000                           HLT                     ;PS NOT EQUAL TO 104
1445
1446  006466  022767  000634  172106           CMP     #STACK4,SSP     ;CHECK THE STACK POINTER (R5)
1447  006474  001401                           BEQ     .+4             ;BRANCH IF OK
1448  006476  104000                           HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
1449
1450  006500  005767  172100                   TST     ANS1            ;CHECK FIRST HALF OF ANSWER
1451  006504  001401                           BEQ     .+4             ;BRANCH IF OK
1452  006506  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 000000
1453
1454  006510  005767  172072                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
1455  006514  001401                           BEQ     .+4             ;BRANCH IF OK
1456  006516  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
1457
1458  006520  122767  000036  172252  END36:  CMPB    #36,    ICNT    ;CHECK THE TEST NUMBER
1459  006526  001401                           BEQ     .+4             ;BRANCH IF OK
1460  006530  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1461
1462  006532  104400                           SCOPE
1463
```

```
1464
1465                                    ;*****************************************************************
1466                                    ;TEST 371       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
1467                                    ;       161616,161616 * 000052,125252 = 000000,000000
1468                                    ;       PS = 204,       STACK POINTER = R3
1469                                    ;*****************************************************************
1470
1471  006534  004567  007562           TST371: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
1472  006540  000052  125252                   ,WORD   000052,125252   ;SECOND OPERAND ON TOP
1473  006544  161616  161616                   ,WORD   161616,161616   ;FIRST OPERAND ON BOTTOM
1474  006550  000217                           ,WORD   217             ;PROCESSOR PRIORITY LEVEL
1475  006552  016606  000340                   ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1476  006556  012703  000630                   MOV     #STACK0,R3      ;SET UP STACK POINTER
1477
1478  006562  000240                           NOP
1479  006564  075023                           FMUL*   R3              ;FLOATING MULTIPLY ON THE R3 STACK
1480
1481  006566  004767  007562                   JSR     PC,     POPR    ;POP THE ANSWER
1482  006572  010367  172004                   MOV     R3,     $SP     ;SAVE "STACK POINTER"
1483  006576  022767  000204  171774           CMP     #204,   $PSW    ;CHECK PS (EXCEPT T BIT)
1484  006604  001401                           BEQ     .+4             ;BRANCH IF OK
1485  006606  104000                           HLT                     ;PS NOT EQUAL TO 204
1486
1487  006610  022767  000634  171764           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
1488  006616  001401                           BEQ     .+4             ;BRANCH IF OK
1489  006620  104000                           HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
1490
1491  006622  005767  171756                   TST     ANS1            ;CHECK FIRST HALF OF ANSWER
1492  006626  001401                           BEQ     .+4             ;BRANCH IF OK
1493  006630  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 000000
1494
1495  006632  005767  171750                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
1496  006636  001401                           BEQ     .+4             ;BRANCH IF OK
1497  006640  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
1498
1499  006642  122767  000037  172130  END371: CMPB    #37,    ICNT    ;CHECK THE TEST NUMBER
1500  006650  001401                           BEQ     .+4             ;BRANCH IF OK
1501  006652  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1502
1503  006654  104400                           SCOPE
1504
```

```
1505
1506                                    ;*****************************************************************
1507                                    ;TEST 401       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
1508                                    ;       176452,125252 * 041500,000000 = 177777,177777
1509                                    ;       PS = 350,       STACK POINTER = SP
1510                                    ;*****************************************************************
1511
1512  006656  004567  007262           TST401: JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1513  006662  041500  000000                   ,WORD   041500,000000   ;SECOND OPERAND ON TOP
1514  006666  176452  125252                   ,WORD   176452,125252   ;FIRST OPERAND ON BOTTOM
1515  006672  000357                           ,WORD   357             ;PROCESSOR PRIORITY LEVEL
1516  006674  016606  000340                   ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1517
1518  006700  000240                           NOP
1519  006702  075026                           FMUL+   SP              ;FLOATING MULTIPLY ON THE STACK
1520
1521  006704  004767  007274                   JSR     PC,     POPS    ;POP THE ANSWER
1522  006710  022706  000500                   CMP     #500,   SP      ;CHECK THE STACK POINTER
1523  006714  001404                           BEQ     TSA40           ;BRANCH IF OK
1524  006716  012706  000500                   MOV     #500,   SP      ;RESTORE STACK POINTER
1525  006722  104000                           HLT                     ;STACK POINTER FOULED UP
1526  006724  000417                           BR      END40           ;SKIP REST OF TEST
1527
1528  006726  022767  000350  171644  TSA401: CMP     #350,   SPSW    ;CHECK PS (EXCEPT T BIT)
1529  006734  001401                           BEQ     .+4             ;BRANCH IF OK
1530  006736  104000                           HLT                     ;PS NOT EQUAL TO 350
1531
1532  006740  022767  177777  171636           CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
1533  006746  001401                           BEQ     .+4             ;BRANCH IF OK
1534  006750  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 177777
1535
1536  006752  022767  177777  171626           CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
1537  006760  001401                           BEQ     .+4             ;BRANCH IF OK
1538  006762  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 177777
1539
1540  006764  122767  000040  172006  END401: CMPB    #40,    ICNT    ;CHECK THE TEST NUMBER
1541  006772  001401                           BEQ     .+4             ;BRANCH IF OK
1542  006774  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1543
1544  006776  104400                           SCOPE
1545
```

```
1546
1547                                    ;*****************************************************************
1548                                    ;TEST 41:       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
1549                                    ;       076452,125252 * 041500,000001 ==> OVERFLOW
1550                                    ;       PS(ON STACK) = 102,     STACK POINTER = SP
1551                                    ;*****************************************************************
1552
1553  007000  004567  007140           TST41:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1554  007004  041500  000001                   ,WORD   041500,000001   ;SECOND OPERAND ON TOP
1555  007010  076452  125252                   ,WORD   076452,125252   ;FIRST OPERAND ON BOTTOM
1556  007014  000105                           ,WORD   105             ;PROCESSOR PRIORITY LEVEL
1557  007016  007042  000357                   ,WORD   ISR41,  357     ;FIS TRAP VECTOR
1558
1559  007022  000240                           NOP
1560  007024  075026                           FMUL+   SP              ;FLOATING MULTIPLY ON THE STACK
1561
1562  007026  004767  007152           RTA41:  JSR     %7,     POPS    ;POP THE "ANSWER"
1563  007032  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1564  007034  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK POINTER
1565  007040  000454                           BR      END41
1566
1567  007042  004767  007172           ISR41:  JSR     %7,     POPES   ;POP ALL DATA OFF THE STACK
1568  007046  022706  000500                   CMP     #500,   SP      ;CHECK THE STACK POINTER
1569  007052  001404                           BEQ     ISA41           ;BRANCH IF OK
1570  007054  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK POINTER
1571  007060  104000                           HLT                     ;STACK POINTER FOULED UP
1572  007062  000443                           BR      END41           ;SKIP REST OF TEST
1573
1574  007064  022767  000357  171506  ISA41:  CMP     #357,   SPSW    ;CHECK PS AFTER FIS TRAP
1575  007072  001401                           BEQ     .+4             ;BRANCH IF OK
1576  007074  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 357
1577
1578  007076  022767  007026  171500           CMP     #RTA41, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1579  007104  001401                           BEQ     .+4             ;BRANCH IF OK
1580  007106  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1581
1582  007110  022767  000102  171470           CMP     #102,   ANS2    ;CHECK PS BEFORE FIS TRAP
1583  007116  001401                           BEQ     .+4             ;BRANCH IF OK
1584  007120  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 102
1585
1586  007122  022767  041500  171460           CMP     #041500,ANS3    ;CHECK DATA FROM THE STACK
1587  007130  001401                           BEQ     .+4             ;BRANCH IF OK
1588  007132  104004                           HLT+4                   ;DATA ON STACK (041500) CHANGED
1589
1590  007134  022767  000001  171450           CMP     #000001,ANS4    ;CHECK DATA FROM STACK
1591  007142  001401                           BEQ     .+4             ;BRANCH IF OK
1592  007144  104004                           HLT+4                   ;DATA ON STACK (000001) CHANGED
1593
1594  007146  022767  076452  171440           CMP     #076452,ANS5    ;CHECK DATA FROM STACK
1595  007154  001401                           BEQ     .+4             ;BRANCH IF OK
1596  007156  104006                           HLT+6                   ;DATA ON STACK (076452) CHANGED
1597
1598  007160  022767  125252  171430           CMP     #125252,ANS6    ;CHECK DATA FROM STACK
1599  007166  001401                           BEQ     .+4             ;BRANCH IF OK
```

```
1600  007170  104006                          HLT+6                   ;DATA ON STACK (125252) CHANGED
1601
1602  007172  122767  000041  171600  END41:  CMPB    #41,    ICNT    ;CHECK THE TEST NUMBER
1603  007200  001401                          BEQ     .+4             ;BRANCH IF OK
1604  007202  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1605
1606  007204  104400                          SCOPE
1607
1608
1609                                  ;*****************************************************************
1610                                  ;TEST 42:       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
1611                                  ;       124252,125252 * 114100,000001 = 000200,000000
1612                                  ;       PS = 200,       STACK POINTER = R1
1613                                  ;*****************************************************************
1614
1615  007206  004567  007110          TST42:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
1616  007212  114100  000001                  .WORD   114100,000001   ;SECOND OPERAND ON TOP
1617  007216  124252  125252                  .WORD   124252,125252   ;FIRST OPERAND ON BOTTOM
1618  007222  000200                          .WORD   200             ;PROCESSOR PRIORITY LEVEL
1619  007224  016606  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1620  007230  012701  000630                  MOV     #STACK0,R1      ;SET UP STACK POINTER
1621
1622  007234  000240                          NOP
1623  007236  075021                          FMUL+   R1              ;FLOATING MULTIPLY ON THE R1 STACK
1624
1625  007240  004767  007110                  JSR     PC,     POPR    ;POP THE ANSWER
1626  007244  010167  171332                  MOV     R1,     $SP     ;SAVE "STACK POINTER"
1627  007250  022767  000200  171322          CMP     #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
1628  007256  001401                          BEQ     .+4             ;BRANCH IF OK
1629  007260  104000                          HLT                     ;PS NOT EQUAL TO 200
1630
1631  007262  022767  000634  171312          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
1632  007270  001401                          BEQ     .+4             ;BRANCH IF OK
1633  007272  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
1634
1635  007274  022767  000200  171302          CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
1636  007302  001401                          BEQ     .+4             ;BRANCH IF OK
1637  007304  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000200
1638
1639  007306  005767  171274                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
1640  007312  001401                          BEQ     .+4             ;BRANCH IF OK
1641  007314  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
1642
1643  007316  122767  000042  171454  END42:  CMPB    #42,    ICNT    ;CHECK THE TEST NUMBER
1644  007324  001401                          BEQ     .+4             ;BRANCH IF OK
1645  007326  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1646
1647  007330  104400                          SCOPE
1648
```

```
1649
1650                                    ;*****************************************************************
1651                                    ;TEST 43:       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
1652                                    ;       024252,125252 * 114100,000000 ==> UNDERFLOW
1653                                    ;       PS(ON STACK) = 312,     STACK POINTER = R0
1654                                    ;*****************************************************************
1655
1656  007332  004567  006764           TST43:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
1657  007336  114100  000000                   ,WORD   114100,000000   ;SECOND OPERAND ON TOP
1658  007342  024252  125252                   ,WORD   024252,125252   ;FIRST OPERAND ON BOTTOM
1659  007346  000305                           ,WORD   305             ;PROCESSOR PRIORITY LEVEL
1660  007350  007400  000057                   ,WORD   ISR43,  057     ;FIS TRAP VECTOR
1661  007354  012700  000630                   MOV     #STACK0,R0      ;SET UP R0 AS STACK POINTER
1662
1663  007360  000240                           NOP
1664  007362  075020                           FMUL+   R0              ;FLOATING MULTIPLY ON THE R0 STACK
1665
1666  007364  004767  006764           RTA43:  JSR     X7,     POPR    ;POP THE "ANSWER"
1667  007370  010067  171206                   MOV     R0,     SSP     ;SAVE STACK POINTER (R0)
1668  007374  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1669  007376  000453                           BR      END43
1670
1671  007400  004767  007002           ISR43:  JSR     X7,     POPER   ;POP ALL DATA OFF THE STACKS
1672  007404  010067  171172                   MOV     R0,     SSP     ;SAVE STACK POINTER (R0)
1673  007410  022767  000057  171162           CMP     #057,   SPSW    ;CHECK PS AFTER FIS TRAP
1674  007416  001401                           BEQ     .+4             ;BRANCH IF OK
1675  007420  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 057
1676
1677  007422  022767  000630  171152           CMP     #STACK0,SSP     ;CHECK THE STACK POINTER (R0)
1678  007430  001401                           BEQ     .+4             ;BRANCH IF OK
1679  007432  104000                           HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK0
1680
1681  007434  022767  007364  171142           CMP     #RTA43, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1682  007442  001401                           BEQ     .+4             ;BRANCH IF OK
1683  007444  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1684
1685  007446  022767  000312  171132           CMP     #312,   ANS2    ;CHECK PS BEFORE FIS TRAP
1686  007454  001401                           BEQ     .+4             ;BRANCH IF OK
1687  007456  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 312
1688
1689  007460  022767  114100  171122           CMP     #114100,ANS3    ;CHECK DATA FROM THE STACK
1690  007466  001401                           BEQ     .+4             ;BRANCH IF OK
1691  007470  104004                           HLT+4                   ;DATA ON STACK (114100) CHANGED
1692
1693  007472  005767  171114                   TST     ANS4            ;CHECK DATA FROM STACK
1694  007476  001401                           BEQ     .+4             ;BRANCH IF OK
1695  007500  104004                           HLT+4                   ;DATA ON STACK (000000) CHANGED
1696
1697  007502  022767  024252  171104           CMP     #024252,ANS5    ;CHECK DATA FROM STACK
1698  007510  001401                           BEQ     .+4             ;BRANCH IF OK
1699  007512  104006                           HLT+6                   ;DATA ON STACK (024252) CHANGED
1700
1701  007514  022767  125252  171074           CMP     #125252,ANS6    ;CHECK DATA FROM STACK
1702  007522  001401                           BEQ     .+4             ;BRANCH IF OK
```

```
1703  007524  104006                          HLT+6                   ;DATA ON STACK (125252) CHANGED
1704
1705  007526  122767  000043  171244  END43:  CMPB    #43,    ICNT    ;CHECK THE TEST NUMBER
1706  007534  001401                          BEQ     .+4             ;BRANCH IF OK
1707  007536  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP,
1708
1709  007540  104400                          SCOPE
1710
1711
1712                                  ;*************************************************************************
1713                                  ;TEST 44:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
1714                                  ;       125252,125252 / 140200,000000 = 025252,125252
1715                                  ;       PS = 200,       STACK POINTER = SP
1716                                  ;*************************************************************************
1717
1718  007542  004567  006376          TST44:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1719  007546  140200  000000                  ,WORD   140200,000000   ;SECOND OPERAND ON TOP
1720  007552  125252  125252                  ,WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
1721  007556  000217                          ,WORD   217             ;PROCESSOR PRIORITY LEVEL
1722  007560  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1723
1724  007564  000240                          NOP
1725  007566  075036                          FDIV+   SP              ;FLOATING DIVIDE ON THE STACK
1726
1727  007570  004767  006410                  JSR     PC,     POPS    ;POP THE ANSWER
1728  007574  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
1729  007600  001404                          BEQ     TSA44           ;BRANCH IF OK
1730  007602  012706  000500                  MOV     #500,   SP      ;RESTORE STACK POINTER
1731  007606  104000                          HLT                     ;STACK POINTER FOULED UP
1732  007610  000417                          BR      END44           ;SKIP REST OF TEST
1733
1734  007612  022767  000200  170760  TSA44:  CMP     #200,   SPSW    ;CHECK PS (EXCEPT T BIT)
1735  007620  001401                          BEQ     .+4             ;BRANCH IF OK
1736  007622  104000                          HLT                     ;PS NOT EQUAL TO 200
1737
1738  007624  022767  025252  170752          CMP     #025252,ANS1    ;CHECK FIRST HALF OF ANSWER
1739  007632  001401                          BEQ     .+4             ;BRANCH IF OK
1740  007634  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 025252
1741
1742  007636  022767  125252  170742          CMP     #125252,ANS2    ;CHECK SECOND HALF OF ANSWER
1743  007644  001401                          BEQ     .+4             ;BRANCH IF OK
1744  007646  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 125252
1745
1746  007650  122767  000044  171122  END44:  CMPB    #44,    ICNT    ;CHECK THE TEST NUMBER
1747  007656  001401                          BEQ     .+4             ;BRANCH IF OK
1748  007660  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP,
1749
1750  007662  104400                          SCOPE
1751
```

```
1752
1753                                    ;*****************************************************************
1754                                    ;TEST 45:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
1755                                    ;       252525,052525 / 100125,125252 ==> DIVIDE BY ZERO
1756                                    ;       PS(ON STACK) = 053,     STACK POINTER = R5
1757                                    ;*****************************************************************
1758
1759  007664  004567  006432           TST45:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
1760  007670  100125  125252                   ,WORD   100125,125252   ;SECOND OPERAND ON TOP
1761  007674  052525  052525                   ,WORD   052525,052525   ;FIRST OPERAND ON BOTTOM
1762  007700  000047                           ,WORD   047             ;PROCESSOR PRIORITY LEVEL
1763  007702  007732  000113                   ,WORD   ISR45,  113     ;FIS TRAP VECTOR
1764  007706  012705  000630                   MOV     #STACK0,R5      ;SET UP R5 AS STACK POINTER
1765
1766  007712  000240                           NOP
1767  007714  075035                           FDIV+   R5              ;FLOATING DIVIDE ON THE R5 STACK
1768
1769  007716  004767  006432           RTA45:  JSR     X7,     POPR    ;POP THE "ANSWER"
1770  007722  010567  170654                   MOV     R5,     SSP     ;SAVE STACK POINTER (R5)
1771  007726  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1772  007730  000454                           BR      END45
1773
1774  007732  004767  006450           ISR45:  JSR     X7,     POPER   ;POP ALL DATA OFF THE STACKS
1775  007736  010567  170640                   MOV     R5,     SSP     ;SAVE STACK POINTER (R5)
1776  007742  022767  000113  170630           CMP     #113,   SPSW    ;CHECK PS AFTER FIS TRAP
1777  007750  001401                           BEQ     .+4             ;BRANCH IF OK
1778  007752  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 113
1779
1780  007754  022767  000630  170620           CMP     #STACK0,SSP     ;CHECK THE STACK POINTER (R5)
1781  007762  001401                           BEQ     .+4             ;BRANCH IF OK
1782  007764  104000                           HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK0
1783
1784  007766  022767  007716  170610           CMP     #RTA45, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1785  007774  001401                           BEQ     .+4             ;BRANCH IF OK
1786  007776  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1787
1788  010000  022767  000053  170600           CMP     #053,   ANS2    ;CHECK PS BEFORE FIS TRAP
1789  010006  001401                           BEQ     .+4             ;BRANCH IF OK
1790  010010  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 053
1791
1792  010012  022767  100125  170570           CMP     #100125,ANS3    ;CHECK DATA FROM THE STACK
1793  010020  001401                           BEQ     .+4             ;BRANCH IF OK
1794  010022  104004                           HLT+4                   ;DATA ON STACK (100125) CHANGED
1795
1796  010024  022767  125252  170560           CMP     #125252,ANS4    ;CHECK DATA FROM STACK
1797  010032  001401                           BEQ     .+4             ;BRANCH IF OK
1798  010034  104004                           HLT+4                   ;DATA ON STACK (125252) CHANGED
1799
1800  010036  022767  052525  170550           CMP     #052525,ANS5    ;CHECK DATA FROM STACK
1801  010044  001401                           BEQ     .+4             ;BRANCH IF OK
1802  010046  104006                           HLT+6                   ;DATA ON STACK (052525) CHANGED
1803
1804  010050  022767  052525  170540           CMP     #052525,ANS6    ;CHECK DATA FROM STACK
1805  010056  001401                           BEQ     .+4             ;BRANCH IF OK
```

```
1806  010060  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
1807
1808  010062  122767  000045  170710  END45:  CMPB    #45,    ICNT    ;CHECK THE TEST NUMBER
1809  010070  001401                          BEQ     .+4             ;BRANCH IF OK
1810  010072  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1811
1812  010074  104400                          SCOPE
1813
1814
1815                                  ;*************************************************************************
1816                                  ;TEST 46:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
1817                                  ;       167452,125251 / 127652,125252 = 077777,177776
1818                                  ;       PS = 100,       STACK POINTER = R0
1819                                  ;*************************************************************************
1820
1821  010076  004567  006220          TST46:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
1822  010102  127652  125252                  ,WORD   127652,125252   ;SECOND OPERAND ON TOP
1823  010106  167452  125251                  ,WORD   167452,125251   ;FIRST OPERAND ON BOTTOM
1824  010112  000111                          ,WORD   111             ;PROCESSOR PRIORITY LEVEL
1825  010114  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1826  010120  012700  000630                  MOV     #STACK0,R0      ;SET UP STACK POINTER
1827
1828  010124  000240                          NOP
1829  010126  075030                          FDIV*   R0              ;FLOATING DIVIDE ON THE R0 STACK
1830
1831  010130  004767  006220                  JSR     PC,     POPR    ;POP THE ANSWER
1832  010134  010067  170442                  MOV     R0,     $SP     ;SAVE "STACK POINTER"
1833  010140  022767  000100  170432          CMP     #100,   $PSW    ;CHECK PS (EXCEPT T BIT)
1834  010146  001401                          BEQ     .+4             ;BRANCH IF OK
1835  010150  104000                          HLT                     ;PS NOT EQUAL TO 100
1836
1837  010152  022767  000634  170422          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
1838  010160  001401                          BEQ     .+4             ;BRANCH IF OK
1839  010162  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
1840
1841  010164  022767  077777  170412          CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
1842  010172  001401                          BEQ     .+4             ;BRANCH IF OK
1843  010174  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 077777
1844
1845  010176  022767  177776  170402          CMP     #177776,ANS2    ;CHECK SECOND HALF OF ANSWER
1846  010204  001401                          BEQ     .+4             ;BRANCH IF OK
1847  010206  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177776
1848
1849  010210  122767  000046  170562  END46:  CMPB    #46,    ICNT    ;CHECK THE TEST NUMBER
1850  010216  001401                          BEQ     .+4             ;BRANCH IF OK
1851  010220  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1852
1853  010222  104400                          SCOPE
1854
```

```
                                ;*****************************************************************************
                                ;TEST 47:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
                                ;       067452,125252 / 127652,125252 ==> OVERFLOW
                                ;       PS(ON STACK) = 242,     STACK POINTER = R4
                                ;*****************************************************************************

 010224  004567  006072        TST47:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
 010230  127652  125252                ,WORD   127652,125252   ;SECOND OPERAND ON TOP
 010234  067452  125252                ,WORD   067452,125252   ;FIRST OPERAND ON BOTTOM
 010240  000242                       ,WORD   242             ;PROCESSOR PRIORITY LEVEL
 010242  010272  000357                ,WORD   ISR47,  357     ;FIS TRAP VECTOR
 010246  012704  000630                MOV     #STACK0,R4      ;SET UP R4 AS STACK POINTER

 010252  000240                       NOP
 010254  075034                       FDIV+   R4              ;FLOATING DIVIDE ON THE R4 STACK

 010256  004767  006072        RTA47:  JSR     %7,     POPR    ;POP THE "ANSWER"
 010262  010467  170314                MOV     R4,     $SP     ;SAVE STACK POINTER (R4)
 010266  104002                       HLT+2                   ;FIS TRAP DIDN'T OCCURE!
 010270  000454                       BR      END47

 010272  004767  006110        ISR47:  JSR     %7,     POPER   ;POP ALL DATA OFF THE STACKS
 010276  010467  170300                MOV     R4,     $SP     ;SAVE STACK POINTER (R4)
 010302  022767  000357  170270        CMP     #357,   $PSW    ;CHECK PS AFTER FIS TRAP
 010310  001401                       BEQ     .+4             ;BRANCH IF OK
 010312  104000                       HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 357

 010314  022767  000630  170260        CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R4)
 010322  001401                       BEQ     .+4             ;BRANCH IF OK
 010324  104000                       HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK0

 010326  022767  010256  170250        CMP     #RTA47, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 010334  001401                       BEQ     .+4             ;BRANCH IF OK
 010336  104001                       HLT+1                   ;FIS TRAP AT WRONG ADDRESS

 010340  022767  000242  170240        CMP     #242,   ANS2    ;CHECK PS BEFORE FIS TRAP
 010346  001401                       BEQ     .+4             ;BRANCH IF OK
 010350  104002                       HLT+2                   ;PS AT FIS TRAP TIME NOT 242

 010352  022767  127652  170230        CMP     #127652,ANS3    ;CHECK DATA FROM THE STACK
 010360  001401                       BEQ     .+4             ;BRANCH IF OK
 010362  104004                       HLT+4                   ;DATA ON STACK (127652) CHANGED

 010364  022767  125252  170220        CMP     #125252,ANS4    ;CHECK DATA FROM STACK
 010372  001401                       BEQ     .+4             ;BRANCH IF OK
 010374  104004                       HLT+4                   ;DATA ON STACK (125252) CHANGED

 010376  022767  067452  170210        CMP     #067452,ANS5    ;CHECK DATA FROM STACK
 010404  001401                       BEQ     .+4             ;BRANCH IF OK
 010406  104006                       HLT+6                   ;DATA ON STACK (067452) CHANGED

 010410  022767  125252  170200        CMP     #125252,ANS6    ;CHECK DATA FROM STACK
 010416  001401                       BEQ     .+4             ;BRANCH IF OK
```

```
1909  010420  104026                          HLT+6                   ;DATA ON STACK (125252) CHANGED
1910
1911  010422  122767  000047  170350  END47:  CMPB    #47,    ICNT    ;CHECK THE TEST NUMBER
1912  010430  001401                          BEQ     .+4             ;BRANCH IF OK
1913  010432  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1914
1915  010434  104400                          SCOPE
1916
1917
1918                                  ;*****************************************************************
1919                                  ;TEST 50:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
1920                                  ;       167452,125252 / 027652,125253 = 177777,177777
1921                                  ;       PS = 310,       STACK POINTER = SP
1922                                  ;*****************************************************************
1923
1924  010436  004567  000502          TST50:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1925  010442  027652  125253                  ,WORD   027652,125253   ;SECOND OPERAND ON TOP
1926  010446  167452  125252                  ,WORD   167452,125252   ;FIRST OPERAND ON BOTTOM
1927  010452  000300                          ,WORD   300             ;PROCESSOR PRIORITY LEVEL
1928  010454  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
1929
1930  010460  000240                          NOP
1931  010462  075036                          FDIV*   SP              ;FLOATING DIVIDE ON THE STACK
1932
1933  010464  004767  000514                  JSR     PC,     POPS    ;POP THE ANSWER
1934  010470  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
1935  010474  001404                          BEQ     TSA50           ;BRANCH IF OK
1936  010476  012706  000500                  MOV     #500,   SP      ;RESTORE STACK POINTER
1937  010502  104000                          HLT                     ;STACK POINTER FOULED UP
1938  010504  000417                          BR      END50           ;SKIP REST OF TEST
1939
1940  010506  022767  000310  170064  TSA50:  CMP     #310,   SPSW    ;CHECK PS (EXCEPT T BIT)
1941  010514  001401                          BEQ     .+4             ;BRANCH IF OK
1942  010516  104000                          HLT                     ;PS NOT EQUAL TO 310
1943
1944  010520  022767  177777  170056          CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
1945  010526  001401                          BEQ     .+4             ;BRANCH IF OK
1946  010530  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777
1947
1948  010532  022767  177777  170046          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
1949  010540  001401                          BEQ     .+4             ;BRANCH IF OK
1950  010542  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777
1951
1952  010544  122767  000050  170226  END50:  CMPB    #50,    ICNT    ;CHECK THE TEST NUMBER
1953  010552  001401                          BEQ     .+4             ;BRANCH IF OK
1954  010554  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1955
1956  010556  104400                          SCOPE
1957
```

```
1958
1959                                    ;*****************************************************************
1960                                    ;TEST 51:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
1961                                    ;       154200,000000 / 014377,177777 ==> OVERFLOW
1962                                    ;       PS(ON STACK) = 242,     STACK POINTER = SP
1963                                    ;*****************************************************************
1964
1965  010560  004567  005360           TST51:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1966  010564  014377  177777                   ,WORD   014377,177777   ;SECOND OPERAND ON TOP
1967  010570  154200  000000                   ,WORD   154200,000000   ;FIRST OPERAND ON BOTTOM
1968  010574  000246                           ,WORD   246             ;PROCESSOR PRIORITY LEVEL
1969  010576  010622  000103                   ,WORD   ISR51,  103     ;FIS TRAP VECTOR
1970
1971  010602  000240                           NOP
1972  010604  075036                           FDIV+   SP              ;FLOATING DIVIDE ON THE STACK
1973
1974  010606  004767  005372           RTA51:  JSR     %7,     POPS    ;POP THE "ANSWER"
1975  010612  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1976  010614  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK POINTER
1977  010620  000453                           BR      END51
1978
1979  010622  004767  005412           ISR51:  JSR     %7,     POPES   ;POP ALL DATA OFF THE STACK
1980  010626  022706  000500                   CMP     #500,   SP      ;CHECK THE STACK POINTER
1981  010632  001404                           BEQ     ISA51           ;BRANCH IF OK
1982  010634  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK POINTER
1983  010640  104000                           HLT                     ;STACK POINTER FOULED UP
1984  010642  000442                           BR      END51           ;SKIP REST OF TEST
1985
1986  010644  022767  000103  167726   ISA51:  CMP     #103,   $PSW    ;CHECK PS AFTER FIS TRAP
1987  010652  001401                           BEQ     .+4             ;BRANCH IF OK
1988  010654  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 103
1989
1990  010656  022767  010606  167720           CMP     #RTA51, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1991  010664  001401                           BEQ     .+4             ;BRANCH IF OK
1992  010666  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1993
1994  010670  022767  000242  167710           CMP     #242,   ANS2    ;CHECK PS BEFORE FIS TRAP
1995  010676  001401                           BEQ     .+4             ;BRANCH IF OK
1996  010700  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 242
1997
1998  010702  022767  014377  167700           CMP     #014377,ANS3    ;CHECK DATA FROM THE STACK
1999  010710  001401                           BEQ     .+4             ;BRANCH IF OK
2000  010712  104004                           HLT+4                   ;DATA ON STACK (014377) CHANGED
2001
2002  010714  022767  177777  167670           CMP     #177777,ANS4    ;CHECK DATA FROM STACK
2003  010722  001401                           BEQ     .+4             ;BRANCH IF OK
2004  010724  104004                           HLT+4                   ;DATA ON STACK (177777) CHANGED
2005
2006  010726  022767  154200  167660           CMP     #154200,ANS5    ;CHECK DATA FROM STACK
2007  010734  001401                           BEQ     .+4             ;BRANCH IF OK
2008  010736  104006                           HLT+6                   ;DATA ON STACK (154200) CHANGED
2009
2010  010740  005767  167652                   TST     ANS6            ;CHECK DATA FROM STACK
2011  010744  001401                           BEQ     .+4             ;BRANCH IF OK
```

```
2012  010746  104006                          HLT+6                   ;DATA ON STACK (000000) CHANGED
2013
2014  010750  122767  000051  170022  END51:  CMPB    #51,    ICNT    ;CHECK THE TEST NUMBER
2015  010756  001401                          BEQ     .+4             ;BRANCH IF OK
2016  010760  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2017
2018  010762  104400                          SCOPE
2019
2020
2021                                  ;*************************************************************************
2022                                  ;TEST 52:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
2023                                  ;       125252,125252 / 065252,125252 = 100200,000000
2024                                  ;       PS = 210,       STACK POINTER = R2
2025                                  ;*************************************************************************
2026
2027  010764  004567  005332          TST52:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
2028  010770  065252  125252                  ,WORD   065252,125252   ;SECOND OPERAND ON TOP
2029  010774  125252  125252                  ,WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
2030  011000  000217                          ,WORD   217             ;PROCESSOR PRIORITY LEVEL
2031  011002  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
2032  011006  012702  000630                  MOV     #STACK0,R2      ;SET UP STACK POINTER
2033
2034  011012  000240                          NOP
2035  011014  075032                          FDIV*   R2              ;FLOATING DIVIDE ON THE R2 STACK
2036
2037  011016  004767  005332                  JSR     PC,     POPR    ;POP THE ANSWER
2038  011022  010267  167554                  MOV     R2,     SSP     ;SAVE "STACK POINTER"
2039  011026  022767  000210  167544          CMP     #210,   SPSW    ;CHECK PS (EXCEPT T BIT)
2040  011034  001401                          BEQ     .+4             ;BRANCH IF OK
2041  011036  104000                          HLT                     ;PS NOT EQUAL TO 210
2042
2043  011040  022767  000634  167534          CMP     #STACK4,SSP     ;CHECK THE STACK POINTER (R2)
2044  011046  001401                          BEQ     .+4             ;BRANCH IF OK
2045  011050  104000                          HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
2046
2047  011052  022767  100200  167524          CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
2048  011060  001401                          BEQ     .+4             ;BRANCH IF OK
2049  011062  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 100200
2050
2051  011064  005767  167516                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
2052  011070  001401                          BEQ     .+4             ;BRANCH IF OK
2053  011072  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
2054
2055  011074  122767  000052  167676  END52:  CMPB    #52,    ICNT    ;CHECK THE TEST NUMBER
2056  011102  001401                          BEQ     .+4             ;BRANCH IF OK
2057  011104  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2058
2059  011106  104400                          SCOPE
2060
```

```
2061
2062                                    ;*****************************************************************************
2063                                    ;TEST 53:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
2064                                    ;       025252,125251 / 065252,125252 ==> UNDERFLOW
2065                                    ;       PS(ON STACK) = 012,     STACK POINTER = R1
2066                                    ;*****************************************************************************
2067
2068  011110  004567  005206           TST53:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
2069  011114  065252  125252                   ,WORD   065252,125252   ;SECOND OPERAND ON TOP
2070  011120  025252  125251                   ,WORD   025252,125251   ;FIRST OPERAND ON BOTTOM
2071  011124  000015                           ,WORD   015             ;PROCESSOR PRIORITY LEVEL
2072  011126  011156  000300                   ,WORD   ISR53,  300     ;FIS TRAP VECTOR
2073  011132  012701  000630                   MOV     #STACK0,R1      ;SET UP R1 AS STACK POINTER
2074
2075  011136  000240                           NOP
2076  011140  075031                           FDIV+   R1              ;FLOATING DIVIDE ON THE R1 STACK
2077
2078  011142  004767  005206           RTA53:  JSR     %7,     POPR    ;POP THE "ANSWER"
2079  011146  010167  167430                   MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
2080  011152  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
2081  011154  000454                           BR      END53
2082
2083  011156  004767  005224           ISR53:  JSR     %7,     POPER   ;POP ALL DATA OFF THE STACKS
2084  011162  010167  167414                   MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
2085  011166  022767  000300  167404           CMP     #300,   $PSW    ;CHECK PS AFTER FIS TRAP
2086  011174  001401                           BEQ     .+4             ;BRANCH IF OK
2087  011176  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 300
2088
2089  011200  022767  000630  167374           CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
2090  011206  001401                           BEQ     .+4             ;BRANCH IF OK
2091  011210  104000                           HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
2092
2093  011212  022767  011142  167364           CMP     #RTA53, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2094  011220  001401                           BEQ     .+4             ;BRANCH IF OK
2095  011222  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2096
2097  011224  022767  000012  167354           CMP     #012,   ANS2    ;CHECK PS BEFORE FIS TRAP
2098  011232  001401                           BEQ     .+4             ;BRANCH IF OK
2099  011234  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 012
2100
2101  011236  022767  065252  167344           CMP     #065252,ANS3    ;CHECK DATA FROM THE STACK
2102  011244  001401                           BEQ     .+4             ;BRANCH IF OK
2103  011246  104004                           HLT+4                   ;DATA ON STACK (065252) CHANGED
2104
2105  011250  022767  125252  167334           CMP     #125252,ANS4    ;CHECK DATA FROM STACK
2106  011256  001401                           BEQ     .+4             ;BRANCH IF OK
2107  011260  104004                           HLT+4                   ;DATA ON STACK (125252) CHANGED
2108
2109  011262  022767  025252  167324           CMP     #025252,ANS5    ;CHECK DATA FROM STACK
2110  011270  001401                           BEQ     .+4             ;BRANCH IF OK
2111  011272  104006                           HLT+6                   ;DATA ON STACK (025252) CHANGED
2112
2113  011274  022767  125251  167314           CMP     #125251,ANS6    ;CHECK DATA FROM STACK
2114  011302  001401                           BEQ     .+4             ;BRANCH IF OK
```

```
2115  011304  104006                          HLT+6                   ;DATA ON STACK (125251) CHANGED
2116
2117  011306  122767  000053  167464  END53:  CMPB    #53,    ICNT    ;CHECK THE TEST NUMBER
2118  011314  001401                          BEQ     .+4             ;BRANCH IF OK
2119  011316  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2120
2121  011320  104400                          SCOPE
2122
2123
2124                                  ;*************************************************************************
2125                                  ;TEST 54:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
2126                                  ;       000000,000000 / 140670,123456 = 000000,000000
2127                                  ;       PS = 104,       STACK POINTER = R3
2128                                  ;*************************************************************************
2129
2130  011322  004567  004774          TST54:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
2131  011326  140670  123456                  ,WORD   140670,123456   ;SECOND OPERAND ON TOP
2132  011332  000000  000000                  ,WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
2133  011336  000105                          ,WORD   105             ;PROCESSOR PRIORITY LEVEL
2134  011340  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
2135  011344  012703  000630                  MOV     #STACK0,R3      ;SET UP STACK POINTER
2136
2137  011350  000240                          NOP
2138  011352  075033                          FDIV+   R3              ;FLOATING DIVIDE ON THE R3 STACK
2139
2140  011354  004767  004774                  JSR     PC,     POPR    ;POP THE ANSWER
2141  011360  010367  167216                  MOV     R3,     $SP     ;SAVE "STACK POINTER"
2142  011364  022767  000104  167206          CMP     #104,   SPSW    ;CHECK PS (EXCEPT T BIT)
2143  011372  001401                          BEQ     .+4             ;BRANCH IF OK
2144  011374  104000                          HLT                     ;PS NOT EQUAL TO 104
2145
2146  011376  022767  000634  167176          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
2147  011404  001401                          BEQ     .+4             ;BRANCH IF OK
2148  011406  104000                          HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
2149
2150  011410  005767  167170                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
2151  011414  001401                          BEQ     .+4             ;BRANCH IF OK
2152  011416  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
2153
2154  011420  005767  167162                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
2155  011424  001401                          BEQ     .+4             ;BRANCH IF OK
2156  011426  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
2157
2158  011430  122767  000054  167342  END54:  CMPB    #54,    ICNT    ;CHECK THE TEST NUMBER
2159  011436  001401                          BEQ     .+4             ;BRANCH IF OK
2160  011440  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2161
2162  011442  104400                          SCOPE
2163
```

```
2164
2165                                    ;*****************************************************************************
2166                                    ;TEST 55:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
2167                                    ;       100052,052525 / 000006,123456 ==> DIVIDE BY ZERO
2168                                    ;       PS(ON STACK) = 353,     STACK POINTER = SP
2169                                    ;*****************************************************************************
2170
2171  011444  004567  004474           TST55:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2172  011450  000006  123456                   ,WORD   000006,123456   ;SECOND OPERAND ON TOP
2173  011454  100052  052525                   ,WORD   100052,052525   ;FIRST OPERAND ON BOTTOM
2174  011460  000357                           ,WORD   357             ;PROCESSOR PRIORITY LEVEL
2175  011462  011506  000311                   ,WORD   ISR55,  311     ;FIS TRAP VECTOR
2176
2177  011466  000240                           NOP
2178  011470  075036                           FDIV+   SP              ;FLOATING DIVIDE ON THE STACK
2179
2180  011472  004767  004506           RTA55:  JSR     %7,     POPS    ;POP THE "ANSWER"
2181  011476  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
2182  011500  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK POINTER
2183  011504  000454                           BR      END55
2184
2185  011506  004767  004526           ISR55:  JSR     %7,     POPES   ;POP ALL DATA OFF THE STACK
2186  011512  022706  000500                   CMP     #500,   SP      ;CHECK THE STACK POINTER
2187  011516  001404                           BEQ     ISA55           ;BRANCH IF OK
2188  011520  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK POINTER
2189  011524  104000                           HLT                     ;STACK POINTER FOULED UP
2190  011526  000443                           BR      END55           ;SKIP REST OF TEST
2191
2192  011530  022767  000311  167042 ISA55:  CMP     #311,   $PSW    ;CHECK PS AFTER FIS TRAP
2193  011536  001401                           BEQ     .+4             ;BRANCH IF OK
2194  011540  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 311
2195
2196  011542  022767  011472  167034           CMP     #RTA55, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2197  011550  001401                           BEQ     .+4             ;BRANCH IF OK
2198  011552  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2199
2200  011554  022767  000353  167024           CMP     #353,   ANS2    ;CHECK PS BEFORE FIS TRAP
2201  011562  001401                           BEQ     .+4             ;BRANCH IF OK
2202  011564  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 353
2203
2204  011566  022767  000006  167014           CMP     #000006,ANS3    ;CHECK DATA FROM THE STACK
2205  011574  001401                           BEQ     .+4             ;BRANCH IF OK
2206  011576  104004                           HLT+4                   ;DATA ON STACK (000006) CHANGED
2207
2208  011600  022767  123456  167004           CMP     #123456,ANS4    ;CHECK DATA FROM STACK
2209  011606  001401                           BEQ     .+4             ;BRANCH IF OK
2210  011610  104004                           HLT+4                   ;DATA ON STACK (123456) CHANGED
2211
2212  011612  022767  100052  166774           CMP     #100052,ANS5    ;CHECK DATA FROM STACK
2213  011620  001401                           BEQ     .+4             ;BRANCH IF OK
2214  011622  104006                           HLT+6                   ;DATA ON STACK (100052) CHANGED
2215
2216  011624  022767  052525  166764           CMP     #052525,ANS6    ;CHECK DATA FROM STACK
2217  011632  001401                           BEQ     .+4             ;BRANCH IF OK
```

```
2218  011634  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
2219
2220  011636  122767  000055  167134  END55:  CMPB    #55,    ICNT    ;CHECK THE TEST NUMBER
2221  011644  001401                          BEQ     .+4             ;BRANCH IF OK
2222  011646  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2223
2224  011650  104400                          SCOPE
2225
2226
2227                                  ;*************************************************************************
2228                                  ;TEST 56:       FADD (KE11F FLOATING ADD INSTRUCTION)
2229                                  ;       004000,105004 + 104000,104000 = 000401,000000
2230                                  ;       PS = 140,       STACK POINTER = PC
2231                                  ;*************************************************************************
2232
2233  011652  004567  004622          TST56:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2234  011656  011702                          .WORD   STK56           ;TOP OF STACK
2235  011660  104000  104000                  .WORD   104000,104000   ;SECOND OPERAND ON TOP
2236  011664  004000  105004                  .WORD   004000,105004   ;FIRST OPERAND ON BOTTOM
2237  011670  000144                          .WORD   144             ;PROCESSOR PRIORITY LEVEL
2238  011672  016606  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
2239
2240  011676  000240                          NOP
2241  011700  075007                          FADD+   PC              ;FLOATING ADD ON FOLLOWING 4 WORDS
2242  011702  104000          STK56:  104000                          ;SHOULD CONTAIN 104000
2243  011704  104000                  104000                          ;SHOULD CONTAIN 104000
2244  011706  004000                  004000                          ;BEFORE FADD, 004000; AFTER, 000401
2245  011710  105004                  105004                          ;BEFORE FADD, 105004; AFTER, 000000
2246
2247  011712  004767  004620                  JSR     PC,     POP7    ;POP THE ANSWER
2248  011716  022767  000140  166654          CMP     #140,   SPSW    ;CHECK PS (EXCEPT T BIT)
2249  011724  001401                          BEQ     .+4             ;BRANCH IF OK
2250  011726  104000                          HLT                     ;PS NOT EQUAL TO 140
2251
2252  011730  022767  104000  166646          CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK56)
2253  011736  001401                          BEQ     .+4             ;BRANCH IF OK
2254  011740  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 104000
2255
2256  011742  022767  104000  166636          CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK56+2)
2257  011750  001401                          BEQ     .+4             ;BRANCH IF OK
2258  011752  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 104000
2259
2260  011754  022767  000401  166626          CMP     #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
2261  011762  001401                          BEQ     .+4             ;BRANCH IF OK
2262  011764  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000401
2263
2264  011766  005767  166620                  TST     ANS4            ;CHECK SECOND HALF OF ANSWER
2265  011772  001401                          BEQ     .+4             ;BRANCH IF OK
2266  011774  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000
2267
2268  011776  122767  000056  166774  END56:  CMPB    #56,    ICNT    ;CHECK THE TEST NUMBER
2269  012004  001401                          BEQ     .+4             ;BRANCH IF OK
2270  012006  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2271
```

```
2272  012010  104400                        SCOPE
2273
2274
2275                                  ;*****************************************************************
2276                                  ;TEST 57:       FSUB (KE11F FLOATING SUBTRACT INSTRUCTION)
2277                                  ;       104000,105004 - 104000,104000 = 100401,000000
2278                                  ;       PS = 250,       STACK POINTER = PC
2279                                  ;*****************************************************************
2280
2281  012012  004567  000462         TST57:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2282  012016  012042                         ,WORD   STK57           ;TOP OF STACK
2283  012020  104000  104000                 ,WORD   104000,104000   ;SECOND OPERAND ON TOP
2284  012024  104000  105004                 ,WORD   104000,105004   ;FIRST OPERAND ON BOTTOM
2285  012030  000252                         ,WORD   252             ;PROCESSOR PRIORITY LEVEL
2286  012032  016666  000340                 ,WORD   TRAPER,340      ;FIS TRAP VECTOR
2287
2288  012036  000240                         NOP
2289  012040  075017                         FSUB+   PC              ;FLOATING SUBTRACT ON FOLLOWING 4 WORDS
2290  012042  104000         STK57:  104000                          ;SHOULD CONTAIN 104000
2291  012044  104000                 104000                          ;SHOULD CONTAIN 104000
2292  012046  104000                 104000                          ;BEFORE FSUB, 104000; AFTER, 100401
2293  012050  105004                 105004                          ;BEFORE FSUB, 105004; AFTER, 000000
2294
2295  012052  004767  000460                 JSR     PC,     POP7    ;POP THE ANSWER
2296  012056  022767  000250  166514         CMP     #250,   $PSW    ;CHECK PS (EXCEPT T BIT)
2297  012064  001401                         BEQ     .+4             ;BRANCH IF OK
2298  012066  104000                         HLT                     ;PS NOT EQUAL TO 250
2299
2300  012070  022767  104000  166506         CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK57)
2301  012076  001401                         BEQ     .+4             ;BRANCH IF OK
2302  012100  104002                         HLT+2                   ;ANS1 NOT EQUAL TO 104000
2303
2304  012102  022767  104000  166476         CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK57+2)
2305  012110  001401                         BEQ     .+4             ;BRANCH IF OK
2306  012112  104002                         HLT+2                   ;ANS2 NOT EQUAL TO 104000
2307
2308  012114  022767  100401  166466         CMP     #100401,ANS3    ;CHECK FIRST HALF OF ANSWER
2309  012122  001401                         BEQ     .+4             ;BRANCH IF OK
2310  012124  104004                         HLT+4                   ;ANS3 NOT EQUAL TO 100401
2311
2312  012126  005767  166460                 TST     ANS4            ;CHECK SECOND HALF OF ANSWER
2313  012132  001401                         BEQ     .+4             ;BRANCH IF OK
2314  012134  104004                         HLT+4                   ;ANS4 NOT EQUAL TO 000000
2315
2316  012136  122767  000057  166634 END57:  CMPB    #57,    ICNT    ;CHECK THE TEST NUMBER
2317  012144  001401                         BEQ     .+4             ;BRANCH IF OK
2318  012146  104000                         HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2319
2320  012150  104400                        SCOPE
2321
```

```
2322
2323                                    ;*****************************************************************
2324                                    ;TEST 60:       FMUL (KE11F FLOATING MULTIPLY INSTRUCTION)
2325                                    ;       134600,073601 * 104000,104000 = 000401,000000
2326                                    ;       PS = 240,       STACK POINTER = PC
2327                                    ;*****************************************************************
2328
2329  012152  004567  004322           TST60:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2330  012156  012202                           ,WORD   STK60           ;TOP OF STACK
2331  012160  104000  104000                   ,WORD   104000,104000   ;SECOND OPERAND ON TOP
2332  012164  134600  073601                   ,WORD   134600,073601   ;FIRST OPERAND ON BOTTOM
2333  012170  000246                           ,WORD   246             ;PROCESSOR PRIORITY LEVEL
2334  012172  016606  000340                   ,WORD   TRAPER,340      ;FIS TRAP VECTOR
2335
2336  012176  000240                           NOP
2337  012200  075027                           FMUL+   PC              ;FLOATING MULTIPLY ON FOLLOWING 4 WORDS
2338  012202  104000           -       STK60:  104000                  ;SHOULD CONTAIN 104000
2339  012204  104000                           104000                  ;SHOULD CONTAIN 104000
2340  012206  134600                           134600                  ;BEFORE FMUL, 134600; AFTER, 000401
2341  012210  073601                           073601                  ;BEFORE FMUL, 073601; AFTER, 000000
2342
2343  012212  004767  004320                   JSR     PC,     POP7    ;POP THE ANSWER
2344  012216  022767  000240  166354           CMP     #240,   $PSW    ;CHECK PS (EXCEPT T BIT)
2345  012224  001401                           BEQ     .+4             ;BRANCH IF OK
2346  012226  104000                           HLT                     ;PS NOT EQUAL TO 240
2347
2348  012230  022767  104000  166346           CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK60)
2349  012236  001401                           BEQ     .+4             ;BRANCH IF OK
2350  012240  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 104000
2351
2352  012242  022767  104000  166336           CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK60+2)
2353  012250  001401                           BEQ     .+4             ;BRANCH IF OK
2354  012252  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 104000
2355
2356  012254  022767  000401  166326           CMP     #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
2357  012262  001401                           BEQ     .+4             ;BRANCH IF OK
2358  012264  104004                           HLT+4                   ;ANS3 NOT EQUAL TO 000401
2359
2360  012266  005767  166320                   TST     ANS4            ;CHECK SECOND HALF OF ANSWER
2361  012272  001401                           BEQ     .+4             ;BRANCH IF OK
2362  012274  104004                           HLT+4                   ;ANS4 NOT EQUAL TO 000000
2363
2364  012276  122767  000060  166474  END60:  CMPB    #60,    ICNT    ;CHECK THE TEST NUMBER
2365  012304  001401                           BEQ     .+4             ;BRANCH IF OK
2366  012306  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2367
2368  012310  104400                           SCOPE
2369
```

```
2370
2371                                    ;*****************************************************************
2372                                    ;TEST 61:       FDIV (KE11F FLOATING DIVIDE INSTRUCTION)
2373                                    ;       102500,146000 / 104000,104000 = 036700,000000
2374                                    ;       PS = 340,       STACK POINTER = PC
2375                                    ;*****************************************************************
2376
2377  012312  004567  004162           TST61:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2378  012316  012342                           ,WORD   STK61           ;TOP OF STACK
2379  012320  104000  104000                   ,WORD   104000,104000   ;SECOND OPERAND ON TOP
2380  012324  102500  146000                   ,WORD   102500,146000   ;FIRST OPERAND ON BOTTOM
2381  012330  000357                           ,WORD   357             ;PROCESSOR PRIORITY LEVEL
2382  012332  016606  000340                   ,WORD   TRAPER,340      ;FIS TRAP VECTOR
2383
2384  012336  000240                           NOP
2385  012340  075037                           FDIV+   PC              ;FLOATING DIVIDE ON FOLLOWING 4 WORDS
2386  012342  104000           STK61:  104000                          ;SHOULD CONTAIN 104000
2387  012344  104000                   104000                          ;SHOULD CONTAIN 104000
2388  012346  102500                   102500                          ;BEFORE FDIV, 102500; AFTER, 036700
2389  012350  146000                   146000                          ;BFFORE FDIV, 146000; AFTER, 000000
2390
2391  012352  004767  004160           JSR     PC,     POP7    ;POP THE ANSWER
2392  012356  022767  000340  166214   CMP     #340,   $PSW    ;CHECK PS (EXCEPT T BIT)
2393  012364  001401                   BEQ     .+4             ;BRANCH IF OK
2394  012366  104000                   HLT                     ;PS NOT EQUAL TO 340
2395
2396  012370  022767  104000  166206   CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK61)
2397  012376  001401                   BEQ     .+4             ;BRANCH IF OK
2398  012400  104002                   HLT+2                   ;ANS1 NOT EQUAL TO 104000
2399
2400  012402  022767  104000  166176   CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK61+2)
2401  012410  001401                   BEQ     .+4             ;BRANCH IF OK
2402  012412  104002                   HLT+2                   ;ANS2 NOT EQUAL TO 104000
2403
2404  012414  022767  036700  166166   CMP     #036700,ANS3    ;CHECK FIRST HALF OF ANSWER
2405  012422  001401                   BEQ     .+4             ;BRANCH IF OK
2406  012424  104004                   HLT+4                   ;ANS3 NOT EQUAL TO 036700
2407
2408  012426  005767  166160           TST     ANS4            ;CHECK SECOND HALF OF ANSWER
2409  012432  001401                   BEQ     .+4             ;BRANCH IF OK
2410  012434  104004                   HLT+4                   ;ANS4 NOT EQUAL TO 000000
2411
2412  012436  122767  000061  166334   END61:  CMPB    #61,    ICNT    ;CHECK THE TEST NUMBER
2413  012444  001401                   BEQ     .+4             ;BRANCH IF OK
2414  012446  104000                   HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2415
2416  012450  104400                   SCOPE
2417
```

```
2418
2419                                    ;*****************************************************************
2420                                    ;TEST 62:       TEST ALL INSTRUCTION TOGETHER
2421                                    ;                       032107,065432 * 045670,123456
2422                                    ;       134343,107070 + -----------------------------  = 137201,115230
2423                                    ;                       (135252,125252 - 040616,016161)
2424                                    ;       PS=150, STACK POINTER=R4
2425                                    ;*****************************************************************
2426
2427  012452  012704  000640            TST62:  MOV     #STACK8,R4      ;SET STACK POINTER
2428  012456  012744  107070                    MOV     #107070,-(R4)   ;LOAD DATA ONTO STACK
2429  012462  012744  134343                    MOV     #134343,-(R4)
2430  012466  012744  065432                    MOV     #065432,-(R4)
2431  012472  012744  032107                    MOV     #032107,-(R4)
2432  012476  012744  123456                    MOV     #123456,-(R4)
2433  012502  012744  045670                    MOV     #045670,-(R4)
2434  012506  012744  125252                    MOV     #125252,-(R4)
2435  012512  012744  135252                    MOV     #135252,-(R4)
2436  012516  012744  016161                    MOV     #016161,-(R4)
2437  012522  012744  040616                    MOV     #040616,-(R4)
2438  012526  012737  000144  177776            MOV     #144,   @#PS    ;SET PROCESSOR STATUS
2439
2440  012534  000240                            NOP
2441  012536  075014                            FSUB    R4      ;135252,125252-040616,016161=140616,017434
2442  012540  075034                            FDIV    R4      ;045670,123456/140616,017434=145246,047065
2443  012542  075024                            FMUL    R4      ;032107,065432*145246,047065=137201,106137
2444  012544  075004                            FADD    R4      ;134343,107070+137201,106137=137201,115230
2445
2446  012546  013767  177776  166024            MOV     @#PS,   $PSW    ;SAVE FINAL PS
2447  012554  042767  000020  166016            BIC     #20,    $PSW    ;CLR T-BIT
2448  012562  012467  166016                    MOV     (R4)+,  ANS1    ;SAVE FIRST HALF OF ANSWER
2449  012566  012467  166014                    MOV     (R4)+,  ANS2    ;SAVE SECOND HALF OF ANSWER
2450  012572  010467  166004                    MOV     R4,     $SP     ;SAVE STACK POINTER
2451  012576  022767  000150  165774            CMP     #150,   $PSW    ;CHECK PS (EXCEPT T BIT)
2452  012604  001401                            BEQ     .+4             ;BRANCH IF OK
2453  012606  104000                            HLT                     ;PS NOT EQUAL TO 150
2454
2455  012610  022767  000640  165764            CMP     #STACK8,$SP     ;CHECK THE STACK POINTER (R4)
2456  012616  001401                            BEQ     .+4             ;BRANCH IF OK
2457  012620  104000                            HLT                     ;STACK POINTER (R4) NOT EQUAL TO 674
2458
2459  012622  022767  137201  165754            CMP     #137201,ANS1    ;CHECK FIRST HALF OF ANSWER
2460  012630  001401                            BEQ     .+4             ;BRANCH IF OK
2461  012632  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 137201
2462
2463  012634  022767  115230  165744            CMP     #115230,ANS2    ;CHECK SECOND HALF OF ANSWER
2464  012642  001401                            BEQ     .+4             ;BRANCH IF OK
2465  012644  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 115230
2466
2467  012646  122767  000062  166124  END62:   CMPB    #62,    ICNT    ;CHECK THE TEST NUMBER
2468  012654  001401                            BEQ     .+4             ;BRANCH IF OK
2469  012656  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2470
2471  012660  104400                            SCOPE
```

```
2472
2473                                    ;*****************************************************************
2474                                    ;TEST 63:       TEST THAT YELLOW ZONE STACK OVERFLOW WORKS WITH FIS
2475                                    ;       STACK POINTER = SP = 356
2476                                    ;*****************************************************************
2477
2478  012662  012701  000356           TST63:  MOV     #356,   R1      ;SET UP STACK
2479  012666  012721  035152                   MOV     #035152,(R1)+   ;PUT DATA ON THE STACK
2480  012672  012721  125252                   MOV     #125252,(R1)+
2481  012676  012721  043125                   MOV     #043125,(R1)+
2482  012702  012721  052525                   MOV     #052525,(R1)+
2483  012706  012737  012772  000004           MOV     #3$,    @#4     ;SETUP STACK ERROR VECTOR
2484  012714  012737  000300  000006           MOV     #300,   @#6
2485  012722  013700  177776                   MOV     @#PS,   R0      ;SAVE THE T-BIT
2486  012726  012746  000340                   MOV     #340,   -(SP)   ;CLR T-BIT
2487  012732  012746  012740                   MOV     #1$,    -(SP)
2488  012736  000002                           RTI
2489  012740  012706  000356           1$:     MOV     #356,   SP      ;OVERFLOW THE STACK
2490
2491  012744  000240                           NOP
2492  012746  075006                           FADD+   SP              ;DO 043125 FLOATING POINT ADD
2493
2494  012750  016767  165022  165622  2$:     MOV     PS,     $PSW    ;SAVE PS FOR TYPING
2495  012756  010667  165620                   MOV     SP,     $SP     ;SAVE STACK POINTER
2496  012762  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK
2497  012766  104000                           HLT                     ;STACK OVERFLOW DION'T TRAP
2498
2499  012770  000454                           BR      4$
2500
2501  012772  016767  165000  165600  3$:     MOV     PS,     $PSW    ;SAVE THE PS
2502  013000  010667  165576                   MOV     SP,     $SP     ;SAVE THE STACK POINTER
2503  013004  010601                           MOV     SP,     R1
2504  013006  012706  000500                   MOV     #500,   SP      ;RESTORE THE STACK
2505  013012  012702  000604                   MOV     #ANS1,  R2      ;TOP OF ANSWER TABLE
2506  013016  012122                           MOV     (R1)+,  (R2)+   ;SAVE THE STACK DATA
2507  013020  012122                           MOV     (R1)+,  (R2)+
2508  013022  012122                           MOV     (R1)+,  (R2)+
2509  013024  012122                           MOV     (R1)+,  (R2)+
2510  013026  022767  000300  165544           CMP     #300,   $PSW    ;CHECK THE PS AFTER THE TRAP
2511  013034  001401                           BEQ     .+4             ;BRANCH IF OK
2512  013036  104000                           HLT                     ;PS NOT EQUAL TO 300
2513
2514  013040  022767  000356  165534           CMP     #356,   $SP     ;CHECK FOR SP AT RIGHT SPOT
2515  013046  001401                           BEQ     .+4             ;BRANCH IF OK
2516  013050  104000                           HLT                     ;STACK POINTER FOULED UP
2517
2518  013052  022767  012750  165524           CMP     #2$,    ANS1    ;CHECK TOP OF STACK FOR RTI ADR.
2519  013060  001401                           BEQ     .+4             ;BRANCH IF OK
2520  013062  104001                           HLT+1                   ;RTI ADDRESS NOT EQUAL TO #2$
2521
2522  013064  022767  000340  165514           CMP     #340,   ANS2    ;CHECK STACK DATA FOR RTI PS
2523  013072  001401                           BEQ     .+4             ;BRANCH IF OK
2524  013074  104002                           HLT+2                   ;RTI PS NOT EQUAL TO 340
2525
```

```
2526  013076  022767  043125  165504          CMP     #043125,ANS3    ;CHECK FIRST HALF OF ANSWER
2527  013104  001401                          BEQ     .+4             ;BRANCH IF OK
2528  013106  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 043125
2529
2530  013110  022767  052526  165474          CMP     #052526,ANS4    ;CHECK SECOND HALF OF ANSWER
2531  013116  001401                          BEQ     .+4             ;BRANCH IF OK
2532  013120  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 052526
2533
2534  013122  010046                  4$:     MOV     R0,     -(SP)   ;RESTORE THE T-BIT
2535  013124  012746  013132                  MOV     #5$,    -(SP)
2536  013130  000002                          RTI
2537  013132  122767  000063  165640  5$:     CMPB    #63,    ICNT    ;CHECK THE TEST NUMBER
2538  013140  001401                          BEQ     .+4             ;BRANCH IF OK
2539  013142  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP,.
2540
2541  013144  104400                          SCOPE
2542
2543
2544                                  ;*****************************************************************
2545                                  ;TEST 64:       TEST THAT RED ZONE STACK OVERFLOW WORKS WITH FIS
2546                                  ;       STACK POINTER = SP = 0
2547                                  ;*****************************************************************
2548
2549  013146  012701  000332          TST64:  MOV     #332,   R1      ;SET UP STACK
2550  013152  012721  025177                  MOV     #025177,(R1)+   ;PUT DATA ON THE STACK
2551  013156  012721  177777                  MOV     #177777,(R1)+
2552  013162  012721  125200                  MOV     #125200,(R1)+
2553  013166  012721  000000                  MOV     #000000,(R1)+
2554  013172  012737  013256  000004          MOV     #3$,    @#4     ;SETUP STACK ERROR VECTOR
2555  013200  012737  000300  000006          MOV     #300,   @#6
2556  013206  013700  177776                  MOV     @#PS,   R0      ;SAVE THE T-BIT
2557  013212  012746  000340                  MOV     #340,   -(SP)   ;CLR T-BIT
2558  013216  012746  013224                  MOV     #1$,    -(SP)
2559  013222  000002                          RTI
2560  013224  012706  000332          1$:     MOV     #332,   SP      ;OVERFLOW THE STACK
2561
2562  013230  000240                          NOP
2563  013232  075006                          FADD+   SP              ;DO 125200 FLOATING POINT ADD
2564
2565  013234  016767  164536  165336  2$:     MOV     PS,     $PSW    ;SAVE PS FOR TYPING
2566  013242  010667  165334                  MOV     SP,     $SP     ;SAVE STACK POINTER
2567  013246  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK
2568  013252  104000                          HLT                     ;STACK OVERFLOW DIDN'T TRAP
2569
2570  013254  000473                          BR      4$
2571
2572  013256  016767  164514  165314  3$:     MOV     PS,     $PSW    ;SAVE THE PS
2573  013264  010667  165312                  MOV     SP,     $SP     ;SAVE THE STACK POINTER
2574  013270  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK
2575  013274  013767  000000  165302          MOV     @#0,    ANS1    ;SAVE RETURN ADR
2576  013302  013767  000002  165276          MOV     @#2,    ANS2    ;SAVE RETURN STATUS
2577  013310  012701  000332                  MOV     #332,   R1      ;POINT TO TOP OF ORIGINAL STACK
2578  013314  012702  000610                  MOV     #ANS3,  R2      ;TOP OF ANSWER TABLE
2579  013320  012122                          MOV     (R1)+,  (R2)+   ;SAVE THE STACK DATA
```

```
2580  013322  012122                          MOV     (R1)+,  (R2)+
2581  013324  012122                          MOV     (R1)+,  (R2)+
2582  013326  012122                          MOV     (R1)+,  (R2)+
2583  013330  022767  000300  165242          CMP     #300,   $PSW    ;CHECK THE PS AFTER THE TRAP
2584  013336  001401                          BEQ     .+4             ;BRANCH IF OK
2585  013340  104000                          HLT                     ;PS NOT EQUAL TO 300
2586
2587  013342  005767  165234                  TST     $SP             ;CHECK FOR SP AT RIGHT SPOT
2588  013346  001401                          BEQ     .+4             ;BRANCH IF OK
2589  013350  104000                          HLT                     ;STACK POINTER FOULED UP
2590
2591  013352  022767  013234  165224          CMP     #2$,    ANS1    ;CHECK TOP OF STACK FOR RTI ADR.
2592  013360  001401                          BEQ     .+4             ;BRANCH IF OK
2593  013362  104001                          HLT+1                   ;RTI ADDRESS NOT EQUAL TO #2$
2594
2595  013364  022767  000340  165214          CMP     #340,   ANS2    ;CHECK STACK DATA FOR RTI PS
2596  013372  001401                          BEQ     .+4             ;BRANCH IF OK
2597  013374  104002                          HLT+2                   ;RTI PS NOT EQUAL TO 340
2598
2599  013376  022767  025177  165204          CMP     #025177,ANS3    ;CHECK DATA FROM THE STACK
2600  013404  001401                          BEQ     .+4             ;BRANCH IF OK
2601  013406  104004                          HLT+4                   ;DATA ON STACK (025177) CHANGED
2602
2603  013410  022767  177777  165174          CMP     #177777,ANS4    ;CHECK DATA FROM THE STACK
2604  013416  001401                          BEQ     .+4             ;BRANCH IF OK
2605  013420  104004                          HLT+4                   ;DATA ON STACK (177777) CHANGED
2606
2607  013422  022767  125200  165164          CMP     #125200,ANS5    ;CHECK DATA FROM THE STACK
2608  013430  001401                          BEQ     .+4             ;BRANCH IF OK
2609  013432  104004                          HLT+4                   ;DATA ON STACK (125200) CHANGED
2610
2611  013434  005767  165156                  TST     ANS6            ;CHECK DATA FROM THE STACK
2612  013440  001401                          BEQ     .+4             ;BRANCH IF OK
2613  013442  104004                          HLT+4                   ;DATA ON STACK (000000) CHANGED
2614
2615  013444  010046                  4$:     MOV     R0,     -(SP)   ;RESTORE THE T-BIT
2616  013446  012746  013454                  MOV     #5$,    -(SP)
2617  013452  000002                          RTI
2618  013454  122767  000064  165316  5$:     CMPB    #64,    ICNT    ;CHECK THE TEST NUMBER
2619  013462  001401                          BEQ     .+4             ;BRANCH IF OK
2620  013464  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2621
2622  013466  104400                          SCOPE
2623
```

```
                                ;*****************************************************************
                                ;TEST 65:       TEST THAT RED ZONE STACK OVERFLOW WORKS WITH FIS
                                ;       STACK POINTER = SP = 0
                                ;*****************************************************************

013470  012701  000326          TST65:  MOV     #326,   R1      ;SET UP STACK
013474  012721  100125                  MOV     #100125,(R1)+   ;PUT DATA ON THE STACK
013500  012721  052525                  MOV     #052525,(R1)+
013504  012721  135753                  MOV     #135753,(R1)+
013510  012721  024642                  MOV     #024642,(R1)+
013514  012737  013600  000004          MOV     #3$,    @#4     ;SETUP STACK ERROR VECTOR
013522  012737  000300  000006          MOV     #300,   @#6
013530  013700  177776                  MOV     @#PS,   R0      ;SAVE THE T-BIT
013534  012746  000340                  MOV     #340,   -(SP)   ;CLR T-BIT
013540  012746  013546                  MOV     #1$,    -(SP)
013544  000002                          RTI
013546  012706  000326          1$:     MOV     #326,   SP      ;OVERFLOW THE STACK

013552  000240                          NOP
013554  075006                          FADD+   SP              ;DO 135753 FLOATING POINT ADD

013556  016767  164214  165014  2$:     MOV     PS,     $PSW    ;SAVE PS FOR TYPING
013564  010667  165012                  MOV     SP,     $SP     ;SAVE STACK POINTER
013570  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK
013574  104000                          HLT                     ;STACK OVERFLOW DIDN'T TRAP

013576  000474                          BR      4$

013600  016767  164172  164772  3$:     MOV     PS,     $PSW    ;SAVE THE PS
013606  010667  164770                  MOV     SP,     $SP     ;SAVE THE STACK POINTER
013612  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK
013616  013767  000000  164760          MOV     @#0,    ANS1    ;SAVE RETURN ADR
013624  013767  000002  164754          MOV     @#2,    ANS2    ;SAVE RETURN STATUS
013632  012701  000326                  MOV     #326,   R1      ;POINT TO TOP OF ORIGINAL STACK
013636  012702  000610                  MOV     #ANS3,  R2      ;TOP OF ANSWER TABLE
013642  012122                          MOV     (R1)+,  (R2)+   ;SAVE THE STACK DATA
013644  012122                          MOV     (R1)+,  (R2)+
013646  012122                          MOV     (R1)+,  (R2)+
013650  012122                          MOV     (R1)+,  (R2)+
013652  022767  000300  164720          CMP     #300,   $PSW    ;CHECK THE PS AFTER THE TRAP
013660  001401                          BEQ     .+4             ;BRANCH IF OK
013662  104000                          HLT                     ;PS NOT EQUAL TO 300

013664  005767  164712                  TST     $SP             ;CHECK FOR SP AT RIGHT SPOT
013670  001401                          BEQ     .+4             ;BRANCH IF OK
013672  104000                          HLT                     ;STACK POINTER FOULED UP

013674  022767  013556  164702          CMP     #2$,    ANS1    ;CHECK TOP OF STACK FOR RTI ADR.
013702  001401                          BEQ     .+4             ;BRANCH IF OK
013704  104001                          HLT+1                   ;RTI ADDRESS NOT EQUAL TO #2$

013706  022767  000340  164672          CMP     #340,   ANS2    ;CHECK STACK DATA FOR RTI PS
013714  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
2678  013716  104002                          HLT+2                   ;RTI PS NOT EQUAL TO 340
2679
2680  013720  022767  100125  164662          CMP     #100125,ANS3    ;CHECK DATA FROM THE STACK
2681  013726  001401                          BEQ     .+4             ;BRANCH IF OK
2682  013730  104004                          HLT+4                   ;DATA ON STACK (100125) CHANGED
2683
2684  013732  022767  052525  164652          CMP     #052525,ANS4    ;CHECK DATA FROM THE STACK
2685  013740  001401                          BEQ     .+4             ;BRANCH IF OK
2686  013742  104004                          HLT+4                   ;DATA ON STACK (052525) CHANGED
2687
2688  013744  022767  135753  164642          CMP     #135753,ANS5    ;CHECK DATA FROM THE STACK
2689  013752  001401                          BEQ     .+4             ;BRANCH IF OK
2690  013754  104004                          HLT+4                   ;DATA ON STACK (135753) CHANGED
2691
2692  013756  022767  024642  164632          CMP     #024642,ANS6    ;CHECK DATA FROM THE STACK
2693  013764  001401                          BEQ     .+4             ;BRANCH IF OK
2694  013766  104004                          HLT+4                   ;DATA ON STACK (024642) CHANGED
2695
2696  013770  010046                  4$:     MOV     R0,     -(SP)   ;RESTORE THE T-BIT
2697  013772  012746  014000                  MOV     #5$,    -(SP)
2698  013776  000002                          RTI
2699  014000  122767  000065  164772  5$:     CMPB    #65,    ICNT    ;CHECK THE TEST NUMBER
2700  014006  001401                          BEQ     .+4             ;BRANCH IF OK
2701  014010  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2702
2703  014012  104400                          SCOPE
2704
```

```
2705
2706                                    ;*************************************************************************
2707                                    ;TEST 66:       TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
2708                                    ;       INSTRUCTION = FADD,     STACK POINTER = R2
2709                                    ;*************************************************************************
2710
2711  014014  012737  014076  000004  TST66:  MOV     #ISR66, @#4     ;SET UP ADDRESS TRAP VECTOR
2712  014022  012737  000340  000006          MOV     #340,   @#6
2713  014030  004567  002266                  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
2714  014034  070707  016161                  ,WORD   070707,016161   ;SECOND OPERAND ON TOP
2715  014040  146314  143434                  ,WORD   146314,143434   ;FIRST OPERAND ON BOTTOM
2716  014044  000143                          ,WORD   143             ;PROCESSOR PRIORITY LEVEL
2717  014046  016606  000340                  ,WORD   TRAPER, 340     ;FIS TRAP VECTOR
2718  014052  012702  000631                  MOV     #STACK1,R2      ;SET UP R2 AS STACK POINTER
2719
2720  014056  000240                          NOP
2721  014060  075002                          FADD+   R2              ;FLOATING ADD ON THE R2 STACK
2722
2723  014062  004767  002266          RTA66:  JSR     %7,     POPR    ;POP THE "ANSWER"
2724  014066  010267  164510                  MOV     R2,     $SP     ;SAVE STACK POINTER (R2)
2725  014072  104002                          HLT+2                   ;FIS TRAP DIDN'T OCCURE!
2726  014074  000454                          BR      END66
2727
2728  014076  004767  002304          ISR66:  JSR     %7,     POPER   ;POP ALL DATA OFF THE STACKS
2729  014102  010267  164474                  MOV     R2,     $SP     ;SAVE STACK POINTER (R2)
2730  014106  022767  000340  164464          CMP     #340,   SPSW    ;CHECK PS AFTER ADR, ERR, TRAP
2731  014114  001401                          BEQ     .+4             ;BRANCH IF OK
2732  014116  104000                          HLT                     ;PS AFTER TRAP NOT EQUAL TO 340
2733
2734  014120  022767  000631  164454          CMP     #STACK1,$SP     ;CHECK THE STACK POINTER (R2)
2735  014126  001401                          BEQ     .+4             ;BRANCH IF OK
2736  014130  104000                          HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK1
2737
2738  014132  022767  014062  164444          CMP     #RTA66, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2739  014140  001401                          BEQ     .+4             ;BRANCH IF OK
2740  014142  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2741
2742  014144  022767  000141  164434          CMP     #141,   ANS2    ;CHECK PS BEFORE FIS TRAP
2743  014152  001401                          BEQ     .+4             ;BRANCH IF OK
2744  014154  104002                          HLT+2                   ;PS AT FIS TRAP TIME NOT 141
2745
2746  014156  022767  070707  164424          CMP     #070707,ANS3    ;CHECK DATA FROM THE STACK
2747  014164  001401                          BEQ     .+4             ;BRANCH IF OK
2748  014166  104004                          HLT+4                   ;DATA ON STACK (070707) CHANGED
2749
2750  014170  022767  016161  164414          CMP     #016161,ANS4    ;CHECK DATA FROM STACK
2751  014176  001401                          BEQ     .+4             ;BRANCH IF OK
2752  014200  104004                          HLT+4                   ;DATA ON STACK (016161) CHANGED
2753
2754  014202  022767  146314  164404          CMP     #146314,ANS5    ;CHECK DATA FROM STACK
2755  014210  001401                          BEQ     .+4             ;BRANCH IF OK
2756  014212  104006                          HLT+6                   ;DATA ON STACK (146314) CHANGED
2757
2758  014214  022767  143434  164374          CMP     #143434,ANS6    ;CHECK DATA FROM STACK
```

```
2759  014222  001401                          BEQ     ,+4             ;BRANCH IF OK
2760  014224  104006                          HLT+6                   ;DATA ON STACK (143434) CHANGED
2761
2762  014226  122767  000066  164544  END66:  CMPB    #66,    ICNT    ;CHECK THE TEST NUMBER
2763  014234  001401                          BEQ     ,+4             ;BRANCH IF OK
2764  014236  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2765
2766  014240  104400                          SCOPE
2767
2768
2769                                  ;*****************************************************************
2770                                  ;TEST 67:       TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
2771                                  ;       INSTRUCTION = FMUL,     STACK POINTER = R5
2772                                  ;*****************************************************************
2773
2774  014242  012737  014312  000004  TST67:  MOV     #ISR67, @#4     ;SET UP ADDRESS TRAP VECTOR
2775  014250  012737  000340  000006          MOV     #340,   @#6
2776  014256  012737  000202  177776          MOV     #202,   @#PS    ;SET PROCESSOR STATUS
2777  014264  012705  160000                  MOV     #160000,R5      ;SET UP R5 AS STACK POINTER
2778
2779  014270  000240                          NOP
2780  014272  075025                          FMUL+   R5              ;FLOATING MULTIPLY ON THE R5 STACK
2781
2782  014274  013767  177776  164276  RTA67:  MOV     @#PS,   SPSW    ;SAVE THE PSW
2783  014302  010567  164274                  MOV     R5,     SSP     ;SAVE STACK POINTER (R5)
2784  014306  104000                          HLT                     ;FIS TRAP DIDN'T OCCURE!
2785  014310  000430                          BR      END67
2786
2787  014312  004767  002070          ISR67:  JSR     %7,     POPER   ;POP ALL DATA OFF THE STACKS
2788  014316  010567  164260                  MOV     R5,     SSP     ;SAVE STACK POINTER (R5)
2789  014322  022767  000340  164250          CMP     #340,   SPSW    ;CHECK PS AFTER ADR, ERR, TRAP
2790  014330  001401                          BEQ     ,+4             ;BRANCH IF OK
2791  014332  104000                          HLT                     ;PS AFTER TRAP NOT EQUAL TO 340
2792
2793  014334  022767  160000  164240          CMP     #160000,SSP     ;CHECK THE STACK POINTER (R5)
2794  014342  001401                          BEQ     ,+4             ;BRANCH IF OK
2795  014344  104000                          HLT                     ;STACK POINTER (R5) NOT EQUAL TO #160000
2796
2797  014346  022767  014274  164230          CMP     #RTA67, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2798  014354  001401                          BEQ     ,+4             ;BRANCH IF OK
2799  014356  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2800
2801  014360  022767  000210  164220          CMP     #210,   ANS2    ;CHECK PS BEFORE FIS TRAP
2802  014366  001401                          BEQ     ,+4             ;BRANCH IF OK
2803  014370  104002                          HLT+2                   ;PS AT FIS TRAP TIME NOT 210
2804
2805  014372  122767  000067  164400  END67:  CMPB    #67,    ICNT    ;CHECK THE TEST NUMBER
2806  014400  001401                          BEQ     ,+4             ;BRANCH IF OK
2807  014402  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2808
2809  014404  104400                          SCOPE
2810
2811  014406  012737  000006  000004          MOV     #6,     @#4     ;RESTORE TIME-OUT VECTOR
2812  014414  005037  000006                  CLR     @#6
```

```
2813  014420  012767  000003  001514          MOV     #3,     TIMES   ;REDUCE NUMBER OF ITTERATIONS
2814
2815                                  ;*****************************************************************
2816                                  ;TEST 70;       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
2817                                  ;       101010,020202 + 000000,000000 = 101010,020202
2818                                  ;       PS = 144,       STACK POINTER = R1
2819                                  ;*****************************************************************
2820
2821  014426  012737  014526  000064  TST70;  MOV     #ISR70, @#64    ;SET UP TELEPRINTER INTERUPT VECTOR
2822  014434  012737  000200  000066          MOV     #200,   @#66
2823  014442  000004  017440                  TYPE,   RETURN          ;TYPE CARRIAGE RETURN, LINE FEED
2824  014446  012767  014454  001464          MOV     #.+6,   LADS    ;RESET LOOP ADDRESS
2825  014454  012777  000100  164172          MOV     #100,   @TPS    ;SET TTY INTERUPT ENABLE
2826  014462  012777  000100  164166          MOV     #100,   @TPB    ;TYPE "@"
2827  014470  004567  001626                  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
2828  014474  000000  000000                  .WORD   000000,000000   ;SECOND OPERAND ON TOP
2829  014500  101010  020202                  .WORD   101010,020202   ;FIRST OPERAND ON BOTTOM
2830  014504  000143                          .WORD   143             ;PROCESSOR PRIORITY LEVEL
2831  014506  016606  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
2832  014512  012701  000630                  MOV     #STACK0,R1      ;SET UP STACK POINTER
2833
2834  014516  000240                          NOP
2835  014520  075011                  RTA70;  FSUB+   R1              ;FLOATING SUBTRACT ON THE STACK
2836  014522  024141                          CMP     -(R1),  -(R1)   ;RESET THE STACK POINTER FOR NEXT PASS
2837  014524  000775                          BR      RTA70           ;REPEAT UNTIL INTERUPTED
2838
2839  014526  022716  014520          ISR70;  CMP     #RTA70, (SP)    ;CHECK IF INTERUPT AT FIS INSTR,
2840  014532  001410                          BEQ     1$              ;BRANCH IF IT DID
2841  014534  022766  014520  000004          CMP     #RTA70, 4(SP)   ;CHECK FOR INTERUPT WITH T-BIT SET
2842  014542  001407                          BEQ     2$              ;BRANCH IF IT DID
2843  014544  012777  000100  164104          MOV     #100,   @TPB    ;CONTINUE TO TYPE "@"
2844  014552  000002                          RTI
2845
2846  014554  004767  001626          1$:     JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
2847  014560  000406                          BR      3$
2848
2849  014562  013767  177776  164010  2$:     MOV     @#PS,   $PSW    ;SAVE THE SPW
2850  014570  022626                          CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
2851  014572  004767  001616                  JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
2852  014576  005077  164052          3$:     CLR     @TPS            ;CLR INTERUPT ENABLE
2853  014602  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
2854  014606  001406                          BEQ     ISA70           ;BRANCH IF OK
2855  014610  010667  163766                  MOV     SP,     $SP     ;SAVE FOR TYPING
2856  014614  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK POINTER
2857  014620  104000                          HLT                     ;STACK POINTER FOULED UP
2858  014622  000450                          BR      END70           ;SKIP REST OF TEST
2859
2860  014624  010167  163752          ISA70;  MOV     R1,     $SP     ;SAVE STACK POINTER
2861  014630  022767  000204  163742          CMP     #204,   $PSW    ;CHECK PS AFTER INTERUPT
2862  014636  001401                          BEQ     .+4             ;BRANCH IF OK
2863  014640  104000                          HLT                     ;PS AFTER INTERUPT NOT EQUAL TO LVLA
2864
2865  014642  022767  000630  163732          CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
2866  014650  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
2867  014652  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
2868
2869  014654  022767  014520  163722          CMP     #RTA70, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2870  014662  001401                          BEQ     .+4             ;BRANCH IF OK
2871  014664  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2872
2873  014666  022767  000144  163712          CMP     #144,   ANS2    ;CHECK PS BEFORE INTERUPT
2874  014674  001401                          BEQ     .+4             ;BRANCH IF OK
2875  014676  104002                          HLT+2                   ;PS AT INTERUPT TIME NOT 144
2876
2877  014700  005767  163704                  TST     ANS3            ;CHECK DATA FROM THE STACK
2878  014704  001401                          BEQ     .+4             ;BRANCH IF OK
2879  014706  104004                          HLT+4                   ;DATA ON STACK (000000) CHANGED
2880
2881  014710  005767  163676                  TST     ANS4            ;CHECK DATA FROM STACK
2882  014714  001401                          BEQ     .+4             ;BRANCH IF OK
2883  014716  104004                          HLT+4                   ;DATA ON STACK (000000) CHANGED
2884
2885  014720  022767  101010  163666          CMP     #101010,ANS5    ;CHECK DATA FROM STACK
2886  014726  001401                          BEQ     .+4             ;BRANCH IF OK
2887  014730  104006                          HLT+6                   ;DATA ON STACK (101010) CHANGED
2888
2889  014732  022767  020202  163656          CMP     #020202,ANS6    ;CHECK DATA FROM STACK
2890  014740  001401                          BEQ     .+4             ;BRANCH IF OK
2891  014742  104006                          HLT+6                   ;DATA ON STACK (020202) CHANGED
2892
2893  014744  122767  000070  164026  END70:  CMPB    #70,    ICNT    ;CHECK THE TEST NUMBER
2894  014752  001401                          BEQ     .+4             ;BRANCH IF OK
2895  014754  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2896
2897  014756  104400                          SCOPE
2898
2899
2900                                  ;*************************************************************************
2901                                  ;TEST 71:       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
2902                                  ;       123456,123456 / 040200,000000 = 123456,123456
2903                                  ;       PS = 051,       STACK POINTER = R4
2904                                  ;*************************************************************************
2905
2906  014760  012737  015060  000064  TST71:  MOV     #ISR71, @#64    ;SET UP TELEPRINTER INTERUPT VECTOR
2907  014766  012737  000200  000066          MOV     #200,   @#66
2908  014774  000004  017440                  TYPE,   RETURN          ;TYPE CARRIAGE RETURN, LINE FEED
2909  015000  012767  015006  001132          MOV     #.+6,   LADS    ;RESET LOOP ADDRESS
2910  015006  012777  000100  163640          MOV     #100,   @TPS    ;SET TTY INTERUPT ENABLE
2911  015014  012777  000100  163634          MOV     #100,   @TPB    ;TYPE "@"
2912  015022  004567  001274                  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
2913  015026  040200  000000                  .WORD   040200,000000   ;SECOND OPERAND ON TOP
2914  015032  123456  123456                  .WORD   123456,123456   ;FIRST OPERAND ON BOTTOM
2915  015036  000040                          .WORD   040             ;PROCESSOR PRIORITY LEVEL
2916  015040  016606  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
2917  015044  012704  000630                  MOV     #STACK0,R4      ;SET UP STACK POINTER
2918
2919  015050  000240                          NOP
2920  015052  075034                  RTA71:  FDIV+   R4               ;FLOATING DIVIDE ON THE STACK
```

```
2921  015054  024444                          CMP     -(R4),  -(R4)   ;RESET THE STACK POINTER FOR NEXT PASS
2922  015056  000775                          BR      RTA71           ;REPEAT UNTIL INTERUPTED
2923
2924  015060  022716  015052          ISR71:  CMP     #RTA71, (SP)    ;CHECK IF INTERUPT AT FIS INSTR,
2925  015064  001410                          BEQ     1$              ;BRANCH IF IT DID
2926  015066  022766  015052  000004        CMP     #RTA71, 4(SP)   ;CHECK FOR INTERUPT WITH T-BIT SET
2927  015074  001407                          BEQ     2$              ;BRANCH IF IT DID
2928  015076  012777  000100  163552        MOV     #100,   @TPB    ;CONTINUE TO TYPE "@"
2929  015104  000002                          RTI
2930
2931  015106  004767  001274          1$:     JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
2932  015112  000406                          BR      3$
2933
2934  015114  013767  177776  163456  2$:     MOV     @#PS,   $PSW    ;SAVE THE SPW
2935  015122  022626                          CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
2936  015124  004767  001264                  JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
2937  015130  005077  163520          3$:     CLR     @TPS            ;CLR INTERUPT ENABLE
2938  015134  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
2939  015140  001406                          BEQ     ISA71           ;BRANCH IF OK
2940  015142  010667  163434                  MOV     SP,     $SP     ;SAVE FOR TYPING
2941  015146  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK POINTER
2942  015152  104000                          HLT                     ;STACK POINTER FOULED UP
2943  015154  000451                          BR      END71           ;SKIP REST OF TEST
2944
2945  015156  010467  163420          ISA71:  MOV     R4,     $SP     ;SAVE STACK POINTER
2946  015162  022767  000204  163410        CMP     #204,   $PSW    ;CHECK PS AFTER INTERUPT
2947  015170  001401                          BEQ     .+4             ;BRANCH IF OK
2948  015172  104000                          HLT                     ;PS AFTER INTERUPT NOT EQUAL TO LVLA
2949
2950  015174  022767  000630  163400        CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R4)
2951  015202  001401                          BEQ     .+4             ;BRANCH IF OK
2952  015204  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK0
2953
2954  015206  022767  015052  163370        CMP     #RTA71, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2955  015214  001401                          BEQ     .+4             ;BRANCH IF OK
2956  015216  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2957
2958  015220  022767  000051  163360        CMP     #051,   ANS2    ;CHECK PS BEFORE INTERUPT
2959  015226  001401                          BEQ     .+4             ;BRANCH IF OK
2960  015230  104002                          HLT+2                   ;PS AT INTERUPT TIME NOT 051
2961
2962  015232  022767  040200  163350        CMP     #040200,ANS3    ;CHECK DATA FROM THE STACK
2963  015240  001401                          BEQ     .+4             ;BRANCH IF OK
2964  015242  104004                          HLT+4                   ;DATA ON STACK (040200) CHANGED
2965
2966  015244  005767  163342                  TST     ANS4            ;CHECK DATA FROM STACK
2967  015250  001401                          BEQ     .+4             ;BRANCH IF OK
2968  015252  104004                          HLT+4                   ;DATA ON STACK (000000) CHANGED
2969
2970  015254  022767  123456  163332        CMP     #123456,ANS5    ;CHECK DATA FROM STACK
2971  015262  001401                          BEQ     .+4             ;BRANCH IF OK
2972  015264  104006                          HLT+6                   ;DATA ON STACK (123456) CHANGED
2973
2974  015266  022767  123456  163322        CMP     #123456,ANS6    ;CHECK DATA FROM STACK
```

```
2975  015274  001401                          BEQ     .+4             ;BRANCH IF OK
2976  015276  104006                          HLT+6                   ;DATA ON STACK (123456) CHANGED
2977
2978  015300  122767  000071  163472  END71:  CMPB    #71,    ICNT    ;CHECK THE TEST NUMBER
2979  015306  001401                          BEQ     .+4             ;BRANCH IF OK
2980  015310  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2981
2982  015312  104400                          SCOPE
2983
2984
2985                                  ;*****************************************************************
2986                                  ;TEST 72:       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
2987                                  ;       107070,070707 * 040200,000000 = 107070,070707
2988                                  ;       PS = 111,       STACK POINTER = R0
2989                                  ;*****************************************************************
2990
2991  015314  012737  015414  000064  TST72:  MOV     #ISR72, @#64    ;SET UP TELEPRINTER INTERUPT VECTOR
2992  015322  012737  000200  000066          MOV     #200,   @#66
2993  015330  000004  017440                  TYPE,   RETURN          ;TYPE CARRIAGE RETURN, LINE FEED
2994  015334  012767  015342  000576          MOV     #.+6,   LADS    ;RESET LOOP ADDRESS
2995  015342  012777  000100  163304          MOV     #100,   @TPS    ;SET TTY INTERUPT ENABLE
2996  015350  012777  000100  163300          MOV     #100,   @TPB    ;TYPE "@"
2997  015356  004567  000740                  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
2998  015362  040200  000000                  .WORD   040200,000000   ;SECOND OPERAND ON TOP
2999  015366  107070  070707                  .WORD   107070,070707   ;FIRST OPERAND ON BOTTOM
3000  015372  000100                          .WORD   100             ;PROCESSOR PRIORITY LEVEL
3001  015374  016606  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
3002  015400  012700  000630                  MOV     #STACK0,R0      ;SET UP STACK POINTER
3003
3004  015404  000240                          NOP
3005  015406  075020                  RTA72:  FMUL+   R0              ;FLOATING MULTIPLY ON THE STACK
3006  015410  024040                          CMP     -(R0),  -(R0)   ;RESET THE STACK POINTER FOR NEXT PASS
3007  015412  000775                          BR      RTA72           ;REPEAT UNTIL INTERUPTED
3008
3009  015414  022716  015406          ISR72:  CMP     #RTA72, (SP)    ;CHECK IF INTERUPT AT FIS INSTR.
3010  015420  001410                          BEQ     1$              ;BRANCH IF IT DID
3011  015422  022766  015406  000004          CMP     #RTA72, 4(SP)   ;CHECK FOR INTERUPT WITH T-BIT SET
3012  015430  001407                          BEQ     2$              ;BRANCH IF IT DID
3013  015432  012777  000100  163216          MOV     #100,   @TPB    ;CONTINUE TO TYPE "@"
3014  015440  000002                          RTI
3015
3016  015442  004767  000740          1$:     JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
3017  015446  000406                          BR      3$
3018
3019  015450  013767  177776  163122  2$:     MOV     @#PS,   $PSW    ;SAVE THE SPW
3020  015456  022626                          CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
3021  015460  004767  000730                  JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
3022  015464  005077  163164          3$:     CLR     @TPS            ;CLR INTERUPT ENABLE
3023  015470  022706  000500                  CMP     #500,   SP      ;CHECK THE STACK POINTER
3024  015474  001406                          BEQ     ISA72           ;BRANCH IF OK
3025  015476  010667  163100                  MOV     SP,     $SP     ;SAVE FOR TYPING
3026  015502  012706  000500                  MOV     #500,   SP      ;RESTORE THE STACK POINTER
3027  015506  104000                          HLT                     ;STACK POINTER FOULED UP
3028  015510  000451                          BR      END72           ;SKIP REST OF TEST
```

```
3029
3030  015512  010067  163064          ISA72:  MOV     R0,     $SP     ;SAVE STACK POINTER
3031  015516  022767  000204  163054          CMP     #204,   $PSW    ;CHECK PS AFTER INTERUPT
3032  015524  001401                          BEQ     .+4             ;BRANCH IF OK
3033  015526  104000                          HLT                     ;PS AFTER INTERUPT NOT EQUAL TO LVLA
3034
3035  015532  022767  000630  163044          CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R0)
3036  015536  001401                          BEQ     .+4             ;BRANCH IF OK
3037  015540  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK0
3038
3039  015542  022767  015406  163034          CMP     #RTA72, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
3040  015550  001401                          BEQ     .+4             ;BRANCH IF OK
3041  015552  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
3042
3043  015554  022767  000111  163024          CMP     #111,   ANS2    ;CHECK PS BEFORE INTERUPT
3044  015562  001401                          BEQ     .+4             ;BRANCH IF OK
3045  015564  104002                          HLT+2                   ;PS AT INTERUPT TIME NOT 111
3046
3047  015566  022767  040200  163014          CMP     #040200,ANS3    ;CHECK DATA FROM THE STACK
3048  015574  001401                          BEQ     .+4             ;BRANCH IF OK
3049  015576  104004                          HLT+4                   ;DATA ON STACK (040200) CHANGED
3050
3051  015600  005767  163006                  TST     ANS4            ;CHECK DATA FROM STACK
3052  015604  001401                          BEQ     .+4             ;BRANCH IF OK
3053  015606  104004                          HLT+4                   ;DATA ON STACK (000000) CHANGED
3054
3055  015610  022767  107070  162776          CMP     #107070,ANS5    ;CHECK DATA FROM STACK
3056  015616  001401                          BEQ     .+4             ;BRANCH IF OK
3057  015620  104006                          HLT+6                   ;DATA ON STACK (107070) CHANGED
3058
3059  015622  022767  070707  162766          CMP     #070707,ANS6    ;CHECK DATA FROM STACK
3060  015630  001401                          BEQ     .+4             ;BRANCH IF OK
3061  015632  104006                          HLT+6                   ;DATA ON STACK (070707) CHANGED
3062
3063  015634  122767  000072  163136  END72:  CMPB    #72,    ICNT    ;CHECK THE TEST NUMBER
3064  015642  001401                          BEQ     .+4             ;BRANCH IF OK
3065  015644  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
3066
3067  015646  104400                          SCOPE
3068
3069  015650  012767  000377  000264          MOV     #377,   TIMES   ;RESET NUMBER OF ITERATIONS
3070  015656  012737  000062  000060          MOV     #62,    @#60    ;RESTORE TTY INTERUPT VECTOR
3071  015664  005037  000062                  CLR     @#62
3072
```

```
3073  015670  000240                    DONE:   NOP
3074  015672  032737  002000  177570            BIT     #SW10,@#SWR     ;RING THE BELL?
3075  015700  001002                            BNE     1$              ;NO!
3076  015702  000004  000007                    TYPE    ,BELL
3077  015706  005046                    1$:     CLR     -(6)            ;CLEAR TRACE TRAP
3078  015710  032737  010000  177570            BIT     #SW12,@#SWR     ;RUN WITH TRT?
3079  015716  001010                            BNE     2$
3080  015720  005167  000056                    COM     ,TBIT
3081  015724  100005                            BPL     2$
3082  015726  052716  000020                    BIS     #20,(6)         ;SET TRACE TRAP
3083  015732  012746  015764                    MOV     #3$,-(6)        ;JUMP TO START OF TEST
3084  015736  000002                            RTI
3085  015740  012746  015746            2$:     MOV     #4$,-(6)        ;JUMP TO START OF TEST
3086  015744  000002                            RTI                     ;RETURN
3087  015746  013700  000042            4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
3088  015752  001404                            BEQ     3$              ;IF NONE
3089  015754  004710                            JSR     7,(0)           ;GO TO MONITOR
3090  015756  000240                            NOP
3091  015760  000240                            NOP
3092  015762  000240                            NOP
3093  015764  062767  000001  163014    3$:     ADD     #1,PASSES+2     ;INC PASS COUNTER
3094  015772  005567  163006                    ADC     PASSES
3095  015776  000137  000200                    JMP     @#200           ;RETURN
3096
3097  016002  000000                    ,TBIT:  0
3098
3099  016004  000006                    YESRT:  RTT                     ;RETURN FROM TRACE TRAP
3100  016006  032737  000400  177570    SCOPE$: BIT     #SW08,@#SWR     ;KILL LOOP ON SPEC, TEST
3101  016014  001404                            BEQ     1$
3102  016016  123767  177570  162754            CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   *SW7=0*
3103  016024  001434                            BEQ     OVER$
3104  016026  032737  040000  177570    1$:     BIT     #SW14,@#SWR     ;LOOP ON TEST
3105  016034  001026                            BNE     KIT$
3106  016036  032737  004000  177570            BIT     #SW11,@#SWR     ;KILL ITERATIONS
3107  016044  001012                            BNE     SVLAD$
3108  016046  105767  162727                    TSTB    ICNT+1
3109  016052  001404                            BEQ     2$              ;BRANCH IF FIRST
3110  016054  126767  000062  162717            CMPB    TIMES,ICNT+1    ;DONE?
3111  016062  001013                            BNE     KIT$            ;BRANCH IF NOT
3112  016064  112767  000001  162707    2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
3113  016072  105267  162702            SVLAD$: INCB    ICNT            ;COUNT TEST NUMBERS
3114  016076  011667  000036                    MOV     (6),LAD$        ;SAVE LOOP ADDRESS
3115  016102  016737  162672  177570            MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO, AND ITERATION COUNT
3116  016110  000002                            RTI                     ;RETURN
3117
3118  016112  105267  162663            KIT$:   INCB    ICNT+1
3119  016116  016737  162656  177570    OVER$:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
3120  016124  005767  000010                    TST     LAD$            ;FIRST ONE?
3121  016130  001760                            BEQ     SVLAD$
3122  016132  016716  000002                    MOV     LAD$,(6)        ;FUDGE RETURN ADDRESS
3123  016136  000002                            RTI                     ;FIXES PS
3124
3125  016140  000000                    LAD$:   0                       ;LOOP ADDRESS
3126  016142  000377                    TIMES:  377                     ;RUN 377 TIMES
```

```
3127
3128                                    ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
3129
3130  016144  005726                    PUSHS:  TST     (SP)+           ;POP STACK BY 1
3131  016146  062705  000010                    ADD     #10,    R5      ;POINT TO END OF DATA
3132  016152  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3133  016154  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3134  016156  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3135  016160  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3136  016162  062705  000010                    ADD     #10,    R5      ;POINT TO END OF DATA
3137  016166  012537  177776                    MOV     (R5)+,  @#PS    ;SET THE PROCESSOR STATUS
3138  016172  012577  162452                    MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
3139  016176  012577  162450                    MOV     (R5)+,  @FISLVL ;TRAP STATUS
3140  016202  000115                            JMP     (R5)            ;RETURN
3141
3142
3143                                    ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
3144                                    ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
3145
3146  016204  013767  177776  162366  POPS:   MOV     @#PS,   $PSW    ;SAVE PROCESSOR STATUS WORD
3147  016212  042767  000020  162360          BIC     #20,    $PSW    ;CLEAR T-BIT
3148  016220  012604                          MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
3149  016222  012667  162356                  MOV     (SP)+,  ANS1    ;SAVE THE ANSWER
3150  016226  012667  162354                  MOV     (SP)+,  ANS2    ;
3151  016232  010667  162344                  MOV     SP,     $SP     ;SAVE THE STACK POINTER
3152  016236  000114                          JMP     (R4)            ;RETURN
3153
3154
3155                                    ;SUBROUTNE TO POP 6 WORDS OFF THE STACK,
3156                                    ;THE FIRST TWO WERE PUT ON BY THE ERROR TRAP,
3157                                    ;THE LAST FOUR WERE THE ORIGINAL INPUT DATA,
3158                                    ;ALSO SAVES THE PS AND STACK POINTER,
3159
3160  016240  013767  177776  162332  POPES:  MOV     @#PS,   $PSW    ;SAVE PROCESSOR STATUS WORD
3161  016246  012604                          MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
3162  016250  012667  162330                  MOV     (SP)+,  ANS1    ;SAVE RTI ADDRESS
3163  016254  011667  162326                  MOV     (SP),   ANS2    ;SAVE RTI STATUS
3164  016260  042767  000020  162320          BIC     #20,    ANS2    ;CLEAR THE T-BIT
3165  016266  012746  016274                  MOV     #1$,    -(SP)
3166  016272  000002                          RTI                     ;RESTORE THE PROCESSOR STATUS
3167  016274  012667  162310          1$:     MOV     (SP)+,  ANS3    ;SAVE DATA
3168  016300  012667  162306                  MOV     (SP)+,  ANS4    ;
3169  016304  012667  162304                  MOV     (SP)+,  ANS5    ;
3170  016310  012667  162302                  MOV     (SP)+,  ANS6    ;
3171  016314  010667  162262                  MOV     SP,     $SP     ;SAVE SP
3172  016320  000114                          JMP     (R4)            ;RTS
3173
3174                                    ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
3175
3176  016322  012704  000630          PUSHR:  MOV     #STACK0,R4      ;SET R4 TO STACK
3177  016326  012524                          MOV     (R5)+,  (R4)+   ;PUT DATA ON STACK
3178  016330  012524                          MOV     (R5)+,  (R4)+   ;
3179  016332  012524                          MOV     (R5)+,  (R4)+   ;
3180  016334  012524                          MOV     (R5)+,  (R4)+   ;
```

```
3073  015670  000240                    DONE:   NOP
3074  015672  032737  002000  177570            BIT     #SW10,@#SWR     ;RING THE BELL?
3075  015700  001002                            BNE     1$              ;NO!
3076  015702  000004  000007                    TYPE    ,BELL
3077  015706  005046                    1$:     CLR     -(6)            ;CLEAR TRACE TRAP
3078  015710  032737  010000  177570            BIT     #SW12,@#SWR     ;RUN WITH TRT?
3079  015716  001010                            BNE     2$
3080  015720  005167  000056                    COM     ,TBIT
3081  015724  100005                            BPL     2$
3082  015726  052716  200020                    BIS     #20,(6)         ;SET TRACE TRAP
3083  015732  012746  015764                    MOV     #3$,-(6)        ;JUMP TO START OF TEST
3084  015736  000002                            RTI
3085  015740  012746  015746            2$:     MOV     #4$,-(6)        ;JUMP TO START OF TEST
3086  015744  000002                            RTI                     ;RETURN
3087  015746  013700  000042            4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
3088  015752  001404                            BEQ     3$              ;IF NONE
3089  015754  004710                            JSR     7,(0)           ;GO TO MONITOR
3090  015756  000240                            NOP
3091  015760  000240                            NOP
3092  015762  000240                            NOP
3093  015764  062767  000001  163014    3$:     ADD     #1,PASSES+2     ;INC PASS COUNTER
3094  015772  005567  163006                    ADC     PASSES
3095  015776  000137  000200                    JMP     @#200           ;RETURN
3096
3097  016002  000000                    ,TBIT:  0
3098
3099  016004  000006                    YESRT:  RTT                     ;RETURN FROM TRACE TRAP
3100  016006  032737  000400  177570    SCOPE$: BIT     #SW08,@#SWR     ;KILL LOOP OR LOOP ON SPEC, TEST
3101  016014  001404                            BEQ     1$
3102  016016  123767  177570  162754            CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   *SW7=0*
3103  016024  001434                            BEQ     OVER$
3104  016026  032737  040000  177570    1$:     BIT     #SW14,@#SWR     ;LOOP ON TEST
3105  016034  001026                            BNE     KIT$
3106  016036  032737  004000  177570            BIT     #SW11,@#SWR     ;KILL ITERATIONS
3107  016044  001012                            BNE     SVLAD$
3108  016046  105767  162727                    TSTB    ICNT+1
3109  016052  001404                            BEQ     2$              ;BRANCH IF FIRST
3110  016054  126767  000062  162717            CMPB    TIMES,ICNT+1    ;DONE?
3111  016062  001013                            BNE     KIT$            ;BRANCH IF NOT
3112  016064  112767  000001  162707    2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
3113  016072  105267  162702            SVLAD$: INCB    ICNT            ;COUNT TEST NUMBERS
3114  016076  011667  000036                    MOV     (6),LAD$        ;SAVE LOOP ADDRESS
3115  016102  016737  162672  177570            MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO, AND ITERATION COUNT
3116  016110  000002                            RTI                     ;RETURN
3117
3118  016112  105267  162663            KIT$:   INCB    ICNT+1
3119  016116  016737  162656  177570    OVER$:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
3120  016124  005767  000010                    TST     LAD$            ;FIRST ONE?
3121  016130  001760                            BEQ     SVLAD$
3122  016132  016716  000002                    MOV     LAD$,(6)        ;FUDGE RETURN ADDRESS
3123  016136  000002                            RTI                     ;FIXES PS
3124
3125  016140  000000                    LAD$:   0                       ;LOOP ADDRESS
3126  016142  000377                    TIMES:  377                     ;RUN 377 TIMES
```

```
3127
3128                                    ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
3129
3132  216144  005726                    PUSHS:  TST     (SP)+           ;POP STACK BY 1
3131  216146  062705  000010                    ADD     #10,    R5      ;POINT TO END OF DATA
3132  216152  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3133  216154  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3134  216156  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3135  216160  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3136  216162  062705  000010                    ADD     #10,    R5      ;POINT TO END OF DATA
3137  216166  012537  177776                    MOV     (R5)+,  @#PS    ;SET THE PROCESSOR STATUS
3138  216172  012577  162452                    MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
3139  216176  012577  162450                    MOV     (R5)+,  @FISLVL ;TRAP STATUS
3140  216202  000115                            JMP     (R5)            ;RETURN
3141
3142
3143                                    ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
3144                                    ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
3145
3146  216204  013767  177776  162366  POPS:   MOV     @#PS,   $PSW    ;SAVE PROCESSOR STATUS WORD
3147  216212  042767  000020  162360          BIC     #20,    $PSW    ;CLEAR T-BIT
3148  216220  012604                          MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
3149  216222  012667  162356                  MOV     (SP)+,  ANS1    ;SAVE THE ANSWER
3150  216226  012667  162354                  MOV     (SP)+,  ANS2    ;
3151  216232  010667  162344                  MOV     SP,     $SP     ;SAVE THE STACK POINTER
3152  216236  000114                          JMP     (R4)            ;RETURN
3153
3154
3155                                    ;SUBROUTNE TO POP 6 WORDS OFF THE STACK,
3156                                    ;THE FIRST TWO WERE PUT ON BY THE ERROR TRAP,
3157                                    ;THE LAST FOUR WERE THE ORIGINAL INPUT DATA,
3158                                    ;ALSO SAVES THE PS AND STACK POINTER,
3159
3160  216240  013767  177776  162332  POPES:  MOV     @#PS,   $PSW    ;SAVE PROCESSOR STATUS WORD
3161  216246  012604                          MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
3162  216250  012667  162330                  MOV     (SP)+,  ANS1    ;SAVE RTI ADDRESS
3163  216254  011667  162326                  MOV     (SP),   ANS2    ;SAVE RTI STATUS
3164  216260  042767  000020  162320          BIC     #20,    ANS2    ;CLEAR THE T-BIT
3165  216266  012746  016274                  MOV     #1$,    -(SP)
3166  216272  000002                          RTI                     ;RESTORE THE PROCESSOR STATUS
3167  216274  012667  162310          1$:     MOV     (SP)+,  ANS3    ;SAVE DATA
3168  216300  012667  162306                  MOV     (SP)+,  ANS4    ;
3169  216304  012667  162304                  MOV     (SP)+,  ANS5    ;
3170  216310  012667  162302                  MOV     (SP)+,  ANS6    ;
3171  216314  010667  162262                  MOV     SP,     $SP     ;SAVE SP
3172  216320  000114                          JMP     (R4)            ;RTS
3173
3174                                    ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
3175
3176  216322  012704  000630          PUSHR:  MOV     #STACK0,R4      ;SET R4 TO STACK
3177  216326  012524                          MOV     (R5)+,  (R4)+   ;PUT DATA ON STACK
3178  216330  012524                          MOV     (R5)+,  (R4)+   ;
3179  216332  012524                          MOV     (R5)+,  (R4)+   ;
3180  216334  012524                          MOV     (R5)+,  (R4)+   ;
```

```
3181  016336  012537  177776                MOV     (R5)+,  @#PS     ;SET THE PROCESSOR STATUS
3182  016342  012577  162302                MOV     (R5)+,  @FISVEC  ;SET UP FIS ERROR TRAP VECTOR
3183  016346  012577  162300                MOV     (R5)+,  @FISLVL  ;TRAP STATUS
3184  016352  000205                        RTS     R5               ;RETURN
3185
3186
3187                                  ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
3188                                  ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
3189
3190  016354  013767  177776  162216  POPR:   MOV     @#PS,   $PSW     ;SAVE PROCESSOR STATUS WORD
3191  016362  042767  000020  162210          BIC     #20,    $PSW     ;CLEAR T-BIT
3192  016370  016767  162240  162206          MOV     STACK4, ANS1     ;SAVE THE ANSWER
3193  016376  016767  162234  162202          MOV     STACK6, ANS2     ;
3194  016404  000227                          RTS     %7
3195
3196
3197                                  ;SUBROUTNE TO POP 6 WORDS OFF THE STACKS,
3198                                  ;THE TWO OFF THE R6 STACK WERE PUT ON BY THE ERROR TRAP,
3199                                  ;THE FOUR OFF THE SOFTWARE STACK WERE THE ORIGINAL INPUT DATA.
3200                                  ;ALSO SAVES THE PS AND STACK POINTER AFTER THE FIS TRAP.
3201
3202  016406  013767  177776  162164  POPER1: MOV     @#PS,   $PSW     ;SAVE PROCESSOR STATUS WORD
3203  016414  012667  000056          POPER1: MOV     (SP)+,  SAVRTS   ;SAVE RTS ADDRESS
3204  016420  012667  162160                  MOV     (SP)+,  ANS1     ;SAVE RTI ADDRESS
3205  016424  011667  162156                  MOV     (SP),   ANS2     ;SAVE RTI STATUS
3206  016430  042767  000020  162150          BIC     #20,    ANS2     ;CLEAR THE T-BIT
3207  016436  012746  016444                  MOV     #1$,    -(SP)
3208  016442  000002                          RTI                      ;RESTORE PROCESSOR STATUS
3209  016444  016767  162160  162136  1$:     MOV     STACK0, ANS3     ;SAVE DATA
3210  016452  016767  162154  162132          MOV     STACK2, ANS4     ;
3211  016460  016767  162150  162126          MOV     STACK4, ANS5     ;
3212  016466  016767  162144  162122          MOV     STACK6, ANS6     ;
3213  016474  000137                          JMP     @(%7)+           ;SIMULATED RTS
3214  016476  000000                  SAVRTS: 0                        ;RTS ADDRESS SAVE
3215
3216                                  ;SUBROUTINE TO PUSH 4 WORDS ONTO THE PC STACK
3217
3218  016500  012504                  PUSH7:  MOV     (R5)+,  R4       ;SET R4 TO STACK
3219  016502  012524                          MOV     (R5)+,  (R4)+    ;PUT DATA ON STACK
3220  016504  012524                          MOV     (R5)+,  (R4)+    ;
3221  016506  012524                          MOV     (R5)+,  (R4)+    ;
3222  016510  012524                          MOV     (R5)+,  (R4)+    ;
3223  016512  042737  177757  177776          BIC     #177757,@#PS     ;CLEAR STATUS EXCEPT T-BIT
3224  016520  052537  177776                  BIS     (R5)+,  @#PS     ;SET THE PROCESSOR STATUS
3225  016524  012577  162120                  MOV     (R5)+,  @FISVEC  ;SET UP FIS ERROR TRAP VECTOR
3226  016530  012577  162116                  MOV     (R5)+,  @FISLVL  ;TRAP STATUS
3227  016534  000205                          RTS     R5               ;RETURN
3228
3229                                  ;SUBROUTINE TO POP 4 WORDS OFF THE PC "STACK"
3230                                  ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
3231
3232  016536  013767  177776  162034  POP7:   MOV     @#PS,   $PSW     ;SAVE PROCESSOR STATUS WORD
3233  016544  042767  000020  162026          BIC     #20,    $PSW     ;CLEAR T-BIT
3234  016552  011600                          MOV     (SP),   R0       ;GET RETURN ADDRESS
```

```
3235  016554  162700  000014              SUB     #14,    R0      ;POINT TO TOP OF "PC STACK"
3236  016560  012067  162020              MOV     (R0)+,  ANS1    ;SAVE 1ST HALF INPUT DATA
3237  016564  012067  162016              MOV     (R0)+,  ANS2    ;SAVE 2ND HALF INPUT DATA
3238  016570  010067  162006              MOV     R0,     $SP     ;SAVE ASSUMED END PC "STACK POINTER"
3239  016574  012067  162010              MOV     (R0)+,  ANS3    ;SAVE 1ST HALF OF ANSWER
3240  016600  012067  162006              MOV     (R0)+,  ANS4    ;SAVE 2ND HALF OF ANSWER
3241  016604  000207                      RTS     %7
3242
3243                                  ;ERRONIOUS TRAP SERVICE ROUTINE
3244
3245  016606  104000              TRAPER: HLT                     ;FIS SHOULDN'T HAVE TRAPED
3246  016610  000002                      RTI
3247
3248  016612  032737  002000  177570  HLT$:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
3249  016620  001402                      BEQ     1$              ;NO - SKIP
3250  016622  000004  000007              TYPE    ,BELL           ;RING BELL
3251  016626  005267  162150          1$: INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
3252  016632  032737  020000  177570      BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
3253  016640  001017                      BNE     2$              ;SKIP TYPEOUTS
3254  016642  000004  017440              TYPE    ,RETURN
3255  016646  011667  000060              MOV     (6),HLTAD$      ;PUT ADDRESS OF INSTRUCTION ON STACK
3256  016652  162767  000002  000052      SUB     #2,HLTAD$
3257  016660  016705  000046              MOV     HLTAD$,TTY      ;TYPE HLTAD$ IN OCTAL
3258  016664  004767  000106              JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
3259  016670  000004  017446              TYPE    ,SPACE+3
3260  016674  004767  000034              JSR     PC,ERROR$       ;GO TO USER ERROR ROUTINE
3261  016700  005737  177570          2$: TST     @#SWR           ;HALT ON ERROR
3262  016704  100001                      BPL     .+4             ;SKIP IF CONTINUE
3263  016706  000000                      HALT                    ;HALT ON ERROR!
3264  016710  032737  001000  177570      BIT     #SW09,@#SWR     ;CHECK FOR INHIBIT LOOP ON ERROR
3265  016716  001001                      BNE     .+4             ;SKIP IF LOOP ON ERROR
3266  016720  000002                      RTI
3267  016722  105067  162053              CLRB    ICNT+1
3268  016726  000167  177160              JMP     KIT$            ;LOOP ON TEST UNTIL NO ERRORS
3269
3270  016732  000000              HLTAD$: 0
3271
3272  016734  117767  177772  000032  ERROR$: MOVB    @HLTAD$,TYPCNT  ;TYPE COUNT IS LOW BYTE OF HLT
3273  016742  105267  000026              INCB    TYPCNT          ;TYPE COUNT = X+1
3274  016746  012703  000600              MOV     #$PSW,  R3      ;TOP OF DATA TO BE TYPED
3275  016752                          ERR1$:
3276  016752  012305                      MOV     (R3)+,TTY       ;TYPE (R3)+ IN OCTAL
3277  016754  004767  000016              JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
3278  016760  000004  017447              TYPE,   SPACE+4         ;SPACE
3279  016764  105367  000004              DECB    TYPCNT          ;CHECK FOR DONE
3280  016770  100370                      BPL     ERR1$           ;BRANCH IF NOT DONE
3281  016772  000207                      RTS     PC
3282
3283  016774  000000              TYPCNT: 0
```

```
3284  017776  112767  000001  000130  PRINTR: MOVB    #1,,PR          ;SET ZERO FILL SWITCH
3285  020004  000402                  BR      ,+6             ;SKIP
3286  020006  005067  000122          PRINTS: CLR     ,PR             ;SUPRESS LEADING ZERO'S
3287  020012  112767  177772  000115          MOVB    #-6,,PR+1       ;SET COUNT
3288  020020  010446                  MOV     R4,-(6)         ;SAVE R4
3289  020022  012704  017124                  MOV     #,PRBUF,R4      ;SET POINTER TO FIRST ASCII CHAR,
3290  020026  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
3291  020030  000405                  BR      ,PRF            ;ROTATE FIRST BIT
3292  020032  105014          ,PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
3293  020034  006105                  ROL     TTY             ;ROTATE BIT INTO C
3294  020036  106114                  ROLB    (4)             ;PACK IT
3295  020040  006105                  ROL     TTY             ;ROTATE BIT INTO C
3296  020042  106114                  ROLB    (4)             ;PACK IT
3297  020044  006105          ,PRF:   ROL     TTY             ;ROTATE BIT INTO C
3298  020046  106114                  ROLB    (4)             ;PACK IT
3299  020050  105714                  TSTB    (4)             ;IS IT ZERO?
3300  020052  001402                  BEQ     ,+6             ;SKIP INC
3301  020054  105267  000054          INCB    ,PR             ;SET FILL SWITCH
3302  020060  105767  000050          TSTB    ,PR             ;CHECK FILL SWITCH
3303  020064  001402                  BEQ     ,+6             ;SKIP BITSET
3304  020066  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
3305  020072  105267  000037          INCB    ,PR+1           ;INC COUNT
3306  020076  001355                  BNE     ,PRL            ;REPEAT
3307  020100  022704  017124          CMP     #,PRBUF,R4      ;EMPTY BUFFER?
3308  020104  001002                  BNE     ,+6             ;SKIP IF NOT
3309  020106  112724  000060          MOVB    #'0,(4)+        ;LOAD 1 ZERO
3310  020112  105014                  CLRB    (4)             ;NULL TERMINATOR
3311  020114  000004  017124          TYPE    ,,PRBUF         ;TYPE IT
3312  020120  012604                  MOV     (6)+,R4         ;RESTORE R4
3313  020122  000207                  RTS     PC              ;RETURN
3314
3315  020124  000004          ,PRBUF: ,BLKW   4               ;OUTPUT BUFFER
3316  020134  000000          ,PR:    0                       ;COUNT AND SWITCH
```

```
3317  017136  012777  017304  000154  PDOWN$: MOV     #ILLUP,@PUVEC$  ;SET FOR FAST UP
3318  017144  012777  000340  000150          MOV     #340,@PUVEC$+2  ;PRIO!7
3319  017152  017767  161476  000132          MOV     @TPS,   SAVTPS  ;SAVE TELEPRINTER STATUS
3320  017160  010046                          MOV     R0,-(6)         ;PUSH R0 ON STACK
3321  017162  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
3322  017164  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
3323  017166  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
3324  017170  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
3325  017172  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
3326  017174  010667  000110                  MOV     SP,,SAVR6       ;SAVE SP
3327  017200  012777  017210  000112          MOV     #PUP$,@PUVEC$   ;SET UP VECTOR
3328  017206  000000                          HALT
3329
3330  017210  016706  000074          PUP$:   MOV     ,SAVR6,SP       ;GET SP
3331  017214  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
3332  017216  005201                  1$:     INC     R1              ;WAIT FOR THE INC
3333  017220  001376                          BNE     1$              ;OF A WORD
3334  017222  012605                          MOV     (6)+,R5         ;POP STACK INTO R5
3335  017224  012604                          MOV     (6)+,R4         ;POP STACK INTO R4
3336  017226  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
3337  017230  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
3338  017232  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
3339  017234  012600                          MOV     (6)+,R0         ;POP STACK INTO R0
3340  017236  012777  017136  000050          MOV     #PDOWN$,@PDVEC$ ;SET UP THE POWER DOWN VECTOR
3341  017244  012777  000340  000044          MOV     #340,@PDVEC$+2  ;PRIO!7
3342  017252  000004  017324                  TYPE    ,POWER$
3343  017256  032767  000100  000026          BIT     #100,   SAVTPS  ;CHECK INT ENB BIT
3344  017264  001406                          BEQ     2$              ;BRANCH IF NOT
3345  017266  012777  000100  161360          MOV     #100,   @TPS    ;SET INT ENB
3346  017274  012777  000100  161354          MOV     #100,   @TPB    ;TYPE AN "@"
3347  017302  000002          .       2$:     RTI
3348
3349  017304  000000                  ILLUP:  HALT                    ;THE POWER UP SEQUENCE WAS STARTED
3350  017306  000776                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
3351
3352  017310  000000                  ,SAVR6: 0                       ;PUT THE SP HERE
3353  017312  000000                  SAVTPS: 0                       ;LOC TO SAVE TELEPRINTER STATUS
3354  017314  000024  000026          PDVEC$: 24,26                   ;POWER DOWN VECTOR
3355  017320  000024  000026          PUVEC$: 24,26                   ;POWER UP VECTOR
3356  017324  005015  047520  042527  POWER$: ,ASCIZ  <15><12>"POWER"
3357  017332  000122
3358                                          ,EVEN
```

```
3359  017334  010546                  ;IOT:   MOV     TTY,-(6)          ;SAVE TTY
3360  017336  017605  000002                  MOV     @2(6),TTY         ;GET ADDRESS TO BE TYPED
3361  017342  032705  177400                  BIT     #177400,TTY       ;IS IT A TYPEM?
3362  017346  001004                          BNE     1$                ;NO
3363  017352  010567  000076                  MOV     TTY,,TYPE         ;GET THE CHARACTER
3364  017354  012705  017452                  MOV     #,TYPE,TTY        ;FUDGE THE ADDRESS
3365  017360  105715                  1$:     TSTB    (TTY)             ;TERMINATOR?
3366  017362  001406                          BEQ     2$                ;GET OUT IF SO
3367  017364  112537  177566                  MOVB    (TTY)+,@#177566   ;LOAD AND TYPE THE CHARACTER
3368  017370  105737  177564                  TSTB    @#177564          ;IS THE PRINTER READY
3369  017374  100375                          BPL     .-4               ;WAIT UNTIL IT IS
3370  017376  000770                          BR      1$                ;GET THE NEXT CHARACTER
3371  017400  017646  000002          2$:     MOV     @2(6),-(6)        ;GET ADDRESS TO BE TYPED
3372  017404  062766  000002  000004          ADD     #2,4(6)           ;ADD 2 TO THE ADDRESS
3373  017412  022666  000002                  CMP     (6)+,2(6)         ;IS IT ,+2?
3374  017416  001006                          BNE     3$                ;NO
3375  017420  062705  000002                  ADD     #2,TTY            ;ADD 2 TO THE ADDRESS
3376  017424  042705  000001                  BIC     #1,TTY            ;BACK UP TO AN EVEN BYTE
3377  017430  010566  000002                  MOV     TTY,2(6)          ;RESTORE ADDRESS
3378  017434  012605                  3$:     MOV     (6)+,TTY          ;RESTORE TTY
3379  017436  000002                          RTI                       ;RETURN
3380
3381  017440  005015     000          RETURN: .ASCIZ  <15><12>          ;RETURN AND LINEFEED
3382  017443     015  020012  020040  SPACE:  .ASCIZ  <15><12>"   "     ;RETURN AND 3 SPACES
3383  017450     000
3384          017452                  .EVEN
3385  017452  000000                  .TYPE:  0                         ;CHARACTER TYPE LOCATION
3386          000001                  .END
```

```
ANS1      000604          ANS2      000606          ANS3      000610          ANS4      000612
ANS5      000614          ANS6      000616          BEGIN     001010          BELL    = 000007
DISPLA  = 177570          DONE      015670          END1      001240          END10     002346
END11     002554          END12     002700          END13     003104          END14     003224
END15     003346          END16     003560          END17     003706          END2      001364
END20     004114          END21     004242          END22     004366          END23     004510
END24     004634          END25     005042          END26     005164          END27     005376
END3      001506          END30     005524          END31     005652          END32     006000
END33     006126          END34     006250          END35     006376          END36     006520
END37     006642          END4      001626          END40     006764          END41     007172
END42     007316          END43     007526          END44     007650          END45     010062
END46     010210          END47     010422          END5      001746          END50     010544
END51     010750          END52     011074          END53     011306          END54     011430
END55     011636          END56     011776          END57     012136          END6      002074
END60     012276          END61     012436          END62     012646          END66     014226
END67     014372          END7      002222          END70     014744          END71     015300
END72     015634          ERRORS    001002          ERROR$    016734          ERR1$     016752
FADD    = 075000          FDIV    = 075030          FISLVL    000652          FISVEC    000650
FMUL    = 075020          FSUB    = 075010          HLT     = 104000          HLTAD$    016732
HLT$      016612          ICNT      001000          ILLUP     017304          ISA13     003000
ISA20     004006          ISA25     004734          ISA41     007064          ISA51     010644
ISA55     011530          ISA70     014624          ISA71     015156          ISA72     015512
ISR11     002430          ISR13     002756          ISR16     003430          ISR20     003764
ISR25     004712          ISR27     005246          ISR41     007042          ISR43     007400
ISR45     007732          ISR47     010272          ISR51     010622          ISR53     011156
ISR55     011506          ISR66     014076          ISR67     014312          ISR70     014526
ISR71     015060          ISR72     015414          KIT$      016112          LAD$      016140
N       = 000073          OVER$     016116          PASSES    001004          PC      =%000007
PDOWN$    017136          PDVEC$    017314          POPER     016406          POPER1    016414
POPES     016240          POPR      016354          POPS      016204          POP7      016536
POWER$    017324          PRINTR    016776          PRINTS    017006          PS      = 177776
PUP$      017210          PUSHR     016322          PUSHS     016144          PUSH7     016500
PUVEC$    017320          RETURN    017440          RTA11     002414          RTA13     002742
RTA16     003414          RTA20     003750          RTA25     004676          RTA27     005232
RTA41     007026          RTA43     007364          RTA45     007716          RTA47     010256
RTA51     010606          RTA53     011142          RTA55     011472          RTA66     014062
RTA67     014274          RTA70     014520          RTA71     015052          RTA72     015406
R0      =%000000          R1      =%000001          R2      =%000002          R3      =%000003
R4      =%000004          R5      =%000005          SAVRTS    016476          SAVTPS    017312
SCOPE   = 104400          SCOPE$    016006          SP      =%000006          SPACE     017443
STACK0    000630          STACK1  = 000631          STACK2    000632          STACK4    000634
STACK6    000636          STACK8    000640          STK56     011702          STK57     012042
STK60     012202          STK61     012342          SVLAD$    016072          SWR     = 177570
SW08    = 000400          SW09    = 001000          SW10    = 002000          SW11    = 004000
SW12    = 010000          SW13    = 020000          SW14    = 040000          SW15    = 100000
TIMES     016142          TPB       000656          TPS       000654          TRAPER    016606
TSA14     003170          TSA15     003310          TSA23     004452          TSA26     005126
TSA4      001572          TSA40     006726          TSA44     007612          TSA5      001712
TSA50     010506          TST1      001132          TST10     002236          TST11     002362
TST12     002570          TST13     002714          TST14     003120          TST15     003240
TST16     003362          TST17     003574          TST2      001254          TST20     003722
TST21     004130          TST22     004256          TST23     004402          TST24     004524
TST25     004650          TST26     005056          TST27     005200          TST3      001400
TST30     005412          TST31     005540          TST32     005666          TST33     006014
```

```
TST34   006142          TST35   006264          TST36   006412          TST37   006534
TST4    001522          TST40   006656          TST41   007000          TST42   007206
TST43   007332          TST44   007542          TST45   007664          TST46   010076
TST47   010224          TST5    001642          TST50   010436          TST51   010560
TST52   010764          TST53   011110          TST54   011322          TST55   011444
TST56   011652          TST57   012012          TST6    001762          TST60   012152
TST61   012312          TST62   012452          TST63   012662          TST64   013146
TST65   013470          TST66   014014          TST67   014242          TST7    002110
TST70   014426          TST71   014760          TST72   015314          TTY    =X000005
TYPCNT  016774          TYPE   = 000004         YESRT   016004          $PSW    000600
$SP     000602          .BIT   = 177777         .IOT    017334          .PR     017134
.PRBUF  017124          .PRF    017044          .PRL    017032          .SAVR6  017310
.TBIT   016002          .TYPE   017452          .      = 017454
```

ERRORS DETECTED: 0